TIBCO EBX® Match and Merge Add-on

Version 2.5.14
December 2021

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

Quick Start Guide

User Guide

Reference Manual

Administration Guide

Developer Guide

Release Notes

Quick Start Guide

CHAPTER 1

Overview

This chapter contains the following topics:

1.1 Overview

The TIBCO EBX® Match and Merge Add-on can locate and automatically correct duplicate or incomplete data. In order to find this data, you configure the add-on by telling it where and how to look, and when to automatically take action. The goal of this *Quick Start Guide* is to get you up and running by using the add-on's configuration wizard.

The wizard walks you through the basic configuration process. Upon successful completion, you can run matching operations. Additionally, the wizard shows you where configuration settings were created. If your business needs require it, you can then fine-tune these settings.

1.2 What's next?

Before you use the configuration wizard, you need to enable add-on features on the desired data model. See Prerequisites [p 13] for further instructions.

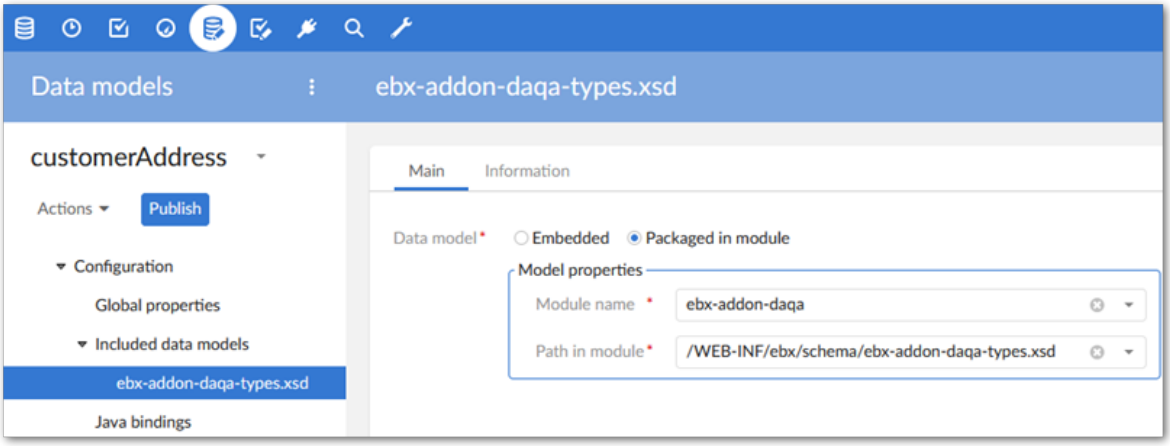
CHAPTER 2

Prerequisites

Before configuring the add-on, you must include add-on functionality in the desired data model. Additionally, you add special add-on metadata to each table you want to participate in matching operations.

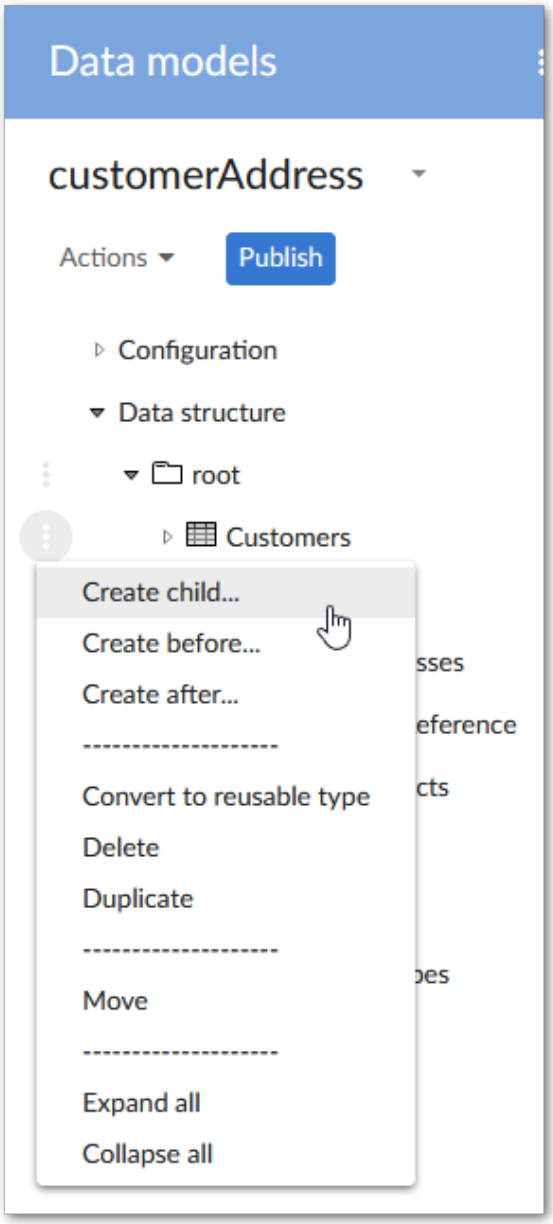
To configure a data model and table:

1. On the TIBCO EBX® main menu bar, select the **Data models** icon and use the drop-down menu to select the desired data model in the Data Modeler Assistant (DMA).
2. In the **Navigation pane**, expand the **Configuration** group and select **Included data models**.
3. Create a new record and specify the following:
   - **Data model**: **Packaged in module**
   - **Module name**: **ebx-addon-daqa**
   - **Path in module**: **/WEB-INF/ebx/schema/ebx-addon-daqa-types.xsd**



4. Select **Save and close**. You can now add add-on metadata to the tables you want to participate in matching operations.

5. In the **Navigation pane**'s **Data structure** group, select the menu to the left of the table where add-on metadata should be included and select **Create child**.



6. After supplying the following information, select **Create**:
   - Enter `DaqaMetaData` in the **Name** field.
   - Optionally, add a user-friendly label.
   - For the **Kind of element** option, select **Group**.
   - For the **Data type** option, choose **Included data models** and from the drop-down menu, select **Inline Match and Merge Data**.
7. After completing the above tasks, select **Publish** and follow the on-screen instructions to push these changes to your data model.

This chapter contains the following topics:

2.1 What's next?

You can now use the configuration wizard to determine how matching executes on the tables configured above. See Creating a configuration [p 17] for instructions on accessing the wizard.

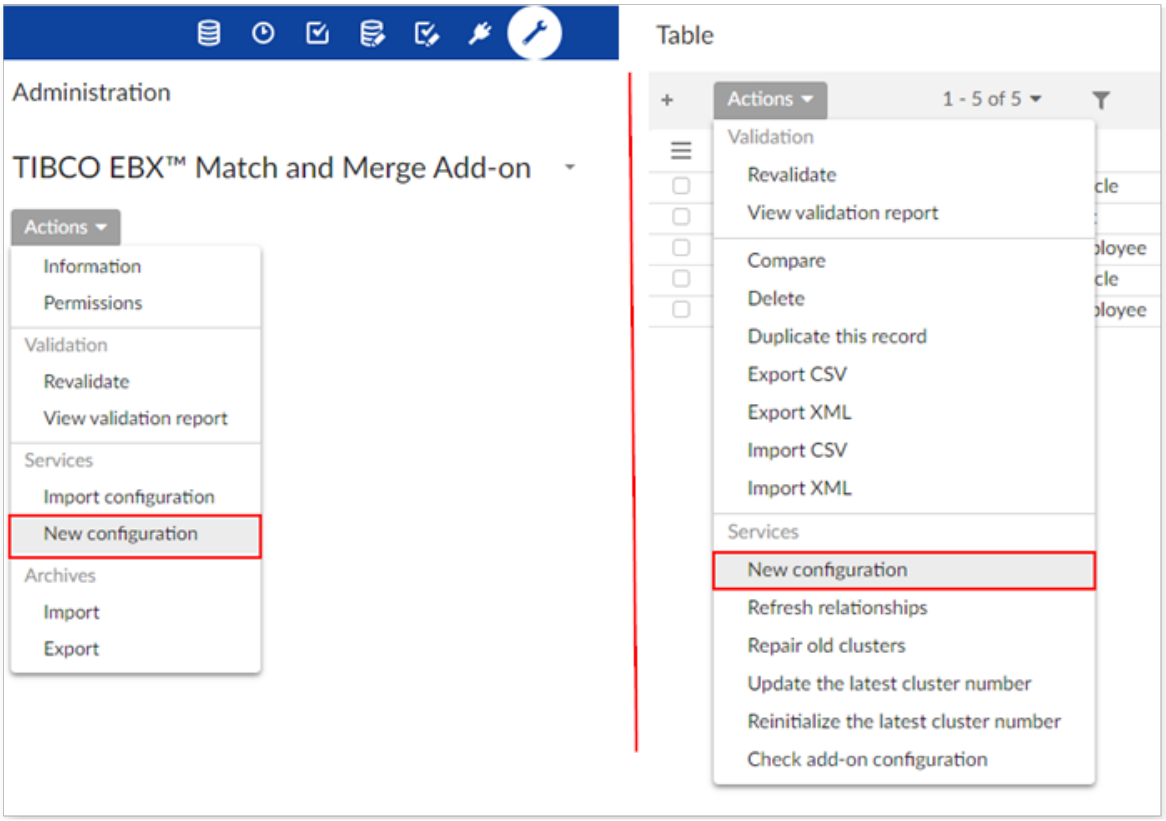
CHAPTER 3

Creating a configuration

This section shows you how to access the configuration wizard and explains its features. You must be an administrator to access the wizard.

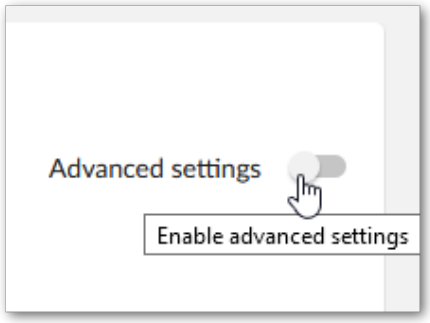
> **Note**
>
> If you have not already enriched your data model with add-on functionality, see Prerequisites [p 13] for instructions.

To access the wizard, navigate to *Administration* > *Data quality and analytics* > *TIBCO EBX® Match and Merge Add-on* and from the **Actions** menu, select **New configuration**. Note you can also access this option from *TIBCO EBX® Match and Merge Add-on* > *Table*.



The following highlights wizard features:

- Each page of the wizard presents you with a basic set of configuration options. Where available, you can also enable edit of advanced settings by selecting the option at the top-right of the screen.



- You can use the buttons at the bottom of the page to navigate.
- When you finish, the wizard displays a summary page that shows you where configuration settings were updated and the name of the records containing these settings. You can use the icons to navigate to these locations.

User Guide

CHAPTER 4

Overview

This chapter contains the following topics:

4.1 Add-on description

The TIBCO EBX® Match and Merge Add-on finds duplicate records in related tables. This facilitates data management and cleansing to avoid data quality issues due to duplicate, or incomplete data. Based on how you configure table field match settings, the add-on automatically identifies which records are suspected duplicates. A rich, TIBCO EBX® integrated, end-user interface provides all the services required to govern duplicate records. Users can reject or merge duplicate records in order to build an accurate record, also known as a *golden record*.

Attention

To use the EBX® Match and Merge Add-on, you must also deploy the TIBCO EBX® Information Search Add-on.

4.2 Intended audience

The following types of users interact with this add-on and will benefit from the information contained in this document:

- A '**Data Steward**' uses EBX® without requiring any specific training for the add-on. Feedback is received when the creation or modification of a record results in a potential duplicate record (based on real-time matching).
- A '**Matching Data Steward**' uses the add-on to manage records that are considered suspect. This is accomplished by using a stewardship user interface provided by the add-on that is also available through an EBX® workflow. For example, a workflow can be automatically executed when Data Stewards create or modify records that result in a suspect record.

- A '**Matching Steward**' uses the add-on to configure and test matching policies. This type of user also uses the add-on to clean up existing data and perform a data cleanse after importing bulk data.

4.3 Integration into EBX® MDM

Since the add-on is part of EBX®, all MDM data governance features of EBX® are fully leveraged, such as:

- **Audit trail:** Every matching operation is historized.
- **Permission management:** User roles can be finely configured, such as at the field, record, or matching service-level.
- **Data staging:**Matching benefits from the dataspace life cycle. For example, it is possible to use specific matching policies configured by dataspace and dataset.
- **Workflow:** Data workflows can be used to create matching tasks performed in collaborative processes.

4.4 User prerequisites

Basic knowledge of EBX® features is a prerequisite to use the add-on.

4.5 Typical use case scenarios

You can see the following typical use cases when using the add-on:

- Use matching to find the golden record and potential duplicates.
- Use matching to find the golden record and merge the potential duplicates.
- Use cleansing to detect, correct inaccurate or missing data, and delete obsolete records.
- Use crosswalk to match data in difference tables.

CHAPTER 5

Getting started

This chapter contains the following topics:

5.1 Initial configuration

The following points are important to understand before starting to use the add-on:

- Every table managed by the add-on is enriched with a data type that declares matching metadata. This metadata is required for add-on execution and is fully managed by the add-on. For instance, the matching metadata includes the matching score and the state of every record. The complete list of this metadata is defined in 'matching metadata' section (refer to 'Installation and first configuration' section to include this data type in a table).
- Before using matching, it is necessary to configure matching policies. Many parameters are used to determine how matching is executed, to decide-depending on certain rules-how the automatic merging of records takes place (or not), which algorithm is used to find duplicate records, etc.

The first part of this user guide is dedicated to explaining how to master matching configuration. Depending on your business context, many different configurations could be created by:

- changing the fields used for matching
- switching which algorithms are employed
- adjusting threshold levels at which automatic merging is triggered.

With the add-on, an unlimited number of matching policies can be configured and coexist in your environment. A matching configuration administrator should be identified in your organization to be responsible for these policies over time. In a given situation, once the most suitable policies have been identified, it should no longer be necessary to modify them. However, as business needs evolve over time, it is important to have the ability to smoothly adapt the policies with the changing needs.

Hence, before starting to configure matching policies, it is important to have a solid understanding of all parameters that the add-on uses.

To start right now you can refer to the *Installation and first configuration* section.

To use the data cleansing feature please refer to the *Data Cleansing* chapter that collects all information related to this functional domain.

5.2 **Vocabulary**

A basic understanding of the following terms is beneficial when working with the add-on:

Topic	Term	Definition
Record state	Suspicious, Suspect, Pivot, Golden, Merged, Unmatched, To be matched, Deleted	Every record holds a state value. This value indicates a record's matching quality level.
Score	Similarity score	A record holds a similarity score against another record. This score is a percentage of similarity. Scores are computed by matching policies configured with the add-on. • '100%' means two records that are identical • '-1' means either a score is not yet computed or is not high enough to consider the two records as similar
Group of records	Cluster	A cluster is a group of records that are suspect with each other. A record can be attached to only one cluster at a time.
User Interface	EBX® regular view	This is the normal EBX® view, also known as the tabular view. The add-on does not impact this view. Two matching data are presented on every record to show the record's state and its cluster identifier. In order to hide this information and to display golden records only you can use a filtered view. EBX® services offer access to the light and full matching views.
	Light matching view	This view is provided by the add-on. It allows you to manage suspect records already identified and presented in a cluster. The light matching view can be integrated into a workflow layout.
	Full matching view	This view is provided by the add-on. It allows you to interact with all available matching features. The user interface is divided into two frames. One to launch matching operations and a second to assess matching outcomes. The full matching view is self-sufficient and can be integrated into a workflow layout.
	Simple matching view	This view is provided by the add-on. It allows you to decide what action to take on a suspicious record (directly set as golden, deleted, move it to a pivot record).
Stewardship	The stewardship process begins when a suspect record must be managed.	
Survivorship	The survivorship process begins when a suspect record can be merged automatically into a pivot or golden record.	
Relationship matching	Matching executes on records that have foreign keys referring to the current record.	
Surrogate matching	When the returned matching result on the configured matching field is lower than 'Field stewardship min score (%)' on a record, the surrogate field values are taken to match instead.	

Special notation:	
✔	Important recommendation to use the add-on.
✖	This feature is not yet available in the current release.

CHAPTER 6

Matching

This chapter contains the following topics:

6.1 Introduction

The EBX® Match and Merge Add-on finds records that might be duplicates. You can run it manually and configure it to run automatically during certain operations. You can search within a table and configure it to search related tables. The results include a score that indicates how closely records match. You can resolve matches, or configure the add-on to automatically handle conflicts in different ways based on the score.

6.2 Matching configuration

This section describes how to configure add-on functions.

Matching configuration overview

Before using matching, configuration is needed to define:

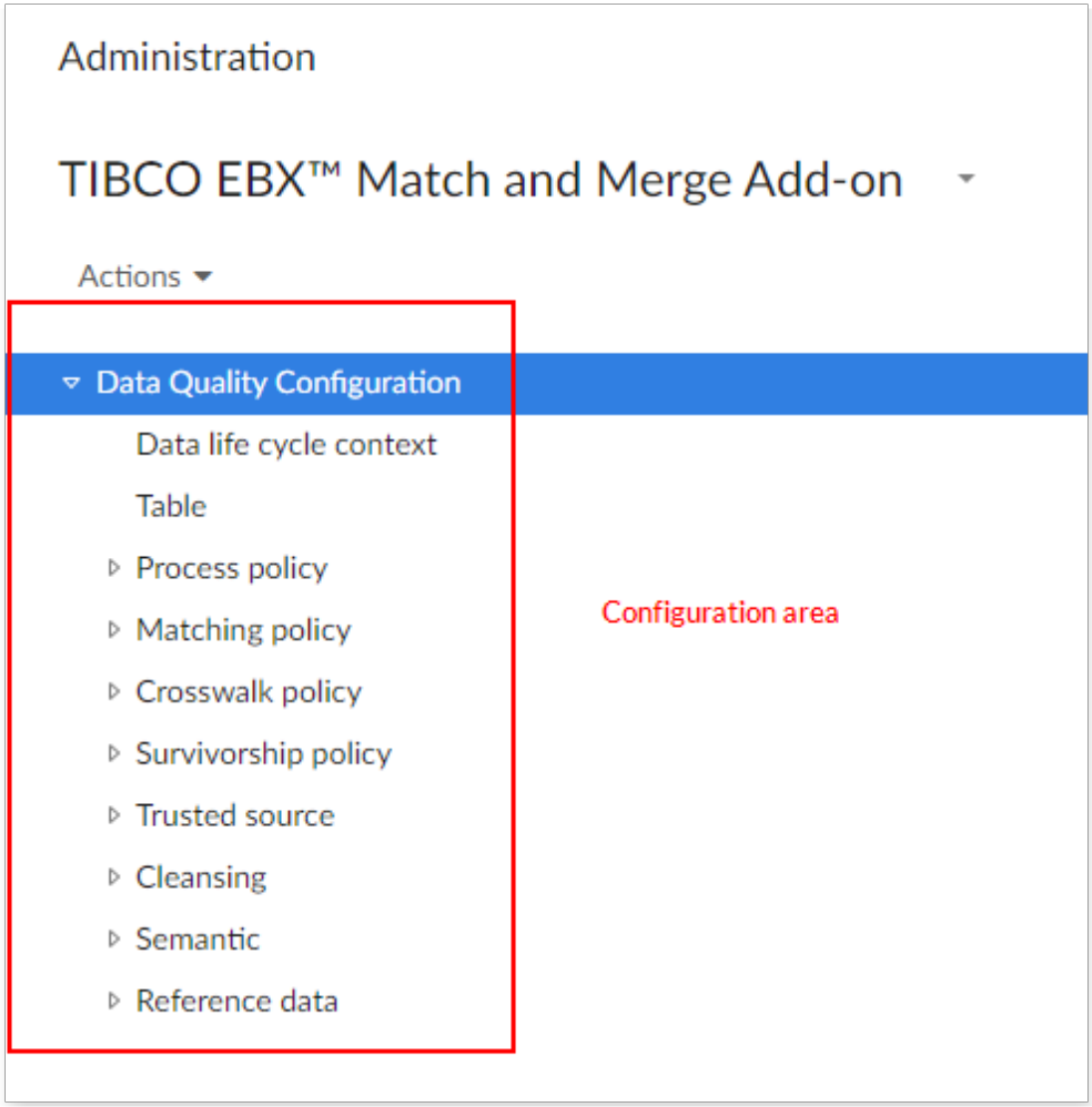
- The tables that are governed by the add-on.
- The operations for which matching is used (create, update).
- The algorithms used to perform the search for duplicate records (phonetic, distance)

> **Note**
>
> You can test new or existing configuration settings to ensure they returns the desired results. See Testing a matching policy [p 78] for more information.

Configuration can be managed by staff that are in a 'Matching Steward' role. During day-to-day data modification, this configuration should not be modified by other staff such as those with 'Data Steward' roles.

The four main levels of matching configuration that require management include: Table configuration, Process policy, Matching policy and Survivorship policy.

A partial view of the logical data model for matching configuration is presented in *EBX® Match and Merge Add-on data model* section. In this user guide, both the business and logical names of each table are highlighted so that it is easy to refer to the logical data model if needed.



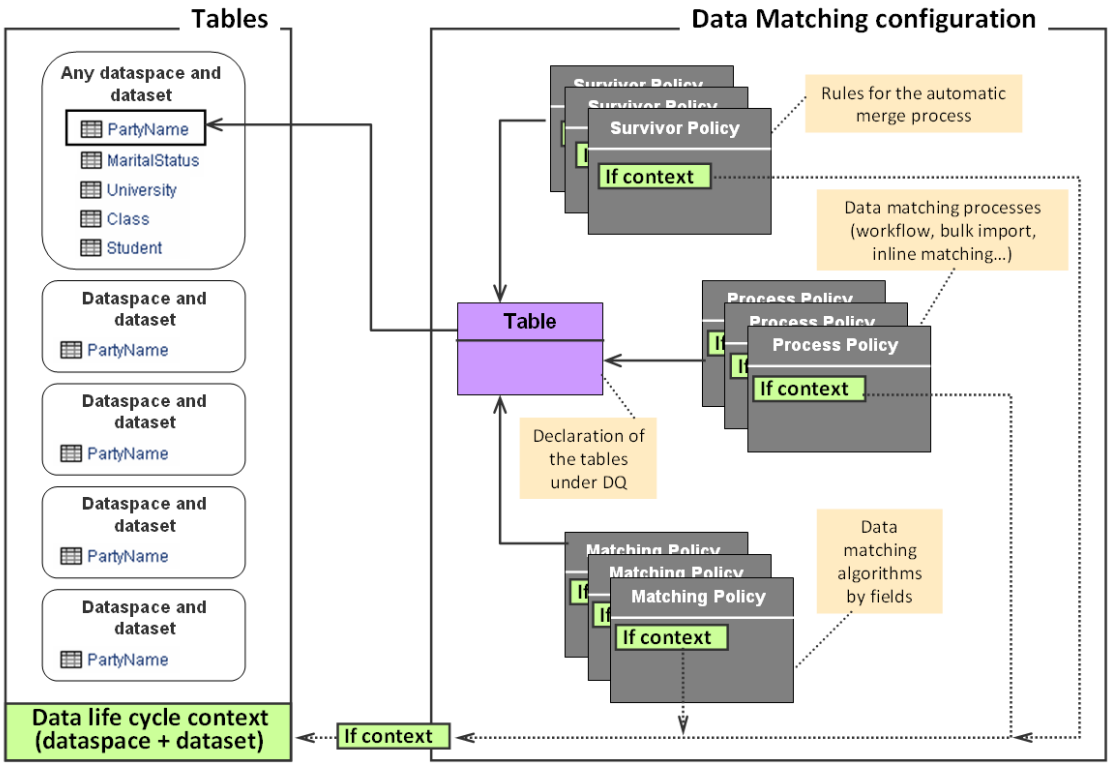
The **TIBCO EBX® Match and Merge Add-on** dataset located under the EBX® 'Administration' tab is used to configure matching policies, cleansing policies and crosswalk policies.

The 'Matching reference data' dataset (also located under the EBX® 'Administration' tab) contains reference data used by the add-on. The 'Matching state machine configuration' dataset must not be modified.

Special notation:	
✔	If the matching configuration is not correct at execution time, then any creation or modification of records is set to unmatched. Even though matching configuration quality and integrity are checked when configuring the add-on, it is still possible that some matching use cases are not covered by the configuration. For example, a business context could be raised but not yet be declared in the current matching configuration.

Concepts

EBX® Match and Merge Add-on configuration relies on these key concepts: table declaration, process policy, matching policy and survivorship policy.



In the above illustration, the 'Table' declares which tables are under the add-on control. The list of existing tables is obtained by configuring a data model containing the targeted tables. In the figure, the 'PartyName' table is set under add-on control. The localization of this table can evolve to other dataspaces and datasets over time. It will not change this declaration. A table requires a unique declaration regardless of its localization in many dataspaces and datasets.

A process policy defines a add-on process strategy such as: use of the workflow, import mode, inline matching, etc. Multiple strategies can be configured for one table. It is possible to change the strategy by data life cycle contexts based on a dataspace and dataset.

A matching policy defines the add-on algorithms to apply on the fields involved in the matching score computation. Several matching policies can be defined by datasets and dataspace, by business contexts (values of fields in the record to match) and by workflow.

A survivorship policy defines the rules to apply when executing the automatic merge of suspect records into golden or pivot records. They can be contextualized by dataspace and dataset.

Creating a test configuration

Once the add-on metadata is included to your table (refer to *Installation and first configuration* section), you can create a simple configuration that helps you determine how well the add-on works in your environment.

After testing this initial configuration, you can create another to import your data or initialize an existing set of records (refer to *Installation and first configuration* section). Then you will be ready to create your own portfolio of matching policies.

This table highlights setting your first test configuration.

Table	Field to configure at minimum
Table	Select the data model and then the table. Select the your data language. Other properties do not need to be used or modified for this minimum configuration.
Process policy	Select your table. Other properties do not need to be used or modified for this minimum configuration. **Your process policy is set up by default to run in a direct matching mode without a workflow.**
Matching policy	Select your table. Set the 'Active' property to 'Yes'. Select the Main matching algorithm. Other properties are not used or modified.
Matching field	Select your matching policy and submit. Select the field you want to use to match your data. Other properties are not used or modified.

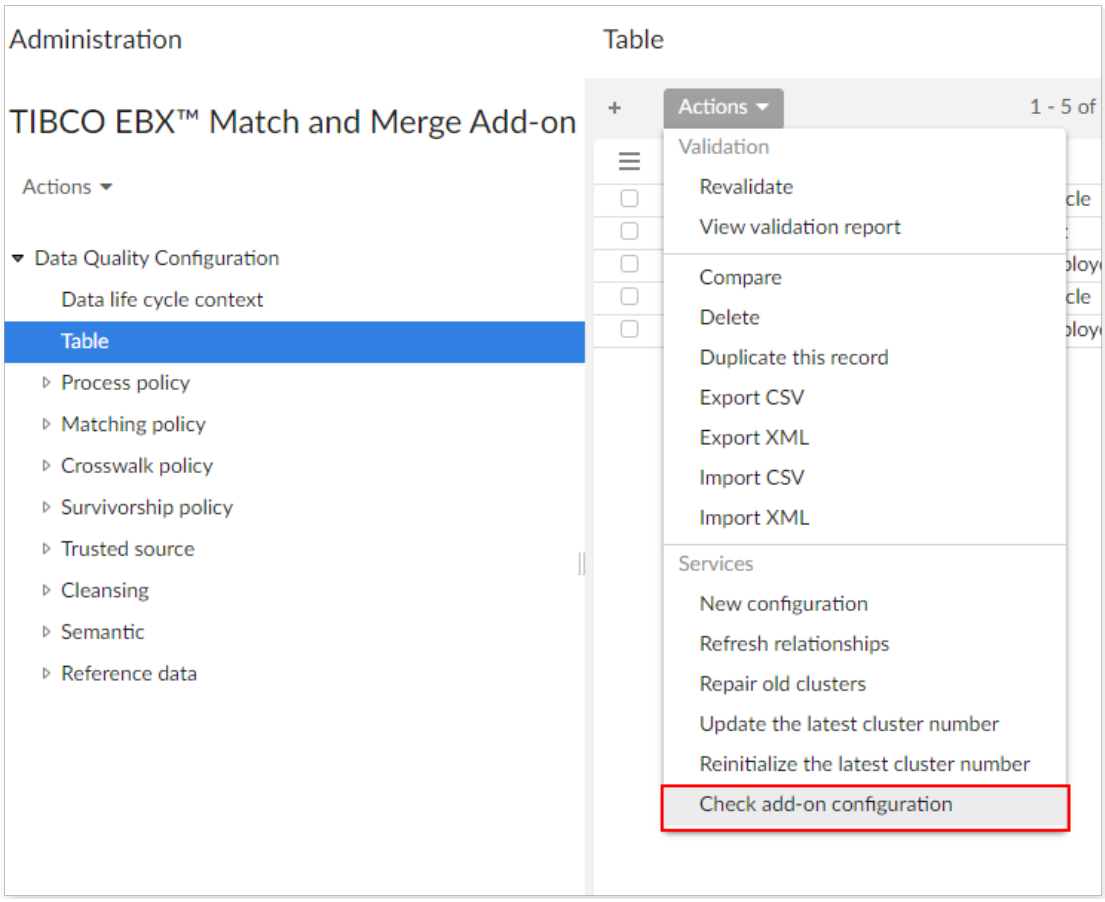
Table 1: A first quick and easy configuration

The Data life cycle context, Matching policy context, Survivorship policy, Survivor field, Source, Table trusted source, Field trusted source tables are not required for your initial test configuration.

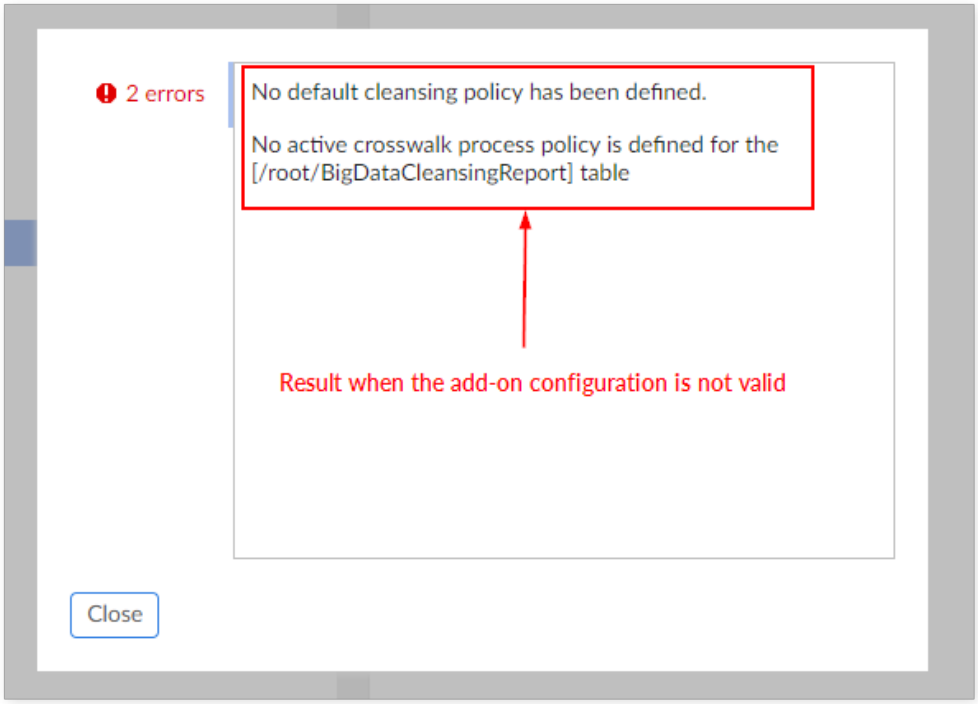
Checking the configurations

You can use the 'Check add-on configuration' operation to determine whether a table's configuration is valid. The operation can raise errors.

Checking matching configuration is available on demand. It does not prevent the add-on from executing.



At execution time, if any errors exist, then user messages are raised and the default policy is applied: records are set to unmatched for any creation and modification.



Special notation	
✔	Checking of Matching policies executes if the 'Use matching' ('Table' configuration) property is set to 'Yes'. Checking of Cleansing policies executes if the 'Use cleansing' property is set to 'Yes'.

Table configuration

Use this table to register your tables with the add-on and enable desired features, such as cleansing and crosswalk.

To register a table with the add-on:

- Code: Use any naming convention to enter a code. The add-on uses this code to uniquely identify this configuration.
- Data model: Select the data model containing the table you want to register.
- Table: Select the table you want to register with the add-on.

Once you save the configuration, other properties display to enable or disable add-on features (Matching, Cleansing, Crosswalk, language, etc.). Please note that these properties only display when the registered table includes DaqaMetaData.

Table table (logical name: **TableConfiguration**)

Properties	Definition
Code	Any naming convention without white spaces.
Data model	The data model that contains the table to set under the add-on configuration.
Table	Name of table managed by the add-on. When the table has not been extended with the matching metadata type, then only the cleansing and crosswalk features are available.
Business ID	One or more fields used as the business identifier. This identifier is used by the matching policy relying on a literal score = Exact in order to group suspect records that have identical business identifiers (no fuzzy search, see example in *Matching policy with exact score* section).
Source field	Field in the table that is used for getting the record source. This parameter is optional and is used when the record selection policy is 'most trusted source'. A multi-valued field cannot be selected as a source field. To use such a field you need to create a new field with a function that aggregates the values into the accepted format. Then, this field can be used as a source field.
Field to exclude records from match	Field in the table that is used for excluding records when the match operation is executed. This parameter is optional and is used when the matching policy 'Exclude records from matching' property is not null.
Language	Language used by the matching procedure. The language is selected from 'ebx.locales', that is, the languages managed by EBX®. For international languages, meaning many languages, English must be selected.
Max number of records in cluster	The maximum number of records that can be matched with the pivot in a cluster when matching is executed. Note that this number does not include the record being matched. For example, if you specify a value of 7, there can be 8 records in the cluster. Set to undefined when the maximum number of records in a cluster is unbounded. Please, note that: • When this number is too high (more than 20 records) careful analysis of suspect records can be difficult. • When this number is too low (less than 5 records) then relevant suspect records can be rejected by the add-on. This is because only the best suspect records are kept in the cluster. For instance, if this limit is set to 5, then no more than five suspect records will be kept in the cluster even if more suspect records have scores that deem them important. • In batch execution mode ('match at once' services) it is better to set the 'Max number of records in cluster' property higher so that the suspect records are more rapidly collected in clusters.
Max number of records for group	The maximum number of records that can be matched with a random record (considered as pivot) in a cluster used by the 'Group at once (unmatched)' and 'Group at once (to be matched)' operations to group records (not including a random record selected by the add-on to). Set to undefined when the maximum number of records in a group is unbounded.
Latest cluster number	The cluster register begins at 11 and is incremented by one to an unbounded level. Every table holds its own cluster register. The 'Latest cluster number' value is used by the add-on to keep in memory the last cluster identifier created for a table. All the datasets that come from the same data model share the same 'Latest cluster number'.
Disable matching trigger	It is possible to deactivate the matching trigger used by the add-on. This trigger allows you to put the table under add-on control. It is embedded in the data type that contains the matching metadata.

Properties	Definition
	The de-activation of the matching trigger is used when it is needed to implement the invocation to the matching in a bespoke trigger (via the add-on's API) that lives together with other add-ons such as the TIBCO EBX® Insight Add-on or any other software invocation at the CRUD time. If the matching trigger is inactive and the add-on's API not implemented then the table is no longer under the control of the add-on. Every new record will get an undefined state value that will be possible to set up later with the 'Set at once' operation. If 'Yes': The matching trigger is inactive. If 'No': The matching trigger is active. Default value: 'No'
Event listener	Java implementation listening to matching events.
Use matching	If 'Yes': The matching feature is enabled for the selected table. If 'No': The matching feature is not enabled. The matching feature finds duplicate records and makes creation of a golden record possible. When 'Use matching' is set to 'No', created and updated records are set to the Unmatched state in the '000' cluster. Records with Merged/Deleted states cannot be modified. Records that have 'Unmatched' or 'To be matched' states can be modified; however, their state won't change.
Use cleansing	If 'Yes': The cleansing feature is enabled for the selected table. If 'No': The cleansing feature is not enabled. The cleansing feature assesses table data quality and fixes defects, such as missing field values.
Use crosswalk	If 'Yes': The crosswalk feature is enabled for the selected table. If 'No': The crosswalk feature is not enabled. The crosswalk feature creates cross-reference records by performing a matching process.
Activate Monitoring	If 'Yes': The TIBCO EBX® Activity Monitoring Add-on stores execution status. If 'No': The EBX® Activity Monitoring Add-on does not store execution status. Default value: 'No' Determines whether the TIBCO EBX® Activity Monitoring Add-on stores execution status. This setting applies to 'Match at once', 'Run match', and 'Exact match' operations.
Number of processed records to update status	Specifies the number of records the add-on must process before triggering a status update in the TIBCO EBX® Activity Monitoring Add-on. This value must be between 500 and 2000. By default, the execution status is updated in the TIBCO EBX® Activity Monitoring Add-on every time 1000 records are processed.

Table 2: Table configuration properties

Process policy

A process policy defines a set of parameters that the add-on uses to execute matching. Multiple process policies can be defined and attached to a table controlled by the add-on. Two important concepts must be understood to configure a process policy: 'double matching' and 'user interaction & workflow'.

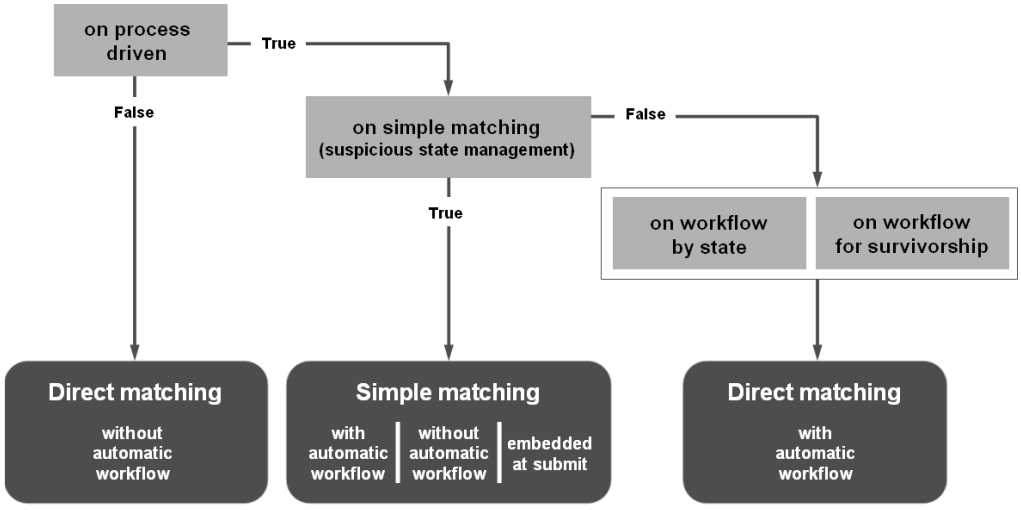
Double matching configuration

Every matching policy can be configured with two matching algorithms. The second algorithm is optional. It is used to deal with 'false negative' records that the first algorithm could ignore by error. A 'false negative' record is a record that should have been identified as a suspect record by the matching

procedure, but was not. The second algorithm is executed if the score of the first algorithm is between 0% and the value specified by the 'Threshold second algorithm' property. All suspects that are detected by the second matching algorithm are set with a score of 'Stewardship min score' + '0.1' to put them in the right area defined for the first algorithm. Above the stewardship area an automatic record merge-based on survivorship rules-is applied. The second algorithm is not able to run an automatic merge and the score of every suspect is set to 'Stewardship min score' + '0.1'.

User and workflow interaction

The add-on can be configured to launch automatic workflows when a duplicate record is identified. A portfolio of workflow user tasks and scripts is provided to facilitate the design of any bespoke workflows (see *Workflow integration* section). When the add-on is configured with no automatic workflows, it still remains possible to launch global EBX® workflows outside the scope of the add-on. These workflows can use all user tasks and scripts provided with the add-on. Two modes of matching can be configured: 'Direct matching' to compute suspect records directly (not requiring user input), or 'Simple matching' with the 'Suspicious' state management (requires user input). See *Record level matching* section for more information. For the 'Simple matching' mode, it is possible to configure the add-on with the 'embedded at submit' property to directly display the duplicate records at submit time without workflow execution. The possible configurations to drive the user interactions and the workflow interaction are highlighted in the figure below.



Here are some common configurations:

Context of use	On process driven	On simple matching	On workflow by state	On workflow for survivorship
Direct matching without workflow	False	any	any	any
Simple matching with automatic workflow	True	True + workflow	any	any
Simple matching with no workflow	True	True	any	any
Simple matching embedded at submit time	True	True + 'embedded at submit time'	any	any
Direct matching with workflow except for survivorship	True	False	True	False
Direct matching with workflow	True	False	True	True

Table 3: Usual process policy configurations

> **Note**
>
> When creating a workflow, the **Auto complete** property must be set to `false` if you want to use the merge process output for the next step.

If the 'On process driven' policy is set to 'Yes' and something is not correct in the workflow configuration, either at the 'Simple matching' level or the 'Workflow by state' level, then the 'Direct matching and no workflow' configuration is used by default.

In addition to these configurations, it is possible to deactivate matching either on creation or modification of any record (see *Processes applied to the creation and modification of records* section). It is also possible to deactivate survivorship execution (automatic merge).

The services applied to a set of records, such as 'Match at once', do not create workflows no matter what configuration is used.

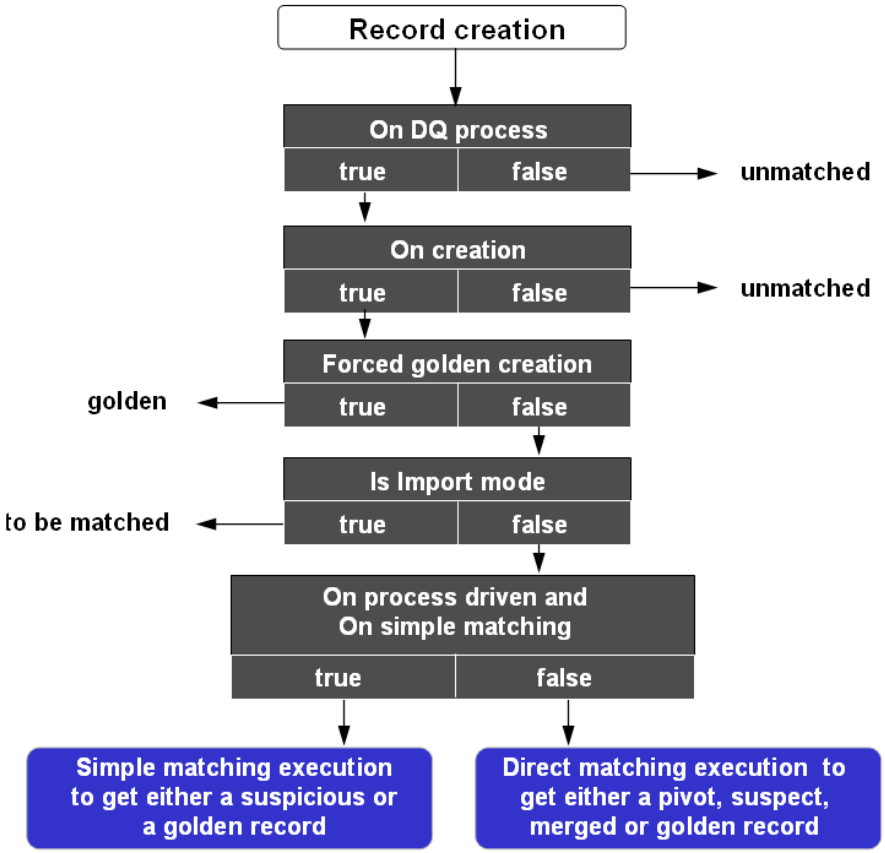
Description	Application	Example
Direct matching without workflow		
Any creation or modification of a record entails a matching execution against the table. The states of the impacted records are immediately updated. The workflow is not used.	**Suitable when cleansing an existing table, importing bulk data and mass data entry.** This method is useful when a company does not need workflows or prefers to manage workflows outside of add-on control (*). **(*) All components of the matching UI (light and full views) can be easily integrated in any workflow, with a data context providing a record or cluster identifier.**	Creation of a record that moves to the pivot state in a cluster that contains suspect records. **Creation of a record that becomes a golden record in the '001' reserved cluster.**
Simple matching (human decision required when suspect records are detected)		
Any creation or modification of a record executes a search for matching records. This search does not entail modification of the impacted record's states that are identified by the add-on. These records are presented to the user who decides to keep the new/updated record as a golden record, to cancel the new/updated record or to manage a merge procedure applied to the matching records. When there are no matching records, then the new/updated record is automatically considered a golden record. A workflow for creation and modification can be configured.	**Suitable in real-time data entry when creation or modification of a record must be matched against the table to determine whether or not it is a suspicious record.** Based on a human decision applied on the suspicious record, either the new/updated record is canceled or confirmed as a golden or a pivot record. Note: it is possible to create any bespoke workflow with a first step for record creation and modification. Then, the second step is the simple matching view. In this case the workflow properties must be set to none.	Creation of a record that is flagged as a suspicious record. Then a workflow is launched to display all potential suspect records. The user decides how this suspicious record must be managed: golden record directly, cancel, merge with other potential suspect records.
Direct matching with workflow		
The creation or modification of a record executes matching on the table. The states of the impacted records are directly updated. Depending on the state of the new/updated record a workflow can be configured. The add-on executes this workflow automatically.	**Suitable in the same use case as 'Direct matching without workflow'.**	Automatic creation of a workflow when a record moves to the pivot state.

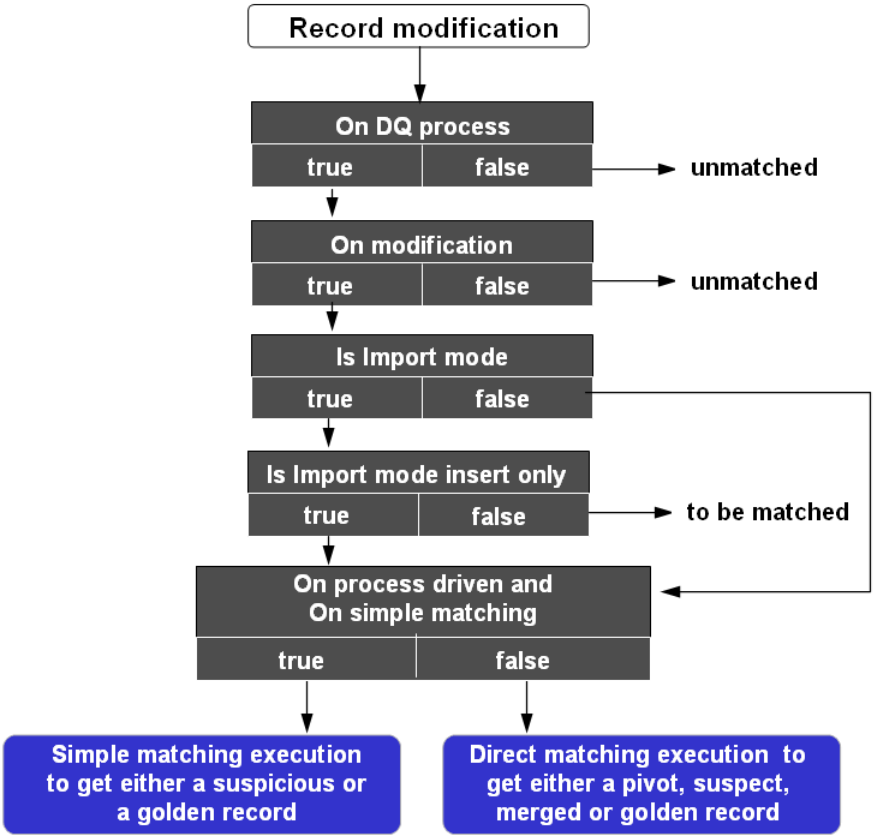
Table 4: On process driven configuration

Processes applied to the creation and modification of records

The configuration of a process policy allows you to define different options applied to the creation and modification of records. The properties used are as follows: 'On matching process', 'On creation',

'On modification', 'Forced golden creation', 'Is import mode', 'Is import mode only' and 'On simple matching'.





The following, shows common use contexts and their corresponding configurations:

Context of use	On DQ Process	On Creation	On modification	Forced golden creation	Is import mode	Is import mode insert only
Interactive matching	True	True	True	False	False	False
Bulk data import	True	True	True	False	True	False
Bulk data import for new records only	True	True	True	False	True	True
Direct creation of golden records (interactive and import)	True	True	True	True	False	False
Deactivation of matching	False	any	any	any	any	any
Deactivation of matching when modification of records	True	True	False	any	any	any

Table 5: Example of processes applied to the creation and modification of records

Process policy configuration

A table's configuration must declare one active process policy with no context. When several process policies are active at the same time for different contexts, then the policy with no context is used by default for every execution that is not related to a defined context.

Process policy table (logical name: **ProcessPolicy**)

<table>
<tr><th>Properties</th><th>Definition</th></tr>
<tr><td>Code</td><td>Any naming convention without white spaces can be used.</td></tr>
<tr><td>Short description</td><td>The policy description.</td></tr>
<tr><td>Long description</td><td>Long description of the policy</td></tr>
<tr><td>Table</td><td>Link to the table used for this policy.</td></tr>
<tr><td>Active</td><td>If 'Yes': This process policy is used.
If 'No': This process policy is not used.
This field indicates whether or not the policy is used when running matches. Since the add-on manages the process policy context, one or more process policies can be active at the same time for a given table. To disable a process policy, set the active field to 'No'.
Only one process policy can be active for each context and a table.
One active process policy with no context must be configured. It is used by default process policy at execution time.
Default value: 'Yes'</td></tr>
<tr><td>Data life cycle context</td><td>A process policy can be defined for a specific dataspace and/or dataset. This property is a link to a data life cycle context (dataspace and/or dataset).
Leave the value undefined to deactivate the use of a context (any dataspace and dataset).
The priority of contexts is as follows:
dataspace and dataset
dataspace
No context
<table><tr><th>Process Policy</th><th>Data life cycle context</th></tr><tr><td>P001</td><td>No context</td></tr><tr><td>P002</td><td>Context 2</td></tr><tr><td>P003</td><td>Context 3</td></tr></table><table><tr><th>Data life cycle context</th><th>Dataspace</th><th>Dataset</th></tr><tr><td>Context 2</td><td>Dataspace 1</td><td></td></tr><tr><td>Context 3</td><td>Dataspace 1</td><td>Dataset 1</td></tr></table>In this example, the process policy P003 has the highest priority, then process policy P002 and P001.</td></tr>
<tr><td>On matching process (See use case in this user guide to understand the way of the add-on works depending on this parameter.)</td><td>If 'Yes': The table benefits from matching in the context defined by the 'Data life cycle context' property (any data space and data set if 'Data life cycle context' is undefined).
If 'No': Matching is inactive for the table in the context. The states of created records and modified records without a state are still systematically set to 'Unmatched'.
Default value: 'Yes'</td></tr>
<tr><td>On creation</td><td>If True: Matching is run at record creation.
If No: Matching is not run at record creation. The state of created records is systematically set to unmatched.
Default value: 'Yes'.</td></tr>
</table>

Properties	Definition
Search before create	When you enable **Search before create**, users can search existing records to avoid creating a potential duplicate.
On modification	If 'Yes': Matching is run at record modification. For Simple matching process (creation of suspicious records), the matching is run when matching fields are modified only. For all other cases, the matching is run when the record is modified whatever the fields. If 'No': Matching is not run at record modification. If a modified record's state was not set, it will be set to Unmatched. Default value: 'Yes'.
On survivorship	If 'Yes': The add-on merges records automatically when the match score is above the threshold set in the 'Stewardship max score' property. Note that any existing Merged records will be taken into account when performing the merge operation. If 'No': Survivorship is disabled. Default value: 'Yes'
Always apply survivorship	If 'Yes': The add-on merges records automatically when the match score is above the threshold set in the 'Stewardship min score' property. It means that every cluster created will be merged automatically. If 'No': The add-on merges records automatically when the match score is above the threshold set in the 'Stewardship max score' property. Default value: 'No'
Forced golden creation	If 'Yes': Record creation is set to a golden state in the '001' cluster without any matching execution. This property has no impact on procedures applied on modification and deletion of records. This property makes the 'Is import mode' property inactive. When the 'Forced golden creation' property is set to 'Yes' then the workflow is deactivated. If 'No': Forced creation of golden records is deactivated. Default value: 'No'
Is import mode	If 'Yes': The policy is used in the context of bulk data import. On creation and/or modification record states are set to 'to be matched' (refer to the next property 'Is import mode insert only'). If 'No': The policy is used in the context of interactive creation and modification of records. When the 'Is import mode' property is set to 'Yes' then the workflow is deactivated . Default value: 'No'
Is import mode insert only	This property is used only if 'Is import mode'='Yes' If 'Yes': Only new records move to 'to be matched'. If 'No': Updated and new records move to 'to be matched' Default value: 'No'
Merged record is recycled	If 'Yes': Modification of merged records is enabled. When a merged record is modified behavior depends on property settings as follows: • If **Modify merged without match** is activated: If the Merged record and its target (Pivot or Golden) are in the same cluster, the merged record remains in the current cluster. Survivorship is applied to merge data from all Merged records in the cluster to the target record—'Is forced to void' is ignored. Otherwise, the Merged record remains untouched. • Otherwise, the merged record becomes 'Unmatched'. If its target is null or 'Deleted', 'Unmatched' or 'To be matched', then a match table operation will be executed on the unmatched record. Otherwise, a match table operation will be executed on its target. Eventually, if the unmatched record remains unaffected by previous steps, a match table operation will be executed on it. If 'No': Modification of merged records is disabled.

Properties	Definition
	This option is disabled when 'Is import mode' is set to 'Yes'. Default value: 'No'.
Stewardship min score (%)	When a match score is lower than this threshold then the add-on does not consider the record as a suspect record.
Stewardship max score (%)	When a match score is higher than this threshold, the add-on enforces an automatic merge of this record into the corresponding pivot or golden record. This procedure is also known as 'survivorship'. It is important to select this value carefully. If it is set too low, records may be automatically merged with pivot and golden records that are not actually related. When scores fall between the 'Stewardship min score' and the 'Stewardship max score' values, human interaction is needed to decide how to manage the suspect record (stewardship process).
Threshold second matching(%)	When the score computed by the first algorithm is between 0% and the value specified by the 'Threshold second matching(%)' property, then the second algorithm is applied to detect potential false negative records. For example, if 'Threshold second matching' is set to '5%' then the second algorithm is executed each time the score of the first algorithm is lower than '5%'. Constraint: 'Threshold second matching' is lower than or equal to 'Stewardship min score'. This property is used when 'Second matching algorithm' in the matching policy is not empty. Default value: '0'
Second level stewardship min score (%)	When the second matching level is configured to find false negative records from a first level matching, then this threshold is used to evaluate the matching score and whether or not it provides a suspect record. This property is used when 'Second matching algorithm' in the matching policy is not empty.
Not suspect with threshold (%)	When a record 'A' is 'Not suspect with' a record 'B', the score of 'B' against 'A' is kept by the add-on. When a new matching is executed, if the variance (new score - former score > 'Not suspect with threshold') then 'B' is no longer considered as 'Not suspect' and becomes a suspect of 'A'. For instance, 'Not suspect with threshold'=10% and score of 'B'=60% when it was stated 'Not suspect with'. If the new score of 'B' against 'A' is higher than or equal to 70% then it comes back as suspect otherwise it remains 'Not suspect with' the record 'A'.
On not suspect with	Allow to deactivate the 'not suspect with' feature. Case 1: 'On not suspect with' = 'Yes' and new score - old score < 'Not suspect with threshold' (note: new score is the score of current matching, old score is the score of suspect record in the list of not suspect of pivot record): • The record is not suspect, and list of not suspect with in Pivot does not change. Case 2: 'On not suspect with' = 'Yes' and new score - old score >= 'Not suspect with threshold': • The record is suspect, and it is removed from the list of not suspect with in the Pivot. Case 3: 'On not suspect with' = 'No' and new score - old score < 'Not suspect with threshold': • Record is Suspect and it is removed from the list of not suspect with in the Pivot. Case 4: 'On not suspect with' = 'No' and new score - old score >= 'Not suspect with threshold': • Record is Suspect and it is removed from the list of not suspect with in the Pivot. If 'Yes': When a match is executed the 'not suspect with' records participate in the matching only if the 'not suspect with threshold' is reached. If 'No': When a match is executed the 'not suspect with' records participate in the matching as regular records. Default value: 'No'

Properties	Definition
Bidirectional not suspect with	If you enable this property and run the 'Not suspect with' service on a Suspect record, the add-on saves any 'Not suspect with' records in the Pivot and Suspect record's metadata tabs.
On suspect record retention	If you set this property to 'Yes', when suspect records no longer match with their Pivot records, the score of the impacted cluster will be recalculated. A new pivot is selected based on the 'Pivot selection mode' property setting. Survivorship is not applied; only the score is recalculated. This strategy can be changed by setting the 'On suspect record retention' property to 'No'. With this configuration, the suspect records that no longer match with the pivot record will be moved to the unmatched state. Default value: 'Yes'

Cluster retention on matching suspect	You can use this property when running a 'Match at once' in full mode, 'Match table' or 'Modify' operation on Suspect records. For the following examples, consider Suspect record A as the Pivot. The add-on matches A against all other records in the table. However, when running a 'Match at once in full mode' or 'Match table' operation, the match excludes records in A's cluster.
	• 'Match at once in full mode' — For this example, assume that record B scores the highest against A. If B's new score is greater than A's current score:
	  • When 'On cluster retention for matching suspect' is set to 'Yes', the add-on moves A into B's current cluster. However, if B is located in a predefined cluster, the add-on moves A and B into a new cluster.
	  • When 'On cluster retention for matching suspect' is set to 'No', the add-on moves A and B into a new cluster.
	• 'Match table' — Assume that records B, C, and D score higher than A and that B has the highest score of all.
	  • When 'On cluster retention for matching suspect' is set to 'Yes', the add-on moves A, C and D into B's current cluster. However, if B is located in a predefined cluster, the add-on moves A, B, C, and D into a new cluster.
	  • When 'On cluster retention for matching suspect' is set to 'No', the add-on moves A, B, C and D into a new cluster.
	• 'Modify' on a Suspect record — This entails the same behavior as a 'Match table' operation on a Suspect record. But, in this case the add-on matches A against all other records in the table including those in A's cluster.
	Default value: 'No'

On process driven	If 'Yes': The configurations defined with 'On simple matching', 'On workflow by state' and 'On workflow for survivorship' are used. If 'No': Direct matching without workflow is executed. Default value:'No'

On simple matching	'On simple matching'
	• This property activates/deactivates the 'On simple matching' property. Simple matching creates suspicious records when records are identified as potentially suspect.
	• If 'Yes': Simple matching is active (creation of suspicious record) and the properties contained in the 'On simple matching' group are used at execution time.
	• If 'No': Simple matching is not active and the properties contained in the 'On simple matching' group are not used.
	• Default value:'No'
	'Under workflow'
	• 'Workflow on creation'
	  • The name of a workflow the add-on automatically launches in case a record is deemed as suspicious at creation time. 'None' if no workflow is to be used.
	  • Default value: 'None'

Properties	Definition

	• 'Workflow on modification'
	  • The name of a workflow the add-on will launch automatically in case a record is deemed as suspicious at modification time. 'None' if no workflow.
	• Default value: 'None'
	'Under submit'
	• 'Embedded at submit'
	  • If 'Yes': In case of a suspicious record, the list of duplicate records is automatically displayed at creation and modification time (the simple matching view is used).
	  • If 'No': No automatic matching UI embedded at the submit on creation and modification.
	  • Default value: 'No'
	• Embedded functions
	  • If 'Yes': the function is available
	  • If 'No': the function is not available
	  • Default value: 'Yes'
	  • Make definitive golden
	  • Make golden
	  • Options on Make golden
	  • Make golden without merging
	  • Make golden and force all records to merged
	  • Make golden and force selected records to merged
	  • Delete
	  • Run stewardship on suspicious
	  • Run stewardship on pivot
	  • Run stewardship on best record
	  • Best record selection mode
	  • Merge and set golden
	  • Merge and set definitive golden
	  • Hide suspicious from list of duplicates
	If a workflow is configured for creation (or modification) then the embedded property at creation time (or modification) is no longer possible.
	When no workflow and no embedded properties are configured then the add-on does not provide any information to the user when a suspicious record is raised. This is the responsibility of the application to manage (or not) a process to inform the user.
	The embedded mode is possible only through the regular EBX® view, not from the light and full matching views since they already provide a full inline integration.

On workflow by state	To make this property active: 'On process driven'='Yes' and 'On simple matching'='No'. This property allows you to declare a workflow that the add-on will launch automatically depending on the matching result. For example, if [suspect, myWorkflowToManageSuspect] is configured, then this workflow will be created after the creation or modification of any record moving to the 'Suspect' state. An undefined value is used to deactivate this configuration. Default value: 'undefined' 1. State: This property allows you to declare the state that records become after a matching operation. 2. Workflow name: Name of the workflow launched when a record moves to the state defined by the 'State' property after a matching operation.

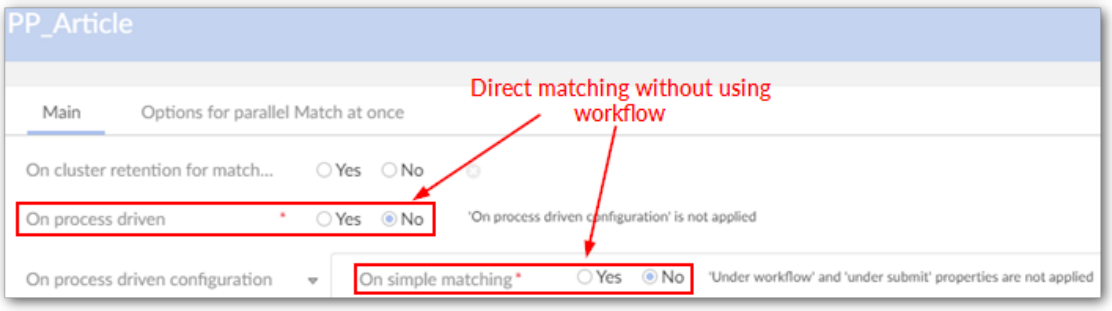
Properties	Definition
	When the add-on launches a workflow due to record creation or modification, the following variables must be declared in the data context. - branch: Reference of the data space. - instance: Reference of the data set. - tablePath: Path of the table. - xpath: XPath of the record. - workflowIdentifier: Unique identifier to track the record.
On workflow for survivorship	To make this property active: 'on process driven'='Yes' and 'On simple matching'='No'. Name of the workflow launched by the add-on in case of survivorship (at least one automatic merge has been executed). An undefined value is used to deactivate this configuration. Default value: 'undefined' When the add-on launches a workflow due to record creation or modification, the following variables must be declared in the data context. - branch: Reference of the data space. - instance: Reference of the data set. - tablePath: Path of the table. - xpath: XPath of the record. - workflowIdentifier: Unique identifier to track the record.
Options for Parallel Match at once: In the 'Match at once' screens, you can configure properties to divide records into groups and execute matching on these groups simultaneously. After matching completes, results are merged into the original data. If the number of groups is bigger than the number of threads, groups are evenly distributed in the different dataspaces/threads. This technical setting is equivalent to configuring the **Filter by** property in the Matching policy. It has no impact on the returned result, but improves the performance(memory and speed).	
Filter by	The add-on uses the fields in 'Filter by' to group records containing the same value into a thread. When you add a filter occurrence, you select one or more fields within the table: • When you choose one field — the add-on groups all records that contain the same value in that field. For example, if you selected Country, all records with the same Country value are put in the same thread; such as, USA in one thread, France in another and Vietnam in another. • When you add more than one field — the values in all specified fields must match to be included in a thread. For example, if you specified Country and Gender, the add-on puts all records with France and Female in one thread and those with France and Male into another thread.
Number of threads	Specifies the number of threads executed in parallel. For each thread, the add-on creates a dedicated temporary dataspace inside the original dataspace on which the Match at once operations are executed. Default value: 4
Commit threshold	The add-on commits results of Match at once operations in temporary dataspaces to the original dataspace when the number of executed records reaches this threshold. This threshold must be tuned according to your database. Default value: 100000

Table 6: Process policy properties

Examples - Process policy configurations

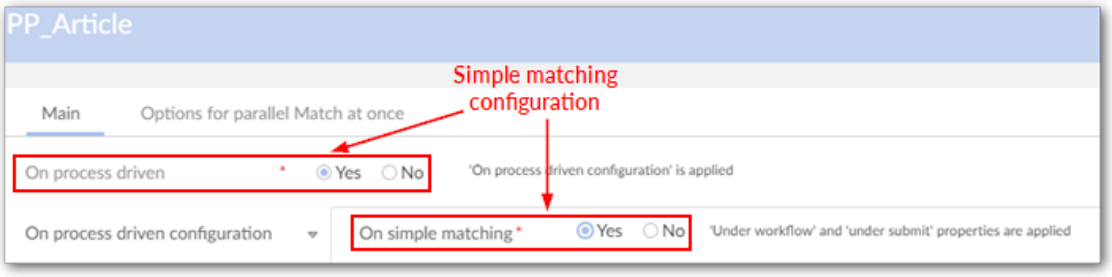
In these examples a table called 'Articles' is considered. You can configure your own table.

Direct matching without workflow

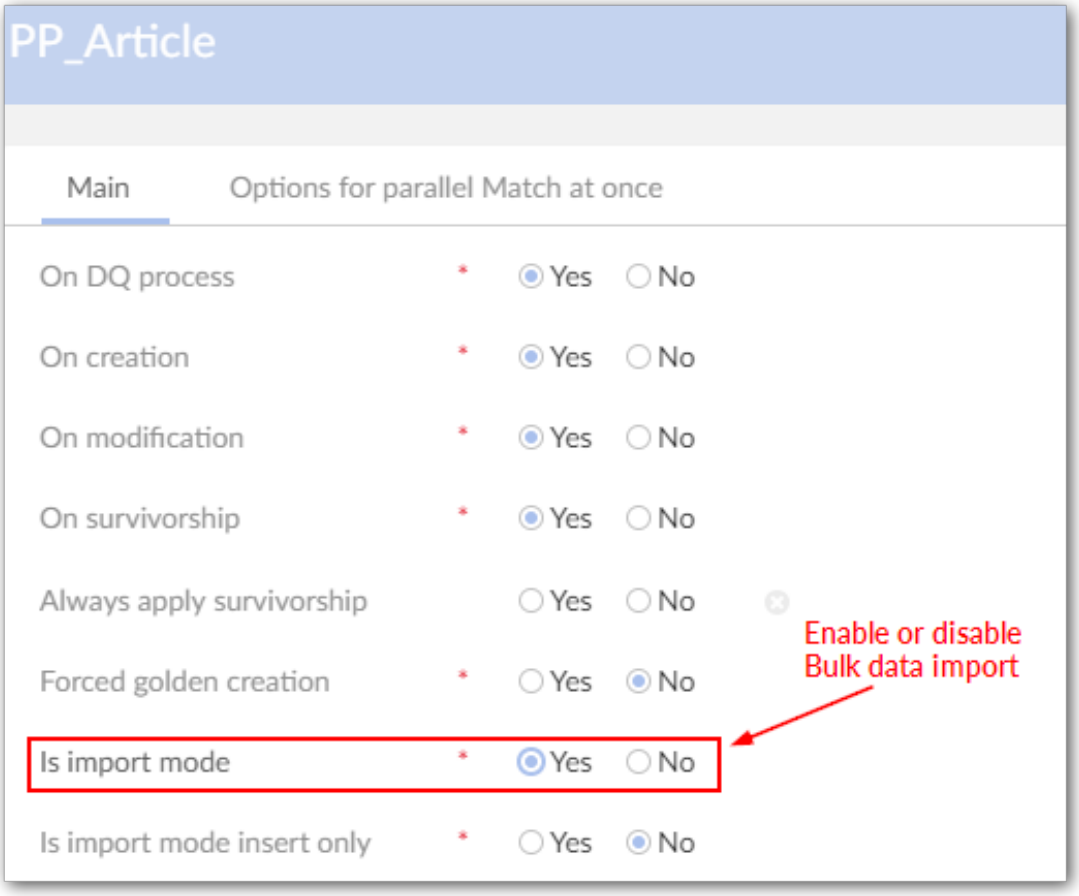


Simple matching

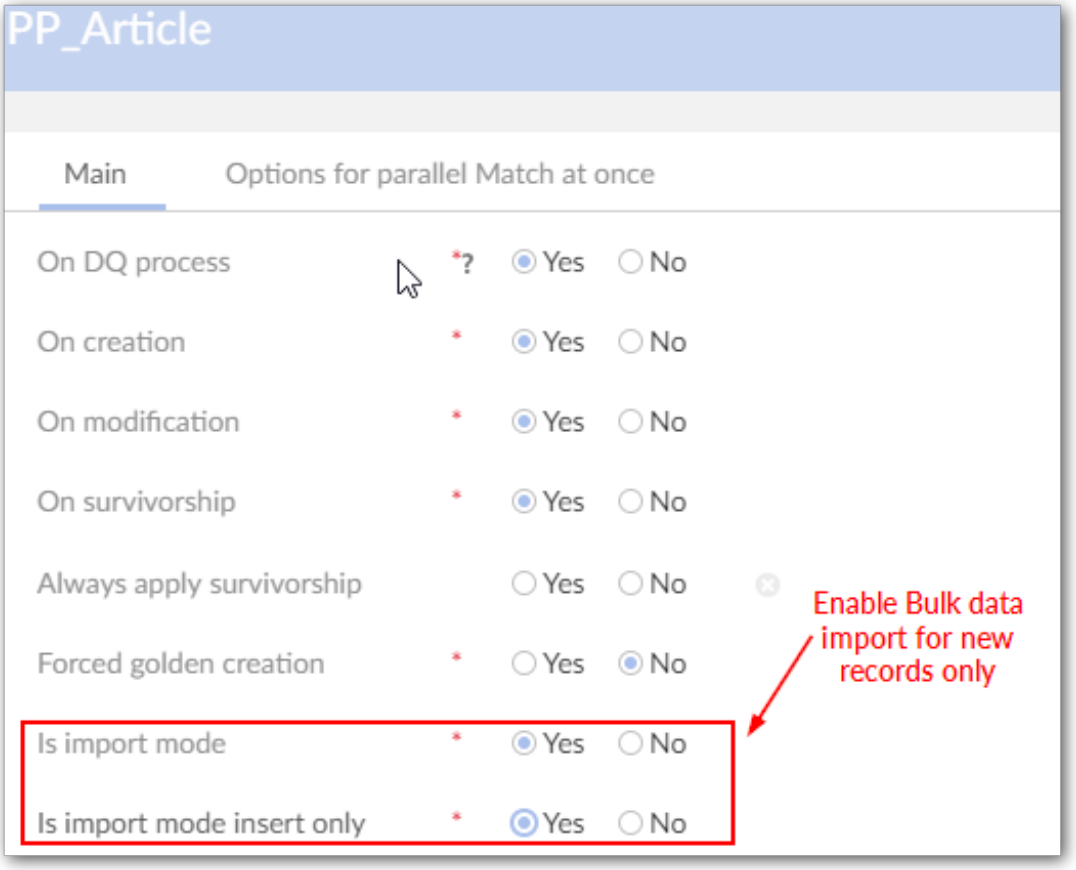
Set the 'On process driven' property and 'On simple matching' to 'Yes' to enable it.



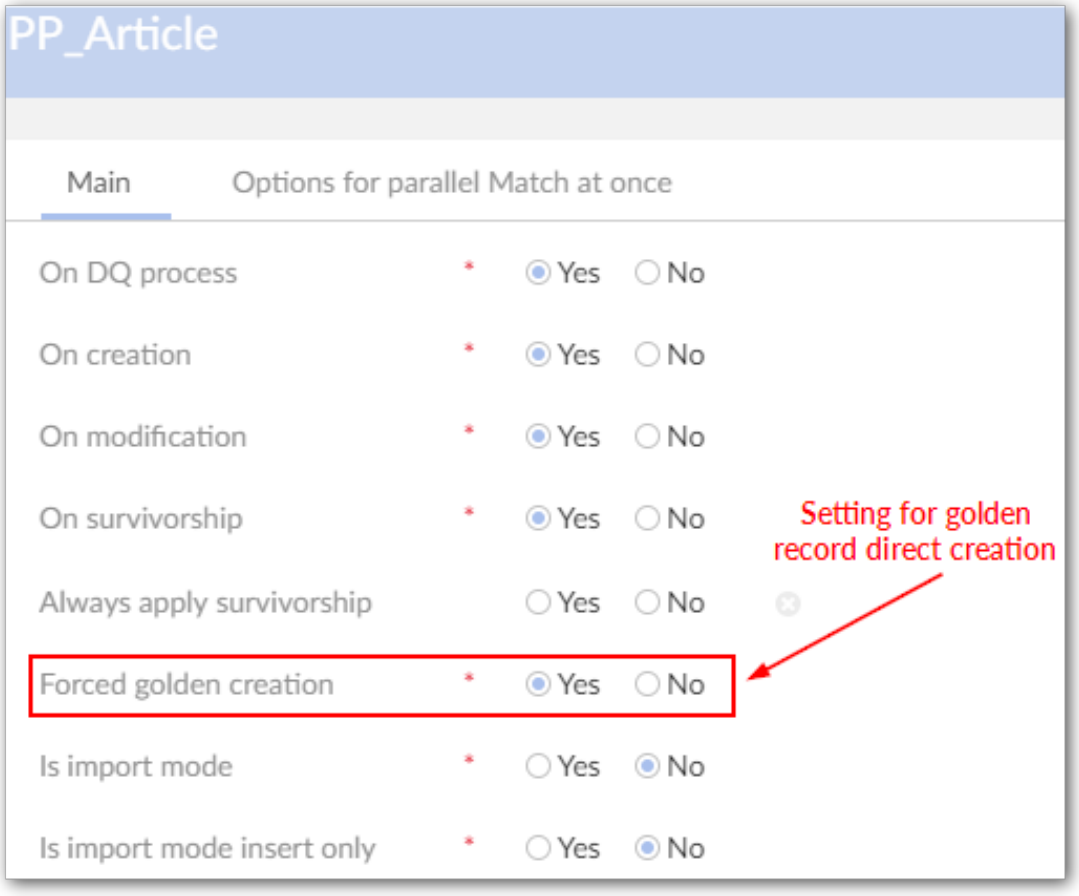
Bulk data import



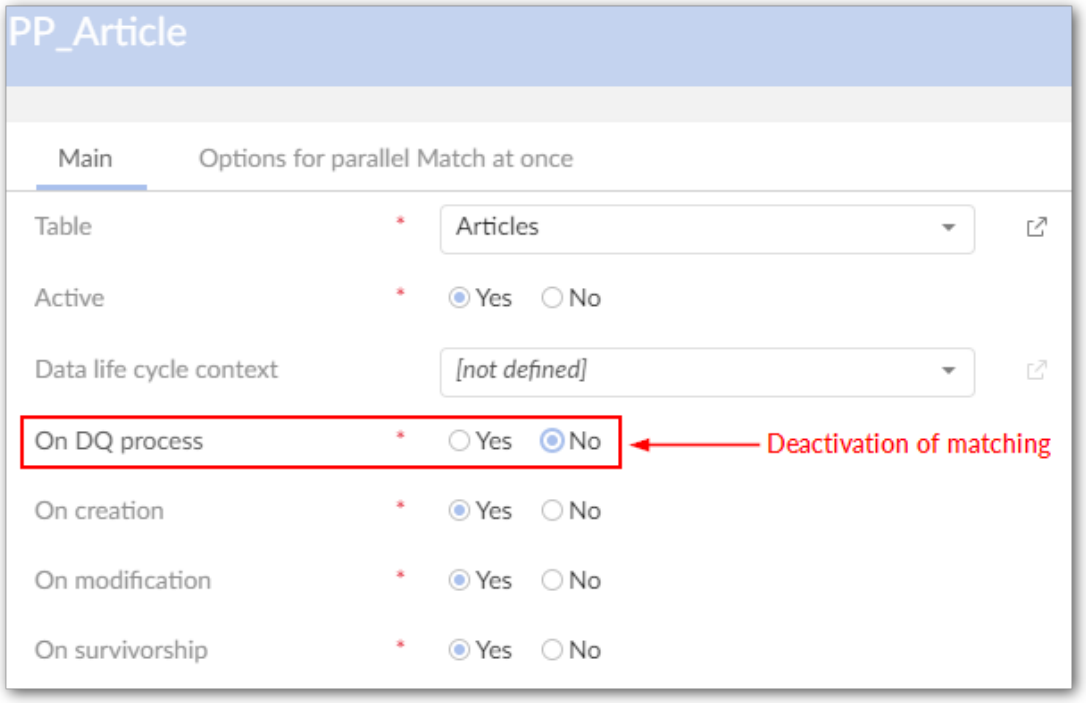
Bulk data import for new records only



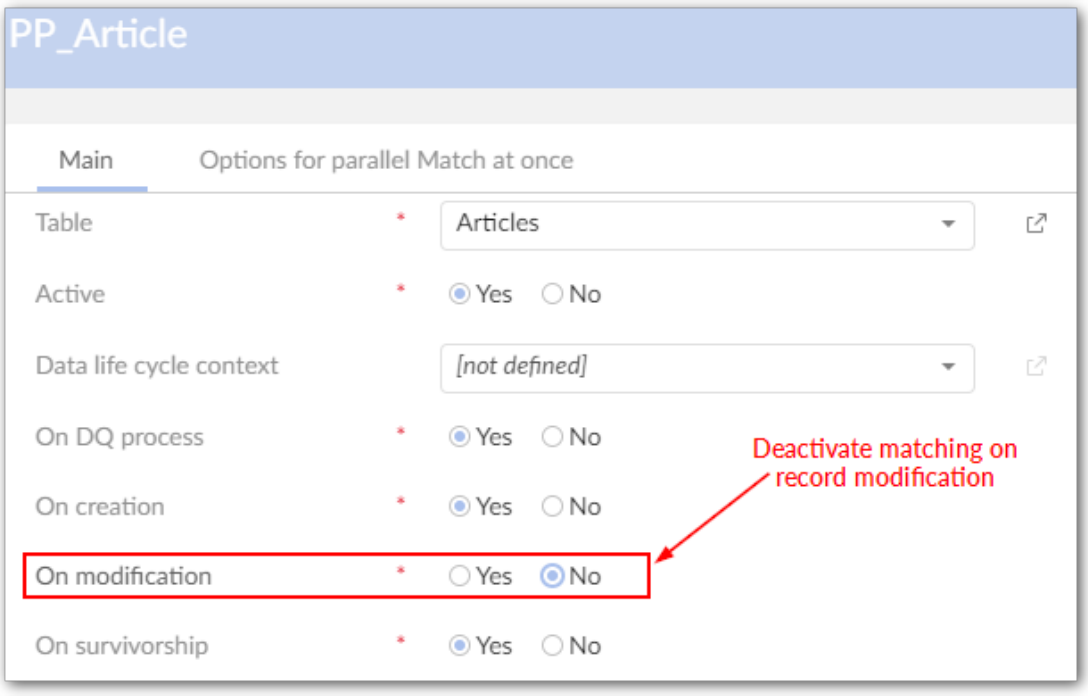
Direct creation of golden records (interactive and import)



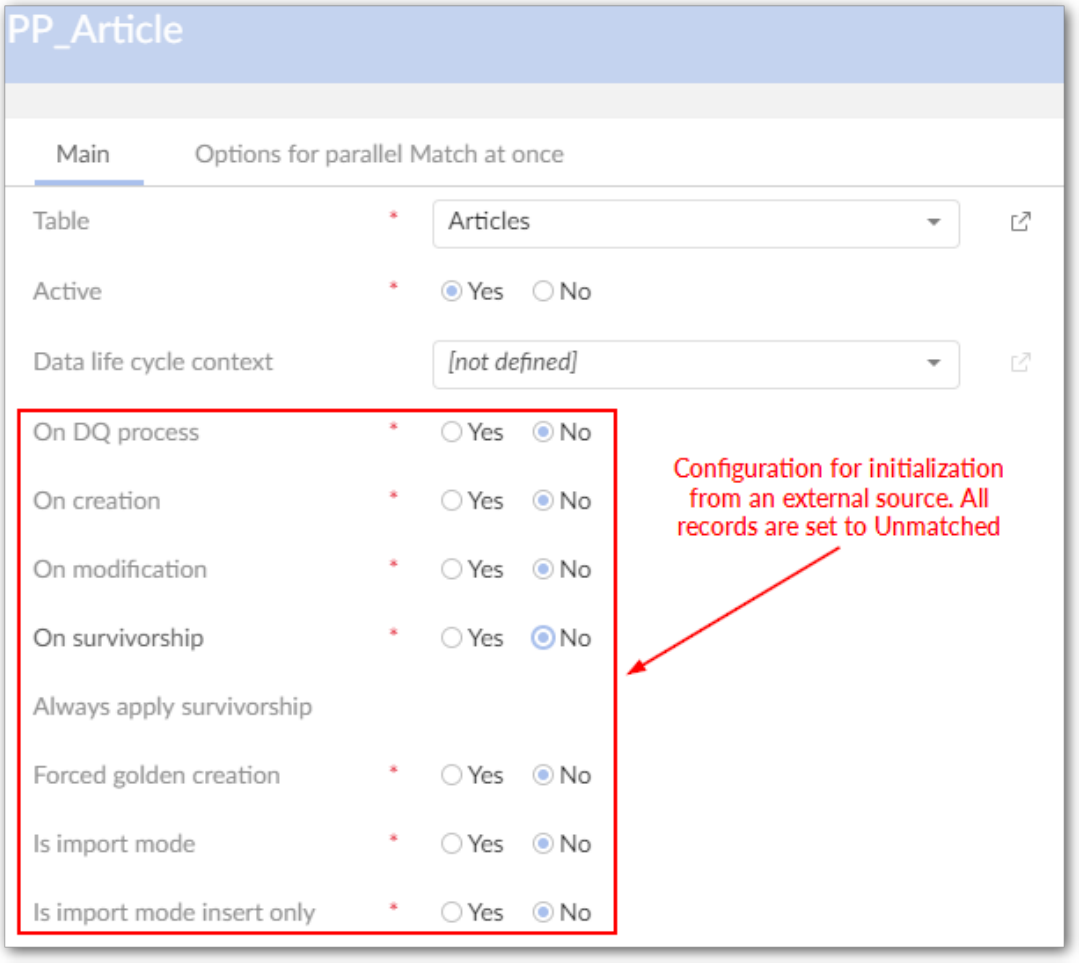
Deactivation of the matching



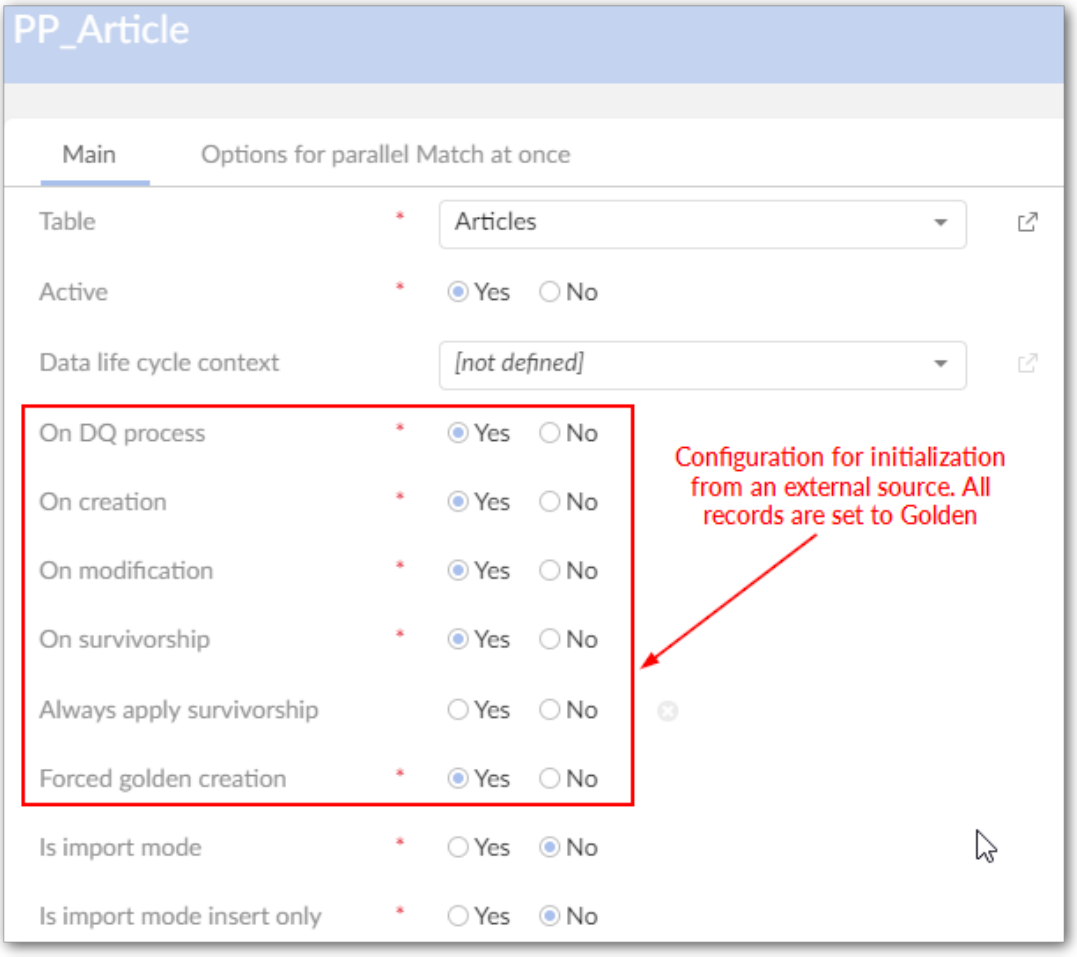
Deactivation of the matching when modification of records



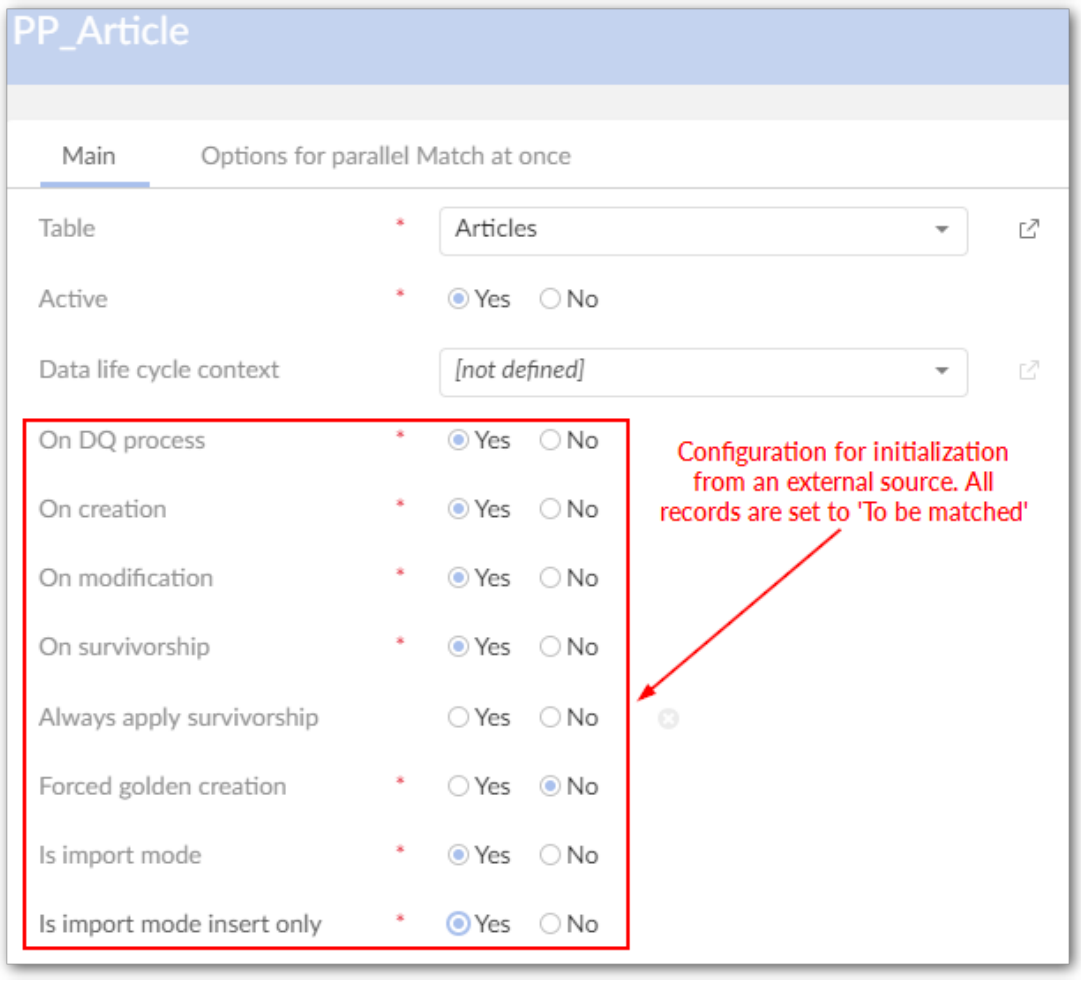
Initialization from an external source. All records are set to unmatched



Initialization from an external source. All records are set to golden

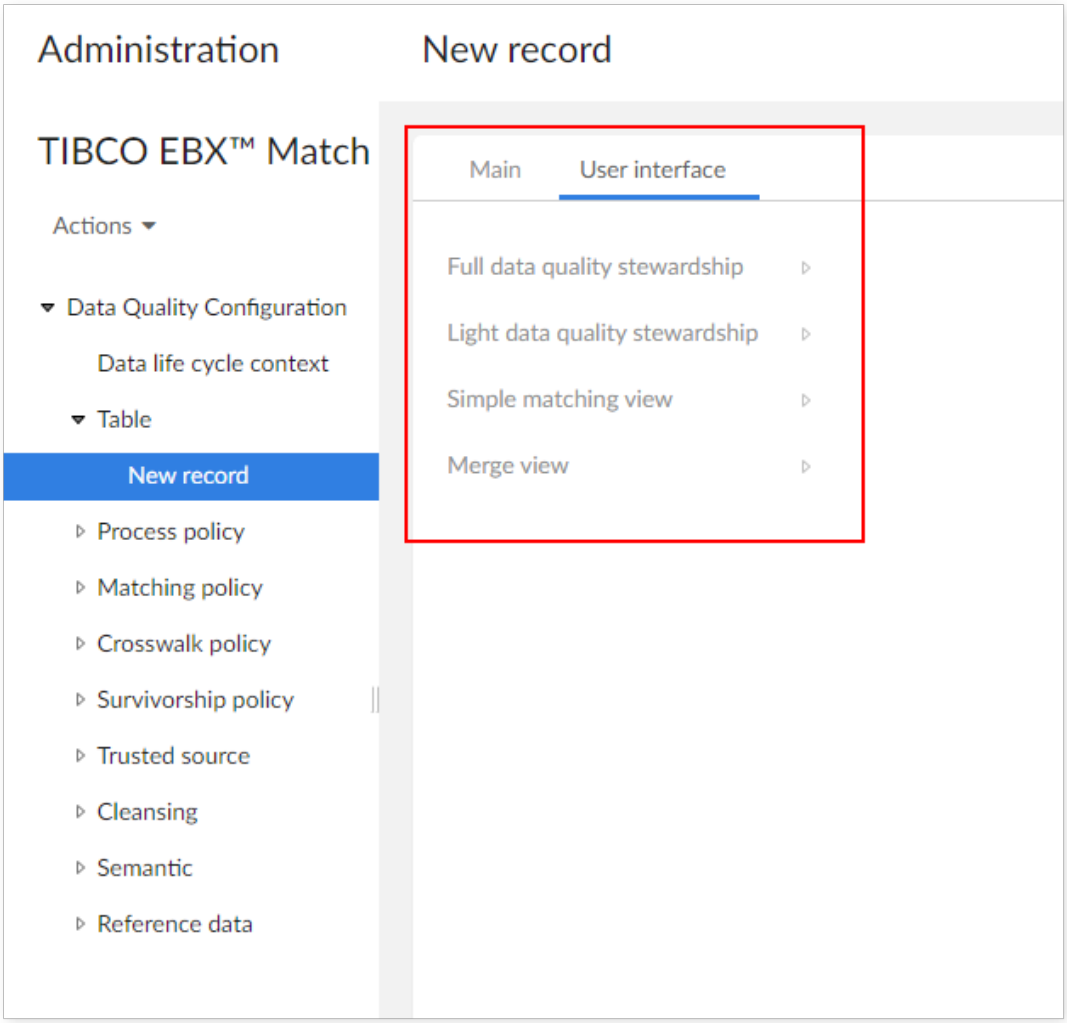


Initialization from an external source. All records are set 'to be matched'



User interface

In this section, you will find information on how to configure the user interface and detailed descriptions of all properties under the **User interface** tab. This tab allows you to regroup all user interface settings for a given table. They are organized by service available to the end user.



Properties	Definition
Full data quality stewardship	
Default view - top part	The add-on provides the matching view for the top part by default. You can customize it to fit your needs. To configure a custom view: • Create a custom view in EBX®. • Publish the custom view. • Navigate to the **User interface** table, fill in the **Default view - top part** property under the **Full data quality stewardship** group with the published name of your custom view. Your custom view must contain the **State** column displayed in order to offer all features necessary for the add-on. If it doesn't, the default table view will be applied. It is also a good practice to ensure the score and cluster columns display.

Properties	Definition
Default view - bottom part	The add-on provides the matching view for the bottom part by default. You can customize it to fit your needs. To configure a custom view: • Create a custom view in EBX®. • Publish the custom view. • Navigate to the **User interface** table, fill in **Default view - bottom part** property under the **Full data quality stewardship** group with the published name of your custom view. Your custom view must contain the **State** column displayed in order to offer all features necessary for the add-on. If it doesn't, the default table view will be applied. It is also a good practice to ensure the score and cluster columns display.
View by user profile	You can bind views to specific user profiles. This enables a user, or set of users to access specific matching views. To configure custom views for specific user profiles: • Create a custom view in EBX®. • Publish the custom view. • Navigate to the **User interface** table, fill in **View - top part** and **View - bottom part** properties with user profiles and the published name of your custom view. Your custom view must contain the **State** column displayed in order to offer all features necessary for the add-on. If it doesn't, the default table view will be applied. It is also a good practice to ensure the score and cluster columns display.
State(s) to filter from selection	You can hide any unwanted states from the **State** button on the top right corner of the matching view.
Light data quality stewardship	
Default view	The add-on provides the matching view by default. You can customize it to fit your needs. To configure a custom view: • Create a custom view in EBX®. • Publish the custom view. • Navigate to the **User interface** table, fill in **Default view** property under the **Light data quality stewardship** group with the published name of your custom view. Your custom view must contain the **State** column displayed in order to offer all features necessary for the add-on. If it doesn't, the default table view will be applied. It is also a good practice to ensure the score and cluster columns display.
View by user profile	You can bind views to specific user profiles. This enables a user, or set of users to access specific matching views. To configure custom views for specific user profiles: • Create a custom view in EBX®. • Publish the custom view. • Navigate to the **User interface** table, fill in **View** property under the **Light data quality stewardship** group with the published name of your custom view. Your custom view must contain the **State** column displayed in order to offer all features necessary for the add-on. If it doesn't, the default table view will be applied. It is also a good practice to ensure the score and cluster columns display.
State(s) to filter from selection	You can hide any unwanted states from the **State** button on the top right corner of the matching view.
Simple matching view	
Default view	The add-on provides the matching view by default. It can be changed for every table depending on the need. To configure a custom view:

Properties	Definition
	• Create a custom view in EBX®. • Publish the custom view. • Navigate to the **User interface** table, fill in the **Default view** property under the **Simple matching** group with the published name of your custom view. Your custom view must contain the columns **State** displayed in order to offer all features necessary for the add-on. If it doesn't, the default table view will be applied. It is also a good practice to ensure the **Simple matching score(%)** column displays.
View by user profile	You can bind views to specific user profiles. This enables a user, or set of users to access specific matching views. To configure custom views for specific user profiles: • Create a custom view in EBX®. • Publish the custom view. • Navigate to the **User interface** table, fill in the **View** property with the desired user profiles and the published name of your custom view. Your custom view must contain the columns **State** displayed in order to offer all features necessary for the add-on. If it doesn't, the default table view will be applied. It is also a good practice to ensure the **Simple matching score(%)** column displays.
Available actions	Determines availability of options in the simple matching view.
Merge view	
Default view	The add-on provides the merge view by default. It can be changed for every table depending on the need. To configure a custom view: • Create a custom view in EBX®. • Publish the custom view. • Navigate to the **User interface** table, fill in the **Default view** property under the **Merge view** with the published name of your custom view. Your custom view must display the primary key columns in order to offer all features necessary for the add-on. If it doesn't, the default table view will be applied. It is also a good practice to ensure the source column displays (if applicable).
View by user profile	You can bind views to specific user profiles. This enables a user, or set of users to access specific merge views. To configure custom views for specific user profiles: • Create a custom view in EBX®. • Publish the custom view. • Navigate to the **User interface** table, fill in the **View** property with user profiles and the published name of your custom view. The **User profile** property lists all profile and user available and **View** is a string parameter to fill a view ID. The mechanism is the same as the one used in the process policy settings for views. Your custom view must display the primary key columns in order to offer all features necessary for the add-on. If it doesn't, the default table view will be applied. It is also recommended that you display the source columns (if applicable).
Apply EBX permission	Determines whether EBX® permissions apply. If a user does not have permission to view a field, it does not display. However, it is good to keep in mind that users in charge of merging must have access to all record fields.
Allow the pivot to be modified	Determines whether users can change the Pivot record in the **Merge view** by selecting its primary key.
Merge option	Defines the options available to merge records manually. The options are described below: • **Set Golden** will set the consolidated record as a golden record. It will participate in the future matching operations.

Properties	Definition
	• **Set definitive golden** will set the consolidated record as a golden record. It won't participate in the future matching operations. • **Both** The user will have the choice between both options at the end of the merge process. Default value: **Set Golden**

Table 7: User interface

Matching policy

A matching policy is applied to a table and a matching context (optional) in order to declare how the add-on executes matching. For example, a matching policy can define which fields are used to compute the matching score and which algorithm is used.

To configure a matching policy:

- Create a record in the 'Matching policy' table and specify a table and which algorithms you want to apply.
- Create a record in the 'Matching field' table that determines which field the add-on uses during matching.
- A matching context is required when the fields to match depend on the nature or type of records.

Below is an example with a table named 'Party', structured as follows:

Field	Definition
Code	Any naming convention without white spaces
Organization type	{Company, Individual}
Nature	If Organization type = Company: {Domestic, Overseas} If Organization type = Individual: {Prospect, Customer, Employee}
Name	
Address	
Country	For Overseas company only
Social number	For Employee only
Sales representative	For Customer only

Table 8: Example: 'Party' table

For this table, depending on the values of Organization type and Nature, specific fields are used to drive the matching process. The matching policy contexts needed are as follows:

Matching policy context		Matching policy (fields used for matching)
Organization type	Nature	
Company	Domestic	Name, Address
Company	Overseas	Name, Country
Individual	Employee	Name, Social number
Individual	Customer	Name, Sales representative
Individual	Prospect	Name, Address
Null	Null	Name

Table 9: Example of matching policy contexts for the table 'Party'

In this example, six different matching policy contexts are defined and associated with five different matching policies: (the policy '(Name, Address)' is used twice).

The add-on relies on three tables to manage the configuration of matching policies:

- Matching policy context
- Matching policy
- Matching field

The relationships between these tables are as follows:

- A matching policy is linked to the table for which matching is configured (mandatory) and to a matching policy context (optional).
- A matching field is linked to a matching policy (mandatory).

You can also define contexts per dataspace and dataset to execute matching policies depending on the data life cycle. This configuration is done directly in the 'Matching policy' table.

Matching policy configuration

In this section, you will find information on how to configure Matching policy and detail description of all properties under Matching policy group.

> **Note**
>
> You can test a new or existing matching policy configuration to ensure it returns the desired results. See Testing a matching policy [p 78] for more information.

Matching policy context

Matching policy context table (logical name: **MatchingPolicyContext**).

Properties	Definition
Code	Any naming convention without white spaces can be used.
Table	A link to a table under the add-on's control.
Name	Name of the matching policy context.
Field contexts	Field context: a field in the table used as the context. Use foreign key value: when enabled, the foreign key's value is used instead of its label. This option only displays when the chosen 'Field context' is a foreign key field. Value: value of the field. The value is case sensitive. • For an empty value use the following constant: osd:is-empty • For a non-empty value use the following constant: osd:is-not-empty. In this case, if the field value is not null or not empty, the matching policy context is applied. A multi-valued field cannot be selected as a field context. To use such a field, you need to create a new field with a function that aggregates the values into the excepted format. Then, this field can be used as a field context. When a foreign key field is defined, the 'Default label' is used to compare by default. If there is no 'Default label' set, 'Programmatic label' is used. In case neither 'Default label' nor 'Programmatic label' set, the field value is considered empty.
Context Java class	A Java class you define to extend the MatchingPolicyContextChecker.java API. • When Field contexts are defined and Context Java class is not defined: The add-on uses context based on Field contexts • When Field contexts are not defined and Context Java class is defined: The add-on uses context based on Java class • If both of them are defined: The add-on uses context based on Java class. But inside Java class, users can get the list of Field contexts defined.

Table 10: Matching policy context properties

The use of matching policy contexts is optional. A matching policy can be configured without a context (see field 'Is context') in the Matching policy table below.

Any number of fields can be defined as part of a context. Fields are combined using the 'and' operator. Defining a context with only one field is possible. A 'field context' can be a value function that dynamically computes a record's context based on business rules. For instance, depending on a list of countries, the 'Language' context field can be computed automatically.

Matching policy

Special Notation:	
✔	The configuration for a table must declare one active matching policy with no context. When several matching policies are active for different contexts, then the policy with no context is used by default for every execution not related to a context.

Matching policy table (logical name: **MatchingPolicy**)

Properties	Definition
Code	Any naming convention without white spaces can be used.
Short description	A description of the policy.
Long description	Long description of the policy.
Table	Table for which this policy is defined.
Active	If 'Yes': This matching policy is used. If 'No': This matching policy is not used. This field indicates whether or not the policy will be used when running matches. Since the add-on manages the matching policy context, it means that one or more matching policies can be active at the same time for a given table. To disable a matching policy, set the active field to 'No'. Only one matching policy can be active for each matching context. One active matching policy with no context must be configured. It is used by default matching at execution time when a matching execution has no relation to a defined context.
Is forced to void	If 'true': The matching policy short-circuits upon execution. This allows you to have an active matching policy for which matching does not execute. If 'No': The matching policy executes per its configuration. When set to 'No', the property 'Is forced to void' has no effect on normal matching execution. Default value: 'false'
Funneling matching	If 'Yes': The funneling matching mode is activated. If 'No': The funneling matching mode is deactivated. Default value: 'No' • By default the add-on computes the matching score by applying a weighted average on the field scores that participate in the matching policy. When a score applied to a field is lower than the 'Stewardship min score' (process policy) then it still participates in the average computation and its weighting is still used. • When the 'Funneling matching' option is used, each field is associated to a 'Field stewardship min score' (refer to Matching field table). If one of the fields does not reach this threshold then the matching is canceled. When each field is scored with a sufficient value, then a weighted average score is computed.
For match at once	The matching policy used for 'Match at once' operations. A table can have only one matching policy active at a time for this type of operation. If no 'For match at once' policy is specified, then the default policy is used by 'Match at once' operations.
Modify Merged without match	When a Merged record is modified with 'Merged record is recycled' activated, you can decide whether to re-run the 'Match table' operation. If the target record (Pivot or Golden) is auto-created: Survivorship applies to merge data from all Merged records which have the same target record and cluster as the modified Merged record. The 'No match when modifying Merged record' option is ignored by the add-on. If 'Yes': When the Merged record and its target (Pivot or Golden) are in the same cluster, the merged retains in the current cluster and the Survivorship is applied to merge data from all Merged records in the cluster to the target record, 'Is forced to void' is ignored. Otherwise, the Merged record is untouched. If 'No': The add-on keeps the current behavior when 'Merged record is recycled' is activated. Default value: 'No'

Properties	Definition
For group at once	The Matching policy used for 'Group at once' operations. A table can have only one matching policy active at a time for this type of operation. If no 'For group at once' policy is specified, the default policy is used by 'group at once' operations.
For search before create	The matching policy used for the 'Search before create' feature. A table can only have one active matching policy with 'Search before create' enabled at a time. If the 'For search before create' policy is not specified, the default policy will be used.
Keep not matched records untouched	Determines whether records keep their current state after a matching operation produces no matches. This setting applies to the following 'Match at once' services: Match at once, Match at once full mode, Parallel match at once and Exact match at once. If set to 'Yes': The Unmatched, To be matched, and Suspicious records remain in their current state when no matches are found. If set to 'No' or not defined: Records that do not have any matches become Golden in cluster 01. Default value: 'No'
Automatically create new golden	If 'Yes': The add-on automatically creates a new golden record when it identifies a positive match between records. This property is only applied in the case of record creation, modification or a 'Match table' operation. It is used only if the survivorship property is 'On' and if survivorship rules are defined. The process used by the add-on for golden creation is as follows: • From the matched records, the add-on selects a pivot record based on the survivorship policy's **Survivorship selection mode** value. • The pivot record moves into the suspect state and the merge process executes. • The golden record is obtained by merging all suspect records. If 'No': A new golden record is not automatically created when there is a match. Default value: 'No' Note that if there is already an 'Auto-created' record in the cluster, that record will be chosen as the new golden.
Auto create new golden in match at once	If set to 'Yes': The 'Automatically create new golden' service applies when running a 'Match at once' service including 'Exact match at once' and 'Exact match at once in memory'. It is used only when you set the survivorship property to 'On' and define survivorship rules. If set to 'No': The 'Automatic create new golden' is not applied when running a 'Match at once' service. Default value: 'No' Note that if there is already an 'Auto-created' record in the cluster, that record will be chosen as the new golden.
Auto create new golden for single golden	If set to 'Yes': When a match results in the creation of a single Golden, the add-on duplicates the record and moves both records into a new cluster. The auto-created record becomes Golden, and the former single Golden becomes Merged targeted to the new auto-created record. The newly created Golden record has the same behavior as other records as it is modified or under a matching operation. This property is applied: • On record creation and modification. • On 'Match table' and 'Match at once' operations. • When running 'Set golden' on Suspicious, Unmatched, Suspect and Pivot records. • When running 'Remove and set golden' on Suspect records. • When running 'Set back golden' on Suspect and Pivot records. If set to 'No' or not defined: The 'Automatic create new golden' is not applied. Default value: 'No'

Properties	Definition
Customize source value for new golden	If 'Yes': When the 'Automatically create new golden' property is activated, the value specified by the 'Source value' property updates the source field's value in the newly created golden record. The 'Source value' property is configured in the matching policy. If 'No': The source field value in the newly created golden record isn't updated. Default value: 'No'.
Source value	A source identifies a system that provides data either at the record or field level. The source is selected from the 'Source' table.
Is context	If 'Yes': This policy is executed for a matching context defined by the link to the 'Data life-cycle context', the 'Matching policy context' table and/or the 'Workflow context'. When several matching policies are candidates for a composite context based on two or three values among 'Data life-cycle context', 'Matching policy context', and 'Workflow context' then a priority is applied as follows, based on a weight average: 'Data life-cycle context' = '2', 'Matching policy context' = '1', 'Workflow context'='4'. The selection procedure is as follows: • Look up all active matching policies for the table with 'Is context'='Yes'. • Filtering by context workflow value if not undefined. • Filtering the previous list by dataspace / dataset if not undefined. • If it rests more than one matching policy at this step, meaning that there is at least one with a matching context. • Then try to use a matching context that fits with the record values. • If OK then the matching policy is found. • If OK either it still remains a matching policy with matching context is undefined and then the add-on uses it. • Or this is impossible to find a matching policy and then the default one must be used ('is context='No' and 'Active'='Yes'). If 'No': This matching policy is not related to any matching context. Default value: 'No'
Data life cycle context	A matching policy can be defined for a specific dataspace and/or dataset. This property is a link to a data life cycle context (dataspace and/or dataset). An undefined value deactivates this context.
Matching policy context	Matching policy context for which this matching policy is defined. An undefined value deactivates this context.
Workflow context	When executing a workflow you can define a property (cf. Workflow User tasks) that specifies execution of a specific matching policy. Then, the 'Workflow context' property is set to the same value as the one used in the user task. An undefined value here deactivates this context.
Pivot record selection mode	This rule is applied by the add-on (Match table and Match at once operations) to decide which of the suspect records is the pivot record (see *Record selection policy* section for further information on the different strategies that the add-on can apply). This policy is disabled when 'Golden is preserved for selection' or 'Automatically create new golden' is applied. Default value: 'New update' Note that auto-created records are given the highest priority.
Golden is preserved for selection	If 'Yes': The Pivot record selection mode will prioritize a record identified as 'Golden' or 'was golden' if it exists in the cluster. If many records with the 'Golden' or 'was golden' identifier exist, the 'Most

Properties	Definition
	recently acquired' record is used. If there is no record with 'was golden' marker or no 'Golden' record, then the 'Pivot record selection mode' executes normally. This property is used to select the pivot record in case of record creation and modification only. It is not used with the 'Match table' and 'Match at once' operations. If 'No': The 'Pivot record selection mode' does not use the 'was golden' indicator or 'Golden' state. Default value: 'No' A record is identified as 'was golden' when its state was golden but has been changed because of a new matching.
Main matching algorithm	The default algorithm for matching all fields that use this matching policy. If required, the matching policy algorithm can be changed at the field-level (see table Matching field). You can refer to the *Matching strategies* section for information that will help you decide which algorithm to use. The selection of the main algorithm is mandatory.
Second matching algorithm	The algorithm used to manage records that could be false negative results after main algorithm execution. It is used after the first matching procedure to ensure that no suspect records have been missed. It is not possible to override this matching algorithm at the field-level. You can refer to the *Matching strategies* section for information that will help you decide which algorithm to use.
Ignore case	If set to 'Yes': All algorithms and the following 'Exact matching' operations ignore case sensitivity: - Filter by - Business ID - Exact match at once - Filtering field rule - Synonym If set to 'No': All algorithms and 'Exact matching' operations will be case sensitive. Default: 'No'
Exclude records from matching	If you want matching to ignore records when a field contains specific values: • Specify the field you want to monitor using the 'Field to exclude records from match' property. You can do this at the 'Table' configuration level. • Use this property ('Exclude records from matching') to define which field values identify records the matching operation should ignore. When one of these values is equal to the field value of record then the record is ignored by the matching operation. When a record is created and has excluded value, then its state moves to 'To be matched'. When a record is modified and has excluded value, then its state is not modified. To declare an empty value, use the following constant: osd:is-empty
Filtering record rule	A business rule can be applied to filter records that must be excluded when matching against the pivot record. This feature is used when the matching policy's 'Exclude record from match table' property is not sufficient (based on a direct equal value of string). The applied rule is configured in the 'Filtering record rule' table. The bespoke parameters group of fields passes parameter values to a rule when needed. Then, the rule needs to be able to manage these parameters. When a record is created and matches with the filtering rules, then its state moves to 'To be matched'. When a record is modified and matches with the filtering rules, then its state is not modified.
Filter by	When matching a large volume of records, perhaps numbering in the millions, you can improve fuzzy match response time by using a fast filter that returns a subset of records to match. The filter used a 'Exact' query. The fuzzy match is then applied on this subset of records only. You can combine multiple fields in the filter, the 'And' operator is then applied. To guarantee the best performance, you must declare the filter field(s) as index (see EBX® documentation). The 'Match at once' operation uses the

Properties	Definition
	'Filter by' property. However, the 'Exact match at once' operation does not because it acts as a filter on its own.
Handle null value for filter by	• [Not defined]: If source and target values are null, comparison results in a value of true. Comparison results in a value of false if either source, or target are not null. • [If source and target values are null, comparison results in a value of false.] Comparison still results in a value of false if either the source, or target are not null. • [If source's value is not null and target's value is null, comparison results in a value of true.] Comparison results in a value of false if the source is null and the target is not null. Comparison results in a value of true if the source is not null and the target is null. Default value: [Not defined]
Literal score	'Exact' value is available. When using the 'Exact' matching policy, this property means that two records are considered as potential duplicates when at least one business identifier field value is equal. This business identifier is defined in the 'Table' (See *Matching policy with exact score* example in the *Matching policy* section.) configuration.
No match records when same source	If 'Yes': Two records from the same source will never match. You can use this option if the 'Source field' is configured at the 'Table' level. The add-on checks to ensure no records from the same source exist in the target cluster prior to moving a record. If there is any record from the same source detected, the record remains in its state in the current cluster. Source field value of suspect records will never be merged into the pivot record. If 'No': The matching is achieved in a normal way when the records come from the same source that is based on the 'Source field' configuration at the 'Table' configuration level. Default value: 'No'
Narrow search	Activate this option to reduce the memory usage for matching. When this option is activated for matching policies that have low stewardship min scores, only records that are closely similar are included during matching. Default value: 'No'
Threshold matching	These threshold matching values are not mandatory. They are used instead of the corresponding values that are defined at the process policy level. If one field is provided then the add-on will use all of them rather than the values specified at the process policy level. In this case, the four values become mandatory. Stewardship min score (%): If undefined, the value configured at the Process policy level is used. When a match score is lower than this threshold then the add-on does not consider the record to be a suspect record. Stewardship max score (%): If undefined, the value configured at the Process policy level is used. When a match score is higher this threshold, the add-on enforces an automatic merge of this record into the pivot or golden record. This procedure is also known as 'survivorship'. Above this value, the record is merged automatically. It is important to select this value carefully, as if it too low, records may be automatically merged with pivot and golden records that are not actually related. When scores fall between the 'Stewardship min score' and the 'Stewardship max score' values, a user decision is needed to determine how to manage the suspect record (stewardship process). Threshold second matching(%) If undefined, the value configured at the Process policy level is used. When the score computed by the first algorithm falls between 0% and the specified 'Threshold second matching(%)' then the second algorithm is applied to detect potential false negative records.

Properties	Definition
	For example, if 'Threshold second matching' is set to '5%' then the second algorithm is executed each time the score of the first algorithm is lower than '5%'. Constraint: 'Threshold second matching' is lower than or equal to 'Stewardship min score'. This property is used when the 'Second matching algorithm' property in the matching policy is not empty. Default value: '0' Second level stewardship min score (%) If undefined, the value configured at the Process policy level is used. When the second level of matching is configured to seek false negative records from the first matching, then this threshold is used to evaluate if the matching score provides a suspect or not. This property is used when the 'Second matching algorithm' property in the matching policy is not empty.
Matching through a relation	
Use matching through a relation(s)	'Relation matching' allows the add-on to match data through existing table or join table relationships. If 'Yes': For the matching operation, the add-on combines the table on which this policy is configured with any table related to it either by join or by relationship. If 'No': The relational matching configuration is ignored. Default value: 'No'
Relation record score weight	This property allows you to configure the matching score weight that is computed for the 'Relation table'. The matching score obtained for the 'Relation table' is combined with the score computed on the 'Table to match' through a weighted average. Default value: '1'
Relation record stewardship min score	This is used when the 'Funneling mode' property in the matching policy is active. If the score of the field is lower than the 'Field stewardship min score' then there is no match. By default the value is set to 100 to make the threshold inactive. Default value: '100'
Use join table	If 'Yes': The relation match is configured through a join table between the 'Table to match' and the 'Relation table'. If 'No': The relation match is configured with a direct relation between the 'Relation table' and the 'Table to match'. Default value: 'No'
Relation match with join table	Join table • Selection of the join table that links the 'table to match' with the 'relation table'. Link: Join table → Table to match • Selection of the foreign key to use for the relation between the join table and the 'table to match'. Relation table • Selection of the 'relation table'. Link: Join table → Relation table • Selection of the foreign key to use for the relation between the join table and the 'relation table'. Matching policy on Relation table • Selection of the matching policy to apply on the 'relation table'.

Properties	Definition
Relation match without join table	Relation table • Selection of the 'relation table' Link: Relation table → Table to match • Selection of the foreign key to use for the relation between the 'relation table' and the 'table to match' Matching policy on Relation table • Selection of the matching to apply on the 'relation table'

Table 11: Matching policy properties

Matching field

Configuration of the fields used in matching execution time.

Matching field table (logical name: **MatchingFieldRule**)

Properties	Definition
Code	Any naming convention without white spaces can be used.
Matching policy	Reference to the 'Matching policy' table.
Field	A field in the table that is analyzed by the referenced matching policy. This field is used during the matching procedure. If the field value is empty at execution time, then matching is not enforced. It is possible to match inside a list. Two lists of values will match when at least one value matches the two lists. For instance (John, Carl, Paul) will match with (Frank, Theo, John).
Foreign key	A list of foreign key fields. You define a path to another field—an alternative to the one defined by the 'Field' property—to execute matching on by creating hops to navigate the relationships. The add-on uses the value from the last hop during the matching procedure. The foreign key 'Default label' is used to compare by default. If there is no 'Default label' set, 'Programmatic label' is used. In case neither 'Default label' nor 'Programmatic label' set, the field value is considered empty.
Handle null value matching	Provides null value matching strategies using the following options: • [Not defined]: The score is calculated base on the algorithm. • If both values are null, score is 100%, 0% otherwise. • If both values are null, score is 0%, ignore otherwise. • If one of the values or both are null, ignore. • If one of the values or both are null, returns 0%, In case the returned matching result is 0%, the Surrogate field (if exists) is used to match instead of the configured matching field. Note that an undefined foreign key matching field will be considered as null, and this property does not support multi-hop foreign key.
Matching algorithm	By default, the algorithm used is the one defined as the main algorithm in the table's matching policy. This default value can be changed to select another algorithm.
Score weight	When several fields of a source table are used for matching, the score weight is used to give a weight to each field score that the add-on uses to compute a weighted average. Example: '0.5' means the score of the field is worth half (50%). If a matching field is empty then the weighted average does not take into account this field. Default value: '1'
Score calculation	Defines how the score is calculated: - Real score: The exact score returned by the algorithm. - Normalize score: The score will be the maximum score of the maximum stewardship defined minus 0.1. This strategy can be used to avoid auto merge of records when they are not strictly identical. - Default value: Real score
Field stewardship min score (%)	Used when the 'Funneling mode' property in the matching policy is active or in a surrogate field matching. - When the 'Funneling mode' property is active: If the score of the field is lower than the 'Field stewardship min score' then there is no match. By default the value is set to 100 to make the threshold inactive. - If the field score is lower than 'Field stewardship min score', surrogate field (if exists) is used to match instead of the configured matching field.

<table>
<tr><th>Properties</th><th colspan="2">Definition</th></tr>
<tr><td>Filtering field rule</td><td colspan="2">A business rule can be applied to filter the field value in order to ignore characters such as N/A, Inc., *, $, etc. before the match execution. A rule can perform any type of filter. One example is removing all figures in a string. The filter is applied to the pivot record and the records involved in the match. It is performed in memory and does not change the actual value of the data in the repository. Only string fields can be filtered. Foreign key values cannot be filtered. The bespoke parameters group of fields is used to pass parameter values to a rule when needed. Then, the rule must be able to manage these parameters. The filtering field rule must depend on the record's fields.</td></tr>
<tr><td>Use synonym group</td><td colspan="2">A group of synonyms can be selected. If a field value in the suspect record does not match with the pivot, then a new matching is achieved with the synonym values rather than the initial value in the suspect field. The first positive match is used to get the final score.
When values are the same with any case difference, the score will be max score - 0.1.</td></tr>
<tr><td>Check synonym in all groups</td><td colspan="2">If 'true': When matching uses the synonym mechanism, it searches through all child synonym groups.
If 'false': When matching uses the synonym mechanism, it searches through each separate child synonym group.</td></tr>
<tr><td rowspan="3">Smart synonym matching</td><td colspan="2">When matching uses the synonym mechanism, it checks whether strings from the specified field match with values in the list of synonyms. If a value matches, the add-on replaces it with the shortest value before running match. If it matches 100% then the record score is equal to the maximum score -0.1. When there are several synonyms with the same length, the first value in the list will be taken to replace for other synonyms.</td></tr>
<tr><td colspan="2">Use smart synonym matching:
If set to 'Yes': The Smart synonym matching is applied.
If set to 'No': The Smart synonym matching is not applied.
Default value: 'No'</td></tr>
<tr><td colspan="2">Ignore case:
If set to 'Yes': Ignores case when Smart synonym matching is activated.
If set to 'No': Smart synonym matching is case sensitive.
Default value: 'No'</td></tr>
<tr><td rowspan="3">Surrogate fields</td><td colspan="2">Defines the fields and comparison mode applied in a matching when the returned score on the configured matching field is lower than the 'Field stewardship min score'.</td></tr>
<tr><td>Field</td><td>List of fields in the matching table with the same data types which can be configured as alternative matching fields.</td></tr>
<tr><td>Comparison mode</td><td>• First match(default): When a surrogate field matches the pivot's field with a score higher than 'Field stewardship min score'. Surrogate field match returns this score.
• Best match: When all surrogate fields have been compared and at least one matches the pivot's field with a score higher than 'Field stewardship minscore'. Surrogate field match returns the best score among all matched surrogate fields.</td></tr>
</table>

Table 12: Matching field properties

Special notation:	
✕	Configuration of the weighted average to take into account an empty field.

Matching algorithm

The add-on provides several predefined algorithms.

Matching algorithm table (logical name: **MatchingAlgorithm**)

This table is located under the EBX® Administration Tab in the 'Matching reference data' dataset.

Properties	Definition
Name	Algorithm name (Eg. DoubleMetaphone, FuzzySearch, Levenshtein, etc.)
Description	Context of use.
Is prebuilt	If 'Yes': The algorithm is provided by the add-on. If 'No': The algorithm is not provided by the add-on.
Is percentage score	If 'Yes': The algorithm provides a similarity score. If 'No': The algorithm provides a binary score (Match=1, Unmatch=0). When the score is equal to Match, then the add-on systematically overwrites the score to 'Stewardship min score + '1' '.
Java class	Specifies a Java reference to provide a custom algorithm.

Table 13: Matching algorithm properties

Predefined algorithms

The add-on provides these predefined matching algorithms.

Algorithm	Default parameters	Description and Parameter configuration (if applicable)
Chinese		Allows matching in Chinese.
Double metaphone	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. 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: 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. 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.
Double metaphone 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. 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. 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.
Exact		This algorithm returns a matching score of 100% for empty field values. Even though this algorithm runs an exact match on the specified field(s), other fields can use different matching algorithms. This means one matching operation may include multiple algorithms. If you want to improve response time by executing a purely 'Exact' match, refer to the matching

Algorithm	Default parameters	Description and Parameter configuration (if applicable)
		policy's 'Filter by' property. Alternatively, you can use the 'Exact match at once' service.
Full text		This algorithm finds a non-case sensitive, exact match of the entered keyword in data of longer strings.
FuzzyFullText	Similarity = 0.7 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. 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. 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.
FuzzyJapanese	Similarity = 0.7 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. 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. The **Prefix length** parameter specifies the number of characters-from the beginning of the search term-that must exactly match in order to return a result. For example: The keyword '####' will match '#####' if the **Prefix length** < 4 and **Similarity** = 0.
FuzzyRussian	Similarity = 0.7 Prefix length = 0	Performs a search on Russian text and finds a similar, or "fuzzy" match. The **Similarity** parameter defines a value between 0 and 1 to set the required similarity between the query term 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.

Algorithm	Default parameters	Description and Parameter configuration (if applicable)
		The **Prefix length** parameter specifies the number of characters-from the beginning of the search term-that must exactly match in order to return a result.
Japanese		Performs search on Japanese text. This algorithm allows you to use the following character types or any combination thereof: Kanji, Katakana and Hiragana.
Jaro Winker	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.
Levenshtein		This algorithm works best for short strings where you expect few differences between the keyword and the data being searched. For example, this works well for dialects spoken in a particular part of the country, or by a specific group of people.
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. 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.
Russian		This algorithm allows you to search text in Russian.
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. When you increase the **Threshold** parameter's value, the score decreases: Score = 100-(distance*100/threshold)

Algorithm	Default parameters	Description and Parameter configuration (if applicable)
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. 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).
Soundex		This phonetic algorithm works best on proper names. It returns similar-sounding words or names by converting the string being searched to a four-character code and returning words with the same code. Note that you cannot use this algorithm to search numeric, date/time, or special characters.
Korean		Perform search on Korean text.
FuzzyKorean	Similarity = 0.7 Prefix length = 0	This algorithm performs a search on Korean text and finds a similar, or "fuzzy" match. 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. The **Prefix length** parameter specifies the number of characters-from the beginning of the search term-that must exactly match in order to return a result

Table 14: Predefined algorithms

Filtering

Filtering record rule

Filtering record rule table (logical name: **FilteringRecordRule**)

This table is located under the EBX® Administration Tab in the 'Matching reference data' dataset.

Properties	Definition
Name	Name of the filtering record rule.
Description	Context of use.
Java class	Specifies a Java reference to provide the rule implementation.
Is prebuilt	If 'Yes': This function is provided by the add-on. If 'No': This function is not provided by the add-on.

Table 15: Filtering record rule properties

Filtering field rule

This table allows you to remove field values from a matching operation to improve results. For example, depending on configuration settings, the values 'Holding Company Inc.' and 'Holding Company' may not register as a match. You could specify that the value of 'Inc.' gets filtered out during the operation, which would then result in a match.

Filtering field rule table (logical name: **FilteringFieldRule**)

This table is located under the EBX® Administration Tab in the 'Matching reference data' dataset.

Properties	Definition
Name	Name of the filtering field rule.
Description	Context of use.
Value filter	The field to exclude during matching.
Java class	Specifies a Java reference to provide the filtering rule. Two built-in classes are available: com.orchestranetworks.addon.daqa.RemoveWordFilter - The add-on applies all values specified in the 'Value filter' property to whole words only. Note that using this class may impact performance. com.orchestranetworks.addon.daqa.RemoveValueFilter - All values configured in 'Value filter' will be applied every time the value is found without taking into account the context. You can define a custom filter by extending the 'MatchingFieldValueFilter' API.
Is prebuilt	If 'Yes': This function is provided by the add-on. If 'No': This function is not provided by the add-on.

Table 16: Filtering field rule properties

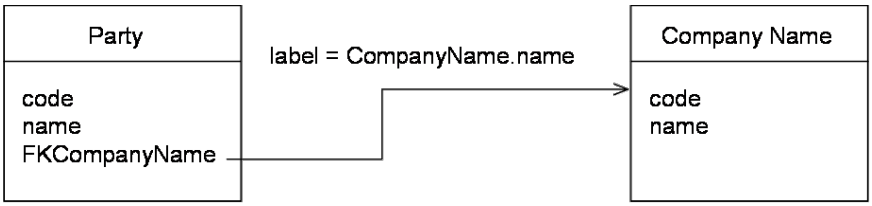
Matching on several tables

The EBX® Match and Merge Add-on can be configured to match data sourced from several tables using their relationships, namely the foreign keys.

Special notation:	
✔	See also the 'Using matching through relation(s)' property in the matching policy configuration. This property allows the add-on to use indirect FK (including join table) and not only direct FK.

The matching value of a foreign key is its label-in the language of the current EBX® session-rather than its language in the 'Table configuration'.

In the example below, a matching policy is configured to match the name of a 'Party' through his foreign key to the 'Company name' table:



The add-on uses the pivot record on the 'Party' table to get the associated pivot record in the 'Company Name' table, then performs matching on the list of company names. All records of the 'Company Name' table with a score higher than the 'stewardship min score' are used to set up a cluster of related 'Party' records.

Here is an example:

Party name	Company name
David	Orchestra
Durand	IZX
Bonnais	Delta
Albert	Delttaz
Bonney	Delttaz

When a match is executed on the pivot record 'Bonney' with a matching policy applied on the company name, then here is the list of suspect records. In this example, the company Delta and Delttaz is considered with a score higher than the minimum threshold to be a suspect.

Bonnais	Delta
Albert	Delttaz
Bonney	Delttaz

When the foreign key label relies on many fields, the add-on uses the average score of these fields.

Matching through relation tables

'Relation matching' allows the add-on to match data through related tables. The matching results on related tables are then used to calculate the score of data on the source table.

Types of relation matching:

- Matching using join table: Matches data of the table defined in the current Matching policy through a join table.

- Matching without join table: Matches data of the table defined in the current Matching policy through a related table.

Special notation:	
✔	Both the source and target table must be registered with the add-on.

Properties	Definition
Relation record score weight	This property allows you to configure the 'Relation table's matching score weight. The add-on combines this table's score with the 'Table to match' score using a weighted average. Default value: '1'
Relation record stewardship min score	This is used when the 'Funneling mode' property in the matching policy is active. If the score of the field is lower than the 'Field stewardship min score' then there is no match. By default the value is set to 100 to make the threshold inactive. Default value: '100'
Using join table	If 'Yes': The relation match is configured through a join table between the 'Table to match' and the 'Relation table'. If 'No': The relation match is configured with a direct relation between the 'Relation table' and the 'Table to match'. Default value: 'No'
Relation match with join table group	The **Relation match with join table** group contains the following options: • **Mode**: Specifies the location of the table referenced by the foreign key. The default is **Same dataset** which implies from the same data model. The **Other dataset** option corresponds to a different dataset, but in the same dataspace. **Other dataspace** implies a dataset located in a different dataspace. • **Join table**: Select the join table that links the **Table to match** with the **Relation table**. Join tables have at least one foreign key to the table defined in the current Matching policy. These tables and the current table can be located in any dataset or dataspace. Although, if the **Mode** is set to **Other dataset** or **Other dataspace**, the join table must be in the same data model or a the dataspace or dataset containing the relation table. You don't need to register the join table with the add-on. • **Link: Join table -> Table to match**: Specify the foreign key used to link the join table and table defined in the current matching policy. • **Relation table**: Select the relation table directly linked by the join table. These tables and the current table can be in any dataset. • **Link: Join table -> Relation table**: Specifies the foreign key used to link the join table and the relation table. • **Matching policy on Relation table**: Select the matching policy to apply to the relation table. Please note that if the join table does not come from the same data model as the source table, it must be located in the same dataset as the relation table.
Relation match without join table	The **Relation match without join table** group contains the following options: • **Mode**: Specifies the location of the table referenced by the foreign key. The default is **Same dataset** which implies from the same data model. The **Other dataset** option corresponds to a different dataset, but in the same dataspace. **Other dataspace** implies a dataset in a different dataspace.

Properties	Definition
	• **Relation table**: Select the related table. These tables and the current table can be in any dataset or dataspace. • **Link: Relation table -> Table to match**: Specifies the foreign key used to link the related table and the table to match against. • **Matching policy on Relation table**: Select the matching policy to apply to the relation table.

Table 17: Relation properties

To apply matching through relation(s):

- Register the desired table with the add-on
- Create a Process policy for the registered table.
- Create a Matching polity for the registered table.
- In the Matching policy :
  - Activate 'Using matching through relation(s)'.
  - Enable 'Use join table' based on your needs.
  - Fill in the corresponding properties for each option.

Testing a matching policy

After creating or editing a matching policy, you might want to use the **Check similarity** service to test whether the policy produces the expected scores. One advantage of testing is that it operates on a small subset of records; you can see results immediately. Also, the **Check similarity** service does not change the matching metadata for the records involved. Therefore, you can try different configurations without having to update matching states after each test.

The **Check similarity** service allows you to compare the similarity of two records in the same table. To test, you choose one of the matching policies configured for the table. The policy specifies which fields to compare and the algorithms to use. You can look at the resulting scores to get a preview of how the add-on would handle these records during a matching operation with this policy.

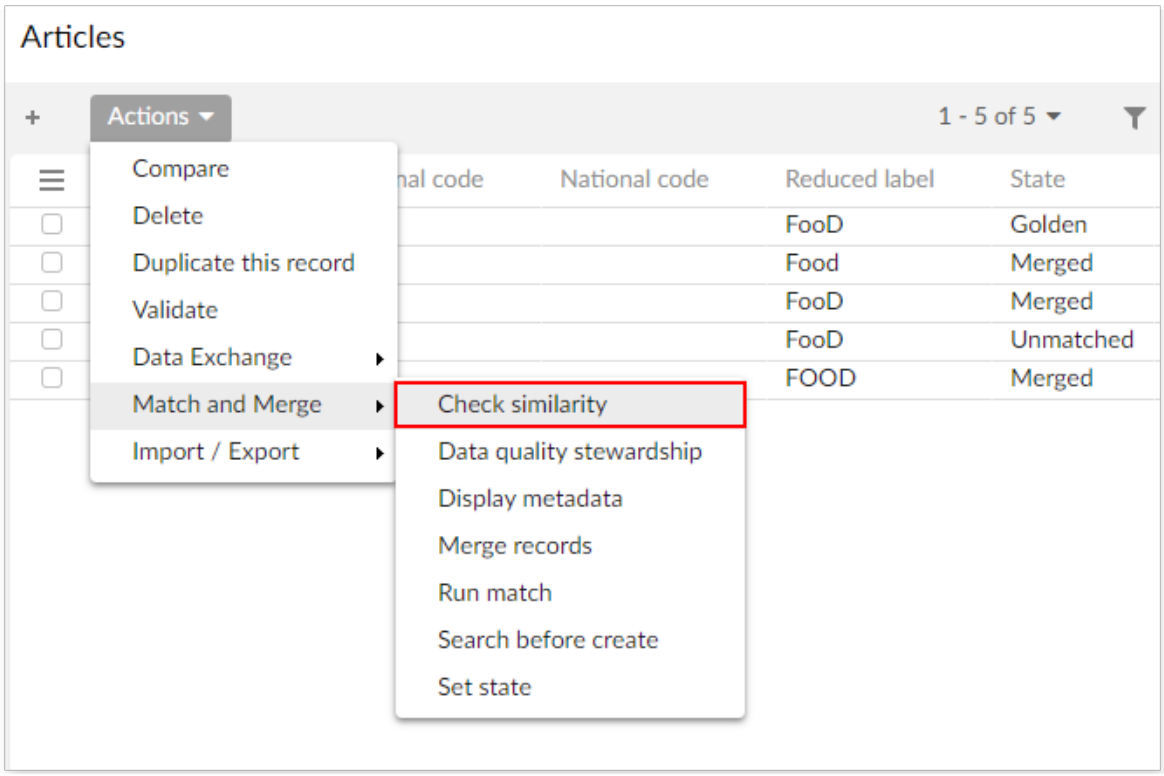
The results returned by the service include the following information for each field:

- The field name.
- The matching **Score weight** shows how much importance is assigned to the field's score when calculating a weighted average.
- The **Algorithm score** shows how closely the fields compare using each algorithm listed. Each algorithm configured for the field is listed here.
- The add-on translates the algorithm's score into the **Similarity percentage score**. Refer to Using matching algorithms with the add-on [p 238] for more information regarding similarity percentages.
- The **Final score** displays an average of the similarity percentages.

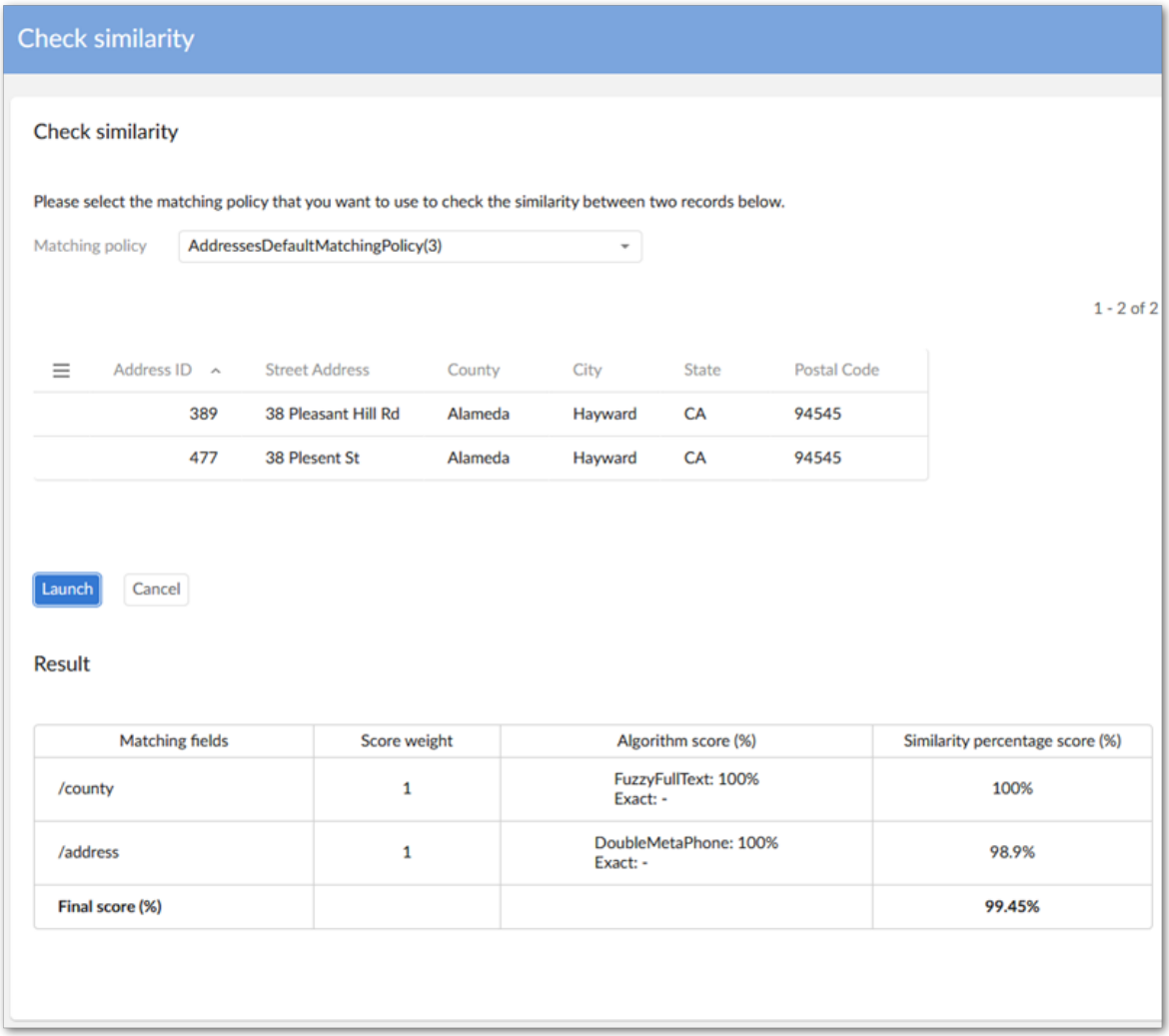
To run the service:

1. Navigate to the table containing the records you want to compare and select two records.

2. From the table's **Actions** menu, select *Match and Merge* > *Check similarity*.



3. Use the **Matching policy** drop-down menu to choose the matching policy you want to test and select **Launch**.



Examples

The sections below show examples of configuring and applying a Matching policy.

Matching policy with exact score

When the configuration literal score = 'Exact' is declared on a matching policy, the business identifier of the table is used as the main criteria for grouping suspect records. This business identifier is not necessarily the primary key, but is one or more fields defined in *TIBCO EBX® Match and Merge Add-on → Table → Business ID.*

When more than one Business ID field is configured, two records are considered as Suspect if any of the Business ID field values are equal.

The following is an example with an Employee table containing the 'Social number' business identifier. The primary key is a technical object identifier (oid).

Two policies are configured: PO1 and PO2 use the metaphone algorithm on the 'Last name' field, and PO1 is configured with literal score = 'Exact'.

When applying PO1, all suspects share the same business identifier value. PO2, however, does not take the business identifier into consideration as it does not specify literal score = 'Exact'.

Existing records

oid (PK)	Social number	Last name	First name
1	987-65-4320	Wood	Nathalie
2	987-65-4320	Dallin	Fred
3	787-14-2277	Wood	Florence

Policy	Social number	Last name	First name	Literal score
PO1		Metaphone		Exact
PO2		Metaphone		

Created record

oid (PK)	Social number	Last name	First name
	987-65-4320	Ewood	John

Matching with PO1

oid (PK)	Social number	Last name	First name	Cluster	State	Score
4	987-65-4320	Ewood	John	120	Pivot	100%
1	987-65-4320	Wood	Nathalie	120	Suspect	78%
2	987-65-4320	Dallin	Fred	120	Suspect	62%
3	787-14-2277	Wood	Florence	001	Golden	100%

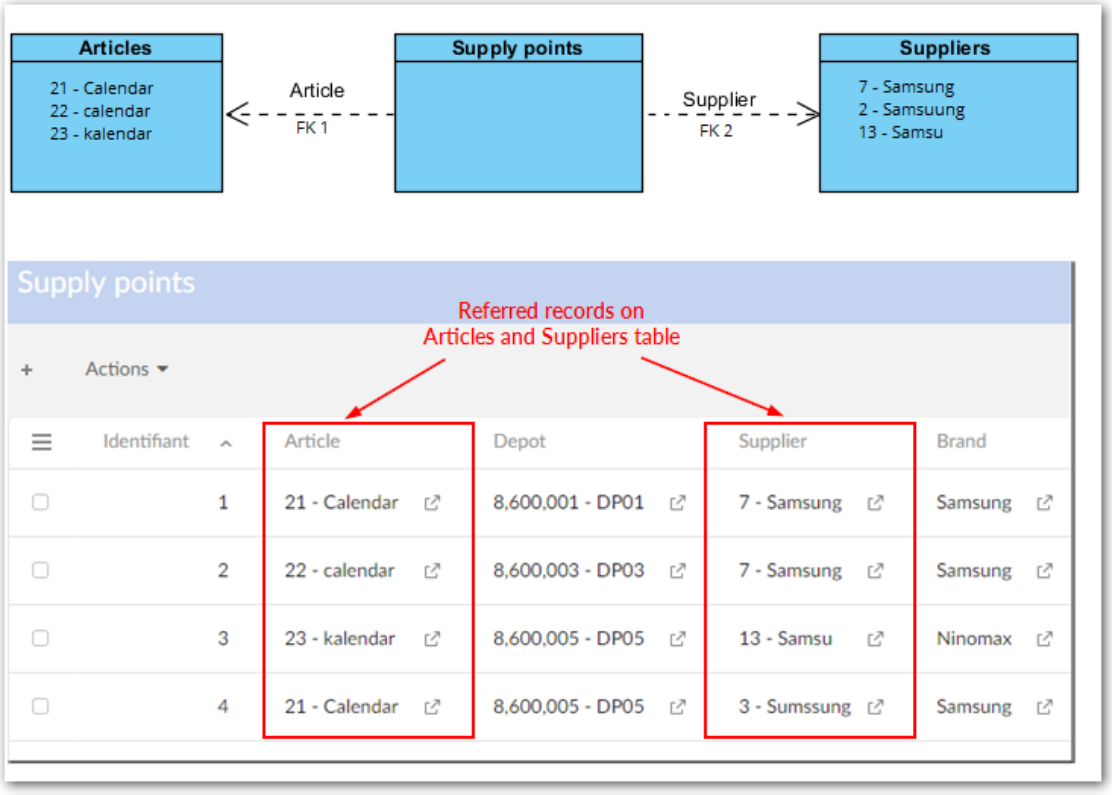
Matching with PO2

oid (PK)	Social number	Last name	First name	Cluster	State	Score
4	987-65-4320	Ewood	John	120	Pivot	100%
1	987-65-4320	Wood	Nathalie	120	Suspect	78%
3	787-14-2277	Wood	Florence	120	Suspect	78%
2	987-65-4320	Dallin	Fred	001	Golden	100%

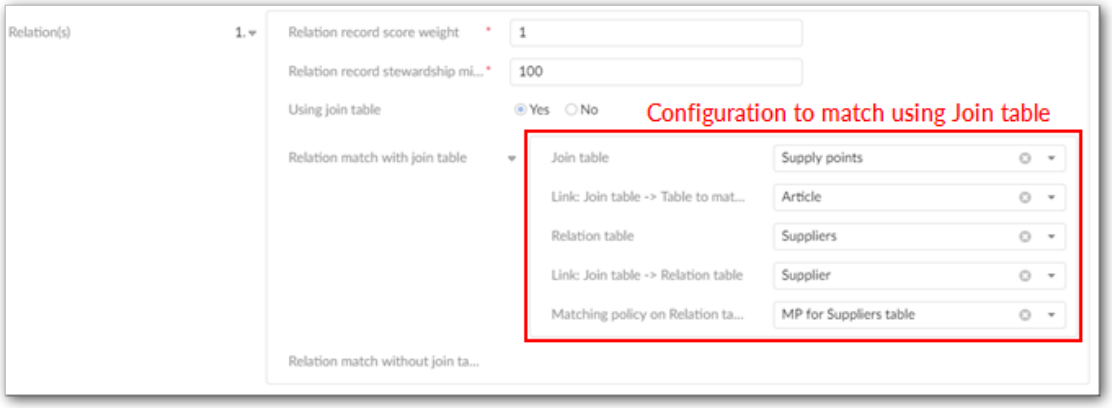
The results are different depending on whether PO1 or PO2 is used.

Matching using Join table

In the Article model, 'Supply points' table is the join table of 'Articles' and 'Suppliers' table.

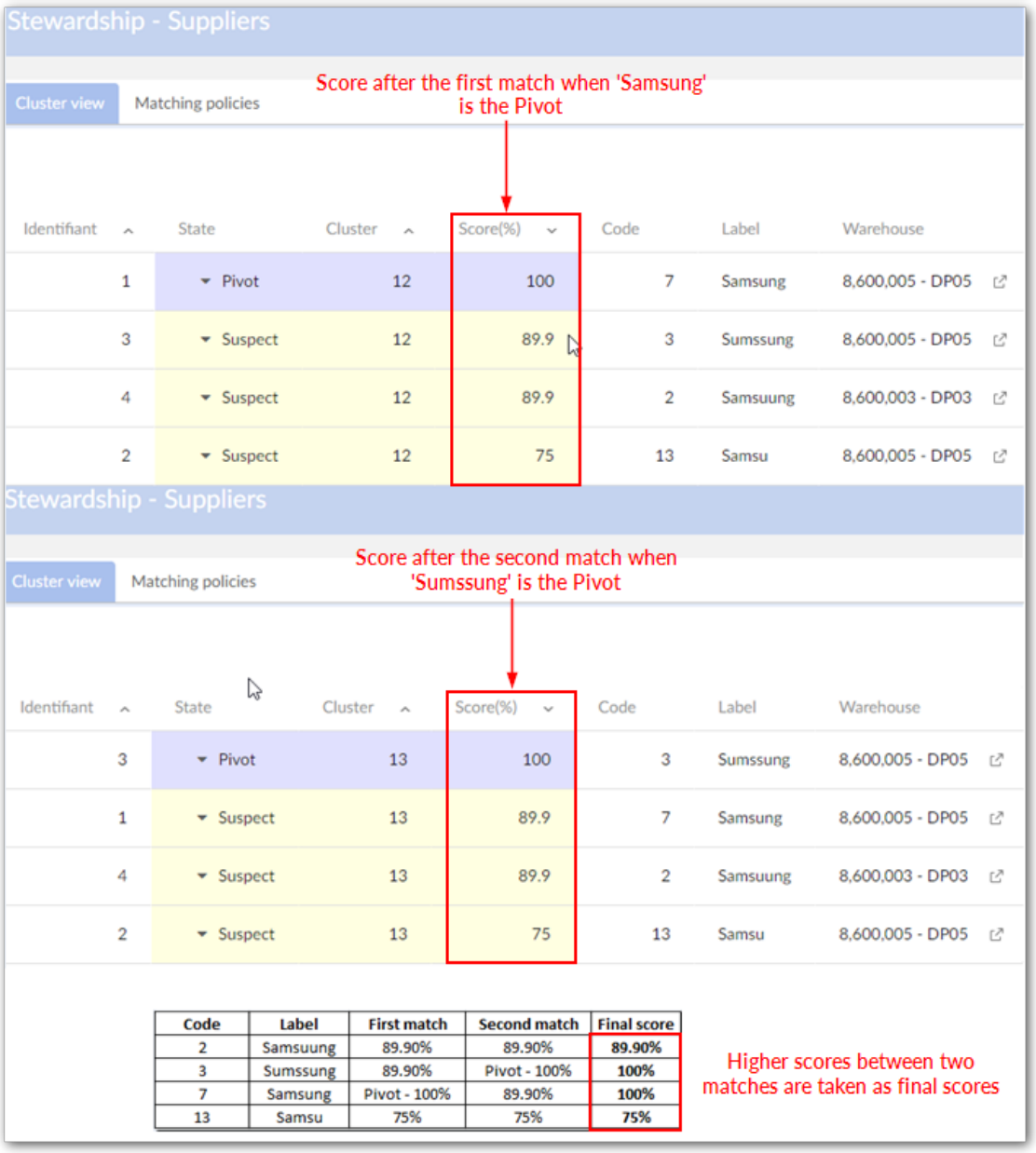


In order to apply matching through the Supply points join table , configure as below:



When you modify record '21-Calendar' on Articles table, the add-on firstly finds records (7-Samsung and 3-Samssungg) in the 'Suppliers' table ('Relation table') that link to the modified record via the 'Supply points' table('Join table'). Those records are considered as the Suppliers table's pivots. The add-on then executes matching on those pivots using the Matching policy configured for the Suppliers

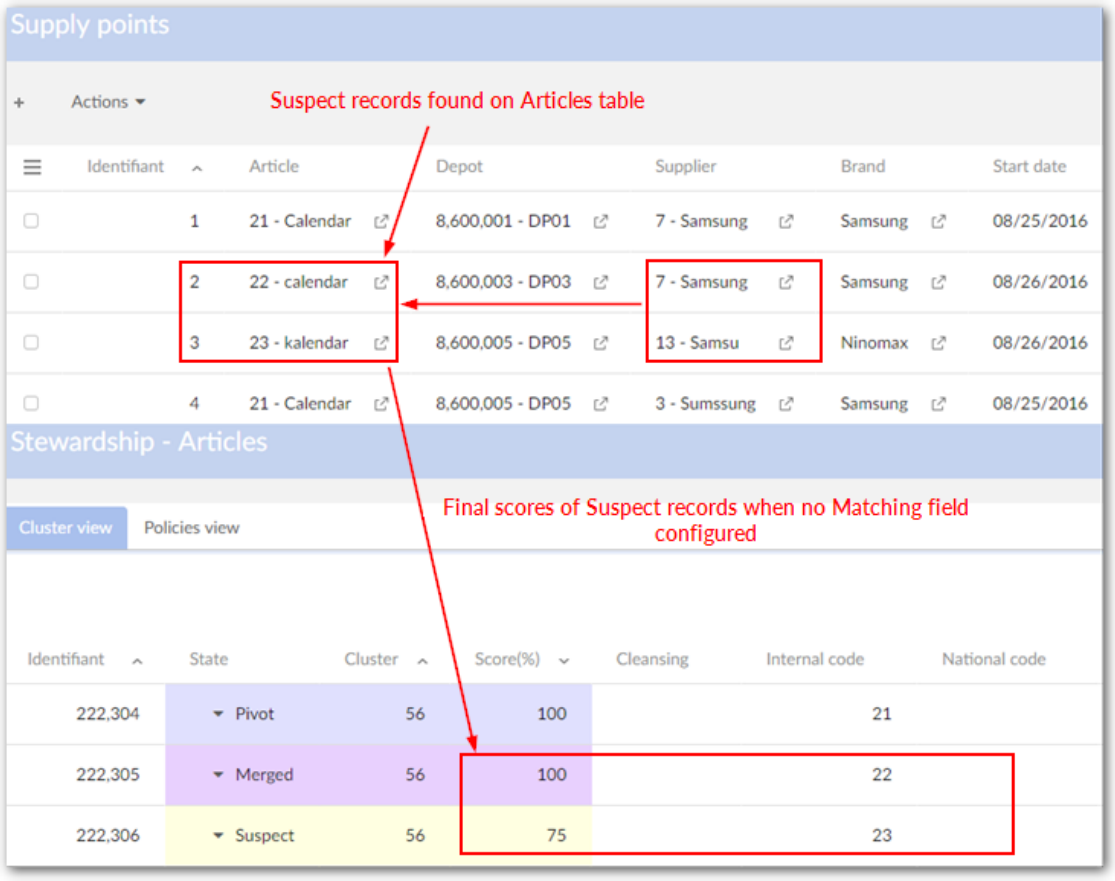
table. The higher scores between two matches are taken as the final score of records in the 'Suppliers table.



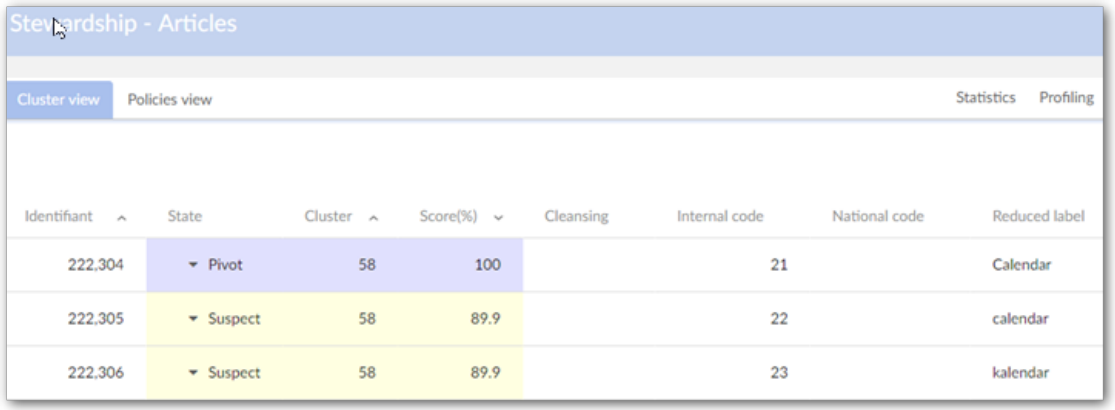
Code	Label	First match	Second match	Final score
2	Samsuung	89.90%	89.90%	**89.90%**
3	Sumssung	89.90%	Pivot - 100%	**100%**
7	Samsung	Pivot - 100%	89.90%	**100%**
13	Samsu	75%	75%	**75%**

Once matching finishes on the Suppliers table, the add-on performs a reverse look up to find Suspect records (22-calendar and 23-kalendar) in the Articles table based on the foreign key. Matching then executes using the Matching policy configured for the Articles table.

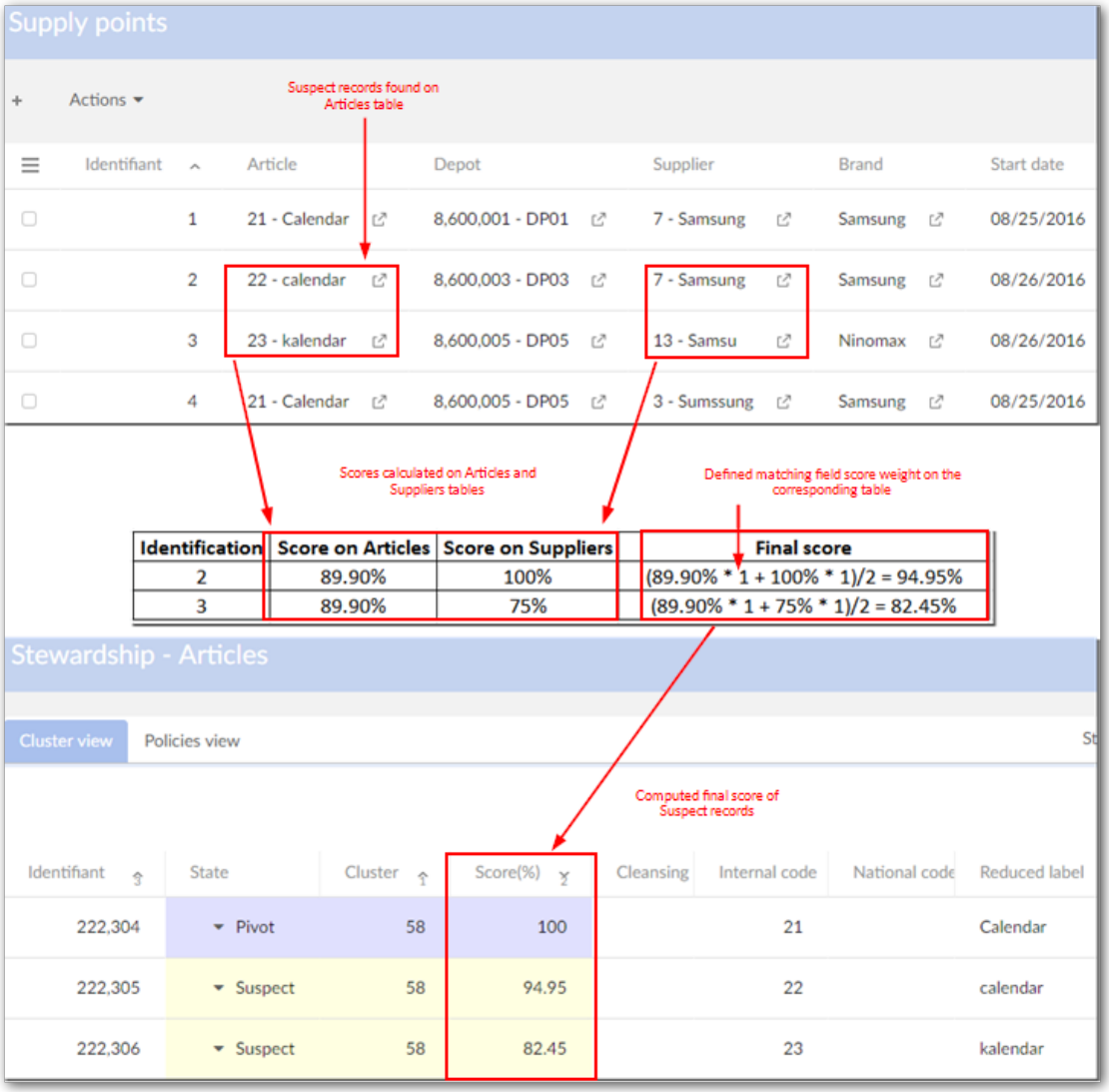
If there is no Matching field is configured, scores of the Suspect records in Articles table are scores of the corresponding records in Suppliers table.



In case 'Reduced label' is configured as the Matching field, the scores of Suspect records on Article are calculated as below:

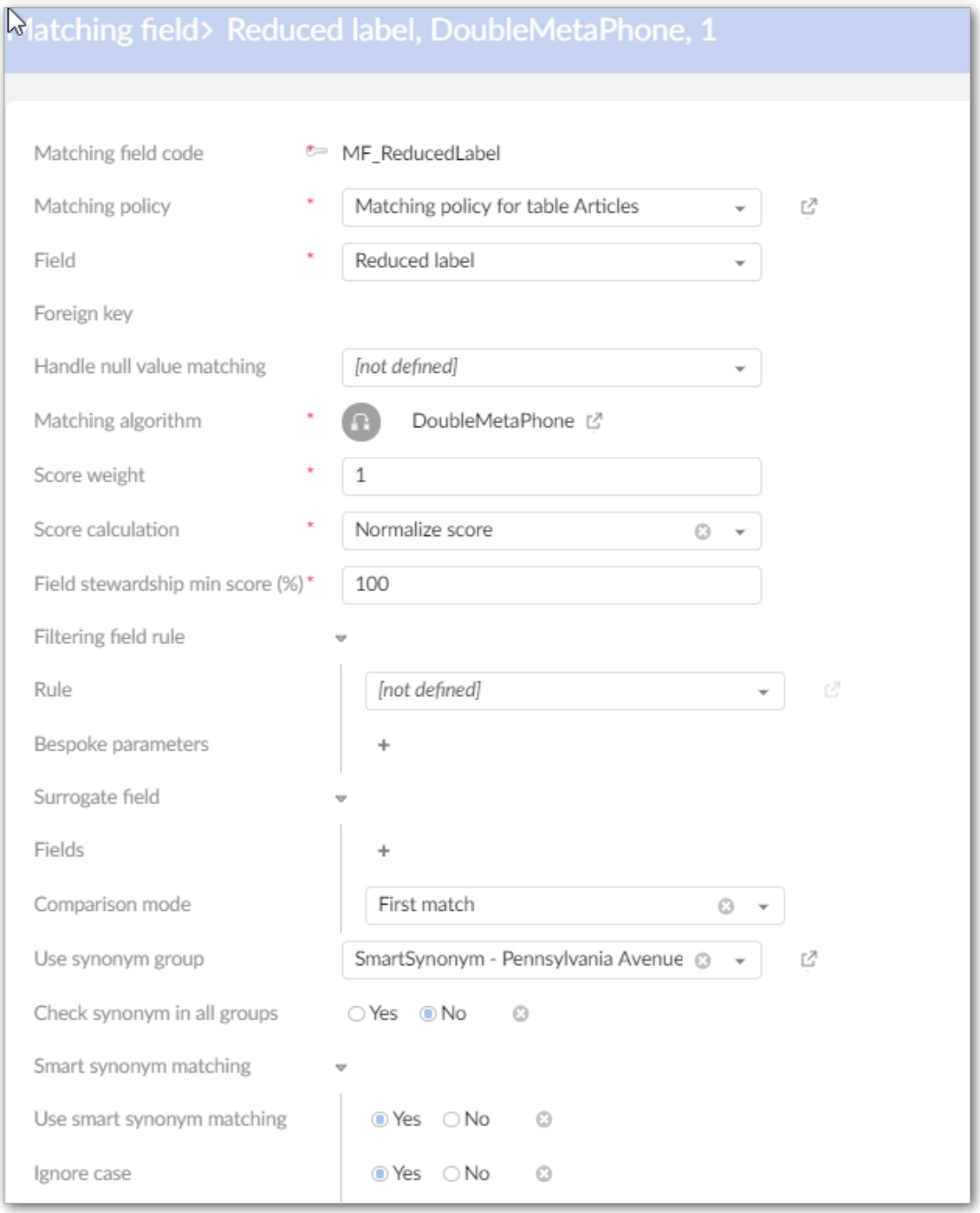


The add-on then combines results and calculates final score of Suspect records in the Articles table using matching field score weights:

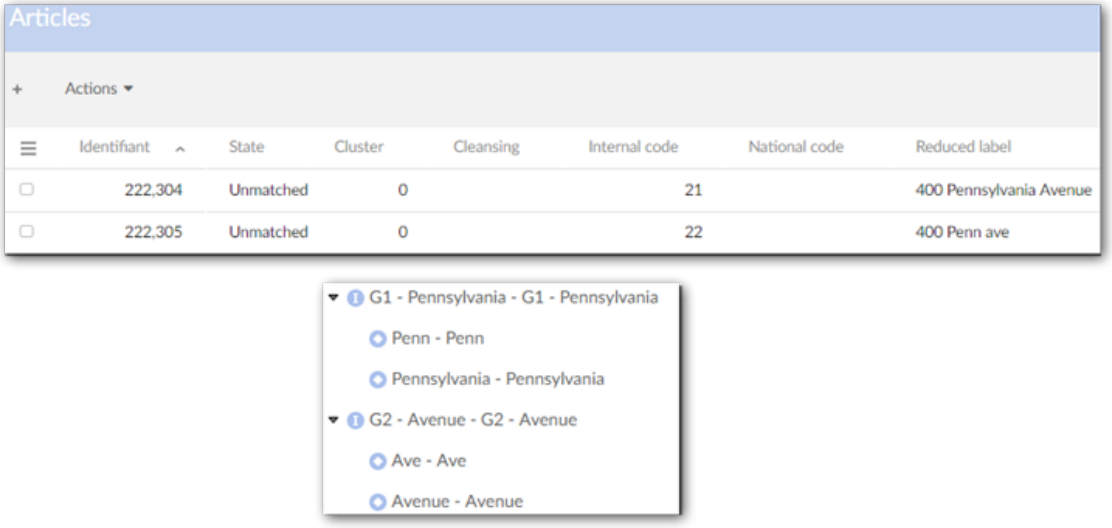


Smart synonym matching

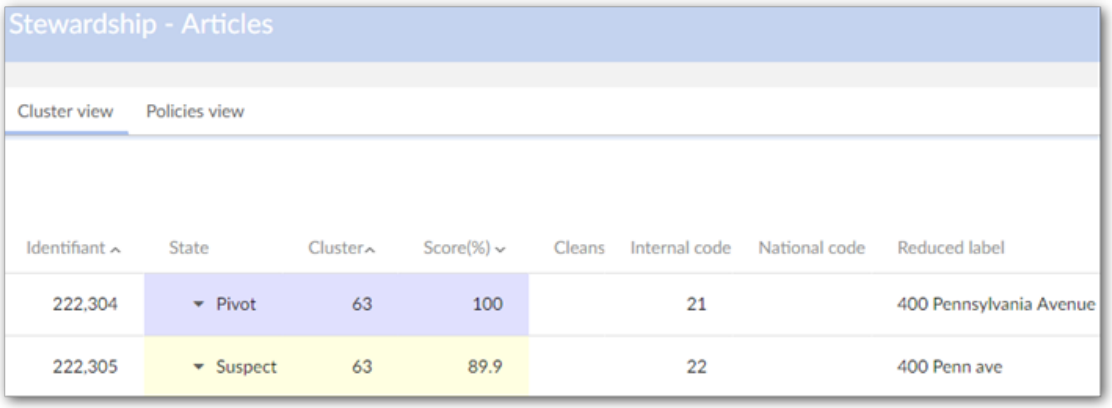
In the following example, the Articles table is configured to use smart synonym matching with the SmartSynonym - Pennsylvania Avenue synonym group.



In the Articles table, there are two records with reduced labels '400 Pennsylvania Avenue' and '400 Penn Ave'. The 'SmartSynonym - Pennsylvania Avenue' group contains two child groups naming 'Pennsylvania' and 'Avenue' as shown below:



When you execute matching on the Articles table, the add-on checks if any strings of '400 Pennsylvania Avenue' existing in recursive child and parent groups of 'SmartSynonym - Pennsylvania Avenue'. If the string exists, the add-on replaces the original string with the shortest value before matching. '400 Pennsylvania Avenue' now becomes '400 Penn Ave' before matching and after matching returns a final score of 89.9% (max score - 0.1).



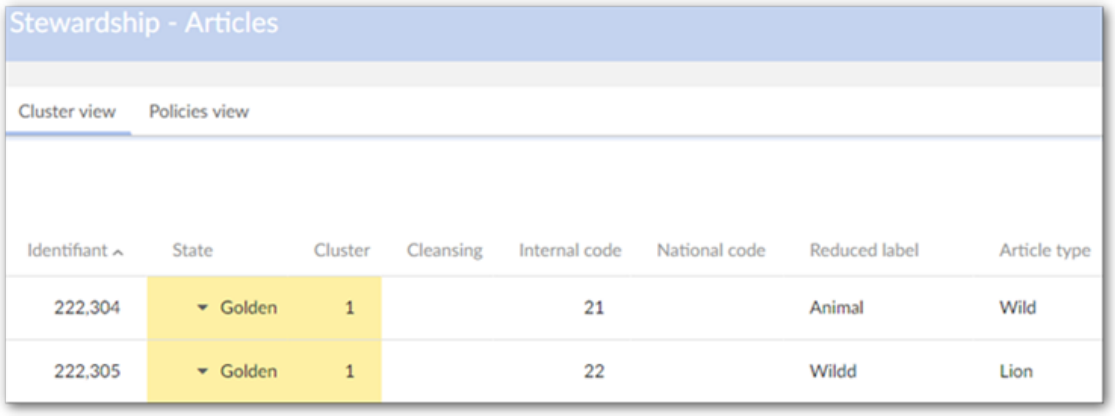
Matching with an alternative field - Surrogate matching

When the matching result on a record is lower than the configured 'Field stewardship min score (%)', you can use another field to match with the current matching field. Make sure that the current matching field is not a complex field, foreign key field, multiple-value field or enumeration field.

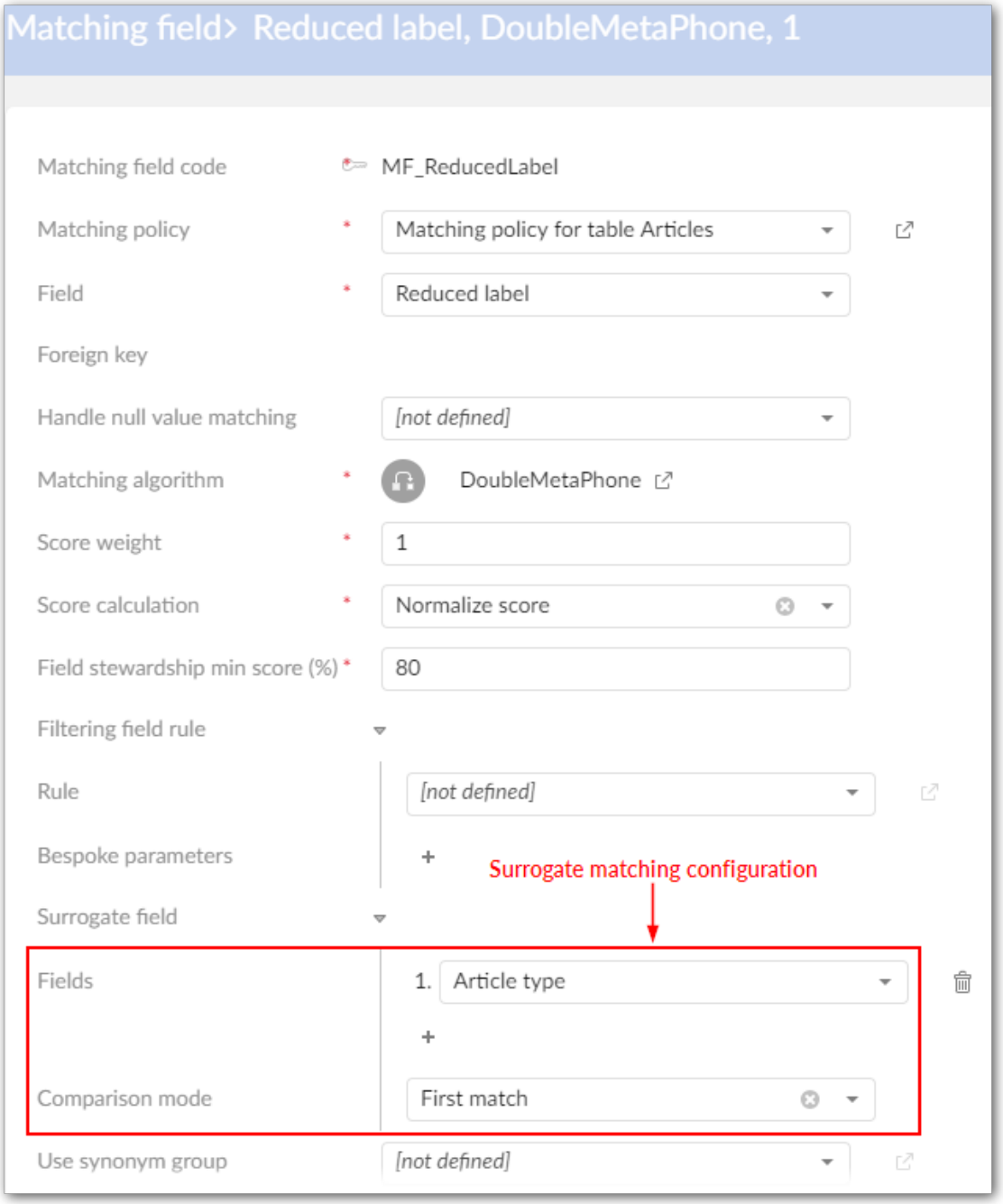
In matching field table → Surrogate field → Fields, select alternative fields in the drop-down list of fields which have the same data type as the current matching field in the matching table. You can configure more than one field to be the alternative matching fields.

As the following example, on Article table, executes 'Match table' on the record with 'Identification' 222305:

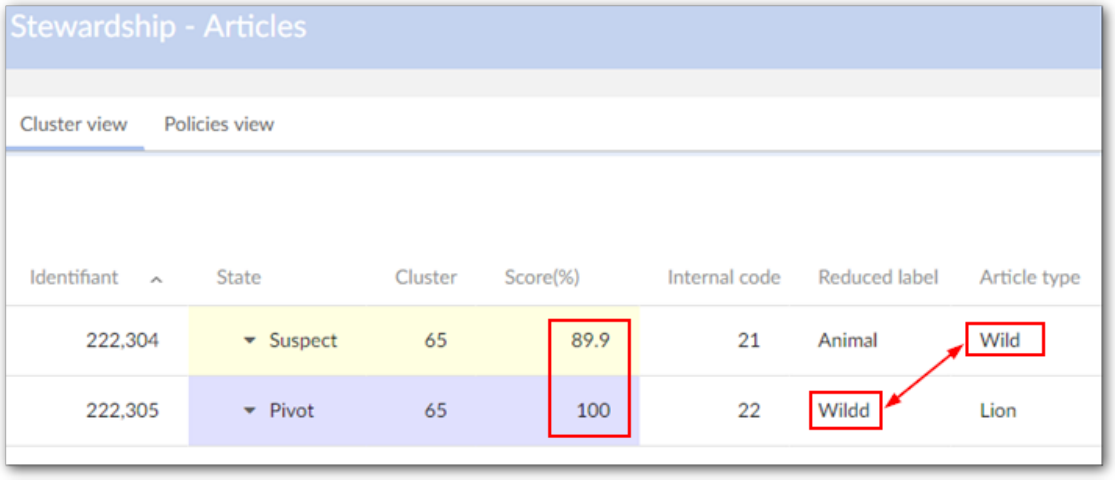
- When the 'Surrogate field' is not defined: no record matches and both become Golden.



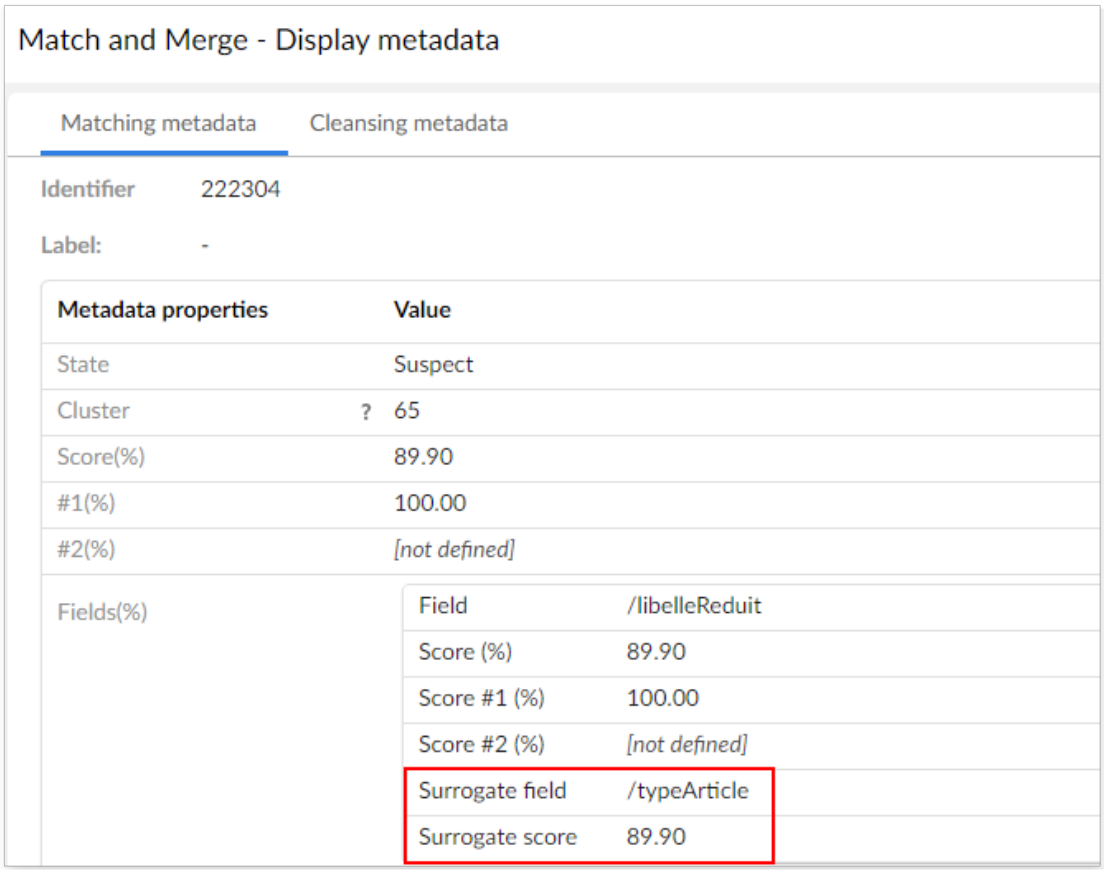
- When 'Article type' is configured as surrogate field with the 'First match' comparison mode:



'Article type' value of record 222304 is used to match with 'Reduced label' value of record 222305. Record 222304 then becomes Suspect with the returned matching score of 89.9%.



All information of the surrogate matching is stored in DaqaMetadata.



Survivorship policy

Introduction

A survivorship policy is applied to a table in order to define how the add-on performs automatic merging of suspect records into a target record.

Once the survivorship target record is identified by applying a 'survivor record selection mode', the add-on executes an automatic merge from suspect records to this target record based on the following rule: only suspect records with a score higher than the stewardship maximum value defined in the process policy are merged.

The add-on relies on four tables to manage the configuration of survivorship policies:

- Survivorship policy context.
- Survivorship policy.
- Survivorship field.
- Survivorship function (in the 'Reference data' part of the configuration).

The relationships between these tables are as follows:

- A survivorship field must be linked to a survivorship policy and a survivorship function.
- A survivorship policy must be linked to the table on which matching is configured.
- A survivorship policy can be linked to a context.

Fields with data types 'value function', 'password' and 'uda' are not merged.

Special notation:	
✔	At execution time, depending on the 'On survivorship' property configured in the 'Process policy', the automatic merge procedure is as follows: • If 'On survivorship' = 'No' no automatic merge. • If 'On survivorship' = 'Yes' and survivorship configuration is not valid or does not exist, then survivorship record is the pivot record and the fields of suspect records are not merged to the pivot.

Survivorship configuration

Survivorship policy context

Survivorship policy contexts ensure that automatic merge occurs only when field values meet certain conditions. Each context can apply to one or more fields. In turn, you can specify multiple values as conditions to satisfy for each field.

Survivorship policy context table (logical name: **SurvivorshipPolicyContext**)

Properties	Definition
Code	Any naming convention without white spaces.
Table	Link to a table under the add-on's control.
Name	Name of the survivorship policy context.
Field contexts	Field context: a field in the table used as the context Use foreign key value: when enabled, the foreign key's value is used instead of its label. This option only displays when the chosen 'Field context' is a foreign key field. Value: value of the field For an empty value this constant must be used: osd:is-empty A multi-valued field cannot be selected as a field context. To use such a field, you need to create a new field with a function to aggregate the values into the excepted format. Then, this field can be used as a field context.

Table 18 :Survivorship policy context properties

Survivorship policy

When defining this type of policy, you can use the **Default survivor field** to apply conditions for survivorship to all of a table's fields (excluding the primary key and source field). To define these conditions, the add-on allows you to:

- Choose a function from the **Survivorship function** list to specify how the add-on selects the most likely duplicate record from a list.
- Use the **Condition for field value survivorship** to enter an expression that defines a condition that must be met for survivorship.
- Specify that the add-on merges values into the golden or pivot record depending on whether the field in the survivorship record is empty.

When you define conditions in the **Default survivor field**, a record must satisfy the record must satisfy the function and condition to be survived.

Special notation:	
✔	When several survivorship policies are active at the same time for different contexts, then the policy with no context is used by default for every execution not related to a defined context. If no survivorship policy is declared, then the pivot will be considered as survivor record and no automatic merge of fields is performed.

Survivorship policy table (logical name: **SurvivorshipPolicy**).

Properties	Definition
Code	Business code of the survivorship policy. Any naming convention is possible.
Short description	Description of the survivorship policy.
Long description	Long description of the policy
Table	Table for which this survivorship policy is defined.
Active	If 'Yes': This survivorship policy is used. If 'No': This survivorship policy is not used. At execution time, depending on the 'On survivorship' property configured in the 'Process policy', the automatic merge procedure is as follows: • If 'On survivorship' = 'No' no automatic merge. • If 'On survivorship' = 'Yes' and survivorship configuration is not valid or does not exist, then survivorship record is the pivot record and the fields of suspect records are not merged to the pivot.
Is context	If 'Yes': This policy is executed for the context defined by the link to the 'Survivorship policy context'. When creating multiple context-based survivorship policies, you can define a default survivorship policy. When conditions do not meet those specified in the context(s), the add-on uses the default survivorship policy. If you choose to not define a default policy and conditions do not meet those specified in the context(s), the add-on takes no survivorship action. If 'No': This survivorship policy is not related to any matching context. Default value: 'No'
Survivorship policy context	Survivorship policy context for which this survivorship policy is defined.
Survivor record selection mode	Rule applied by the add-on to decide which record will be the survivor amongst a set of suspect records: refer to 'Record selection policy'. The add-on still automatically chooses a record based on the current settings if no survivor field is configured. Note that auto-created records are given the highest priority.
Merge all records in cluster	If 'Yes': The add-on merges all data in the cluster including records with a score of -1. Their states then become 'Merged'. Note that this property is ignored when executing a 'Run survivorship on clusters' operation. If 'No': The add-on merges data whose matching score against the Golden or Pivot record is higher than or equal to max score. In case of 'Automatic merge', data of Merged (-1) is not merged into Pivot or Golden record. Default value: 'No'
Default survivor field	Function that will be applied for every attributes of the table (except the primary key that is selected with the 'Survivor record selection mode') to select the value to be survived in the golden record. The default merge function will be overwritten if a survivor field is defined.
Survivorship function	A function that is applied to the field to select the best value amongst a list of records. These functions take into account any conditions specified in the 'Condition for field value survivorship' property.

Properties	Definition
Condition for field value survivorship	The condition is defined as a predicate expression. The Survivorship function will be checked first before moving on to Condition for field value survivorship. A record must satisfy both the Survivorship function and the predicate expression to be survived. If the value of this field is empty or invalid, it will be ignored by the add-on. In case, the predicate expression is invalid, it will be logged in the add-on log file. For example:osd:is-not-null(./AddressLine1) and osd:is-null(./AddressLine2)
Execute if empty only	Decision on whether to apply survivorship function to merge data to Golden/Pivot record or not based on the field value of the survivorship record. If set to: • 'Yes': The survivorship function is executed only if the field value in the survivorship record is empty. • 'No': The survivorship function is executed even if the field value in the survivorship record is not empty. Default value: 'No'

Table 19: Survivorship policy properties

Survivorship field

Configure field merges.

Survivorship field table (logical name: **SurvivorFieldRule**).

Properties	Definition
Survivorship field code	Any naming convention without white spaces can be used.
Survivorship policy	A reference to the 'Survivorship policy' table.
Field	A field in the table for which a survivorship function is defined.
Survivorship function	A function that is applied to the field to select the best value amongst a list of records. For example, in an 'Employee' table, a maximum value function can be applied to the age field to automatically select the greatest age amongst the related suspect records. These functions take into account any conditions specified in the 'Condition for field value survivorship' property. For a list of survivorship functions along with their description, please see Predefined survivorship functions [p 96].
Condition for field value survivorship	The condition is defined as a predicate expression. The Survivorship function will be checked first before moving on to Condition for field value survivorship. A record must satisfy both the Survivorship function and the predicate expression to be survived. If the value of this field is empty or invalid, it will be ignored by the add-on. In case, the predicate expression is invalid, it will be logged in the add-on log file.
Executed if empty only	Decision on whether to apply survivorship function to merge data to Golden/Pivot record or not based on the field value of the survivorship record. If 'Yes': The survivorship function is executed only if the field value in the survivorship record is empty. If 'No': The survivorship function is executed even if the field value in the survivorship record is not empty. Default value: 'Yes'

Table 20: Survivorship field properties

Survivorship field function

Survivorship function table (logical name: **SurvivorshipFunction**).

This table is located under the EBX® Administration Tab in the 'Matching reference data' dataset.

Properties	Definition
Name	The function's name.
Description	The function's objective.
Is prebuilt	If 'Yes': This function is provided by the add-on. If 'No': This function is not provided by the add-on.
Java class	Specifies a Java reference to provide a custom algorithm.

Table 21: Survivorship function properties

Special notation:	
✔	Currently for survivorship functions, if there are more than one value can be survived, the value of the record with the latest timestamp will be selected by default.

Predefined survivorship functions

The add-on provides predefined survivorship functions to drive the merge at the field level. When no function is defined, the 'Best score' function is used by default.

Survivorship function	Description
Best score	The field value used is that of the record with the best score amongst all related suspects.
Constant	The field value used is set to value defined as constant, whatever merged records values.
Most trusted source	The field value is taken from the record with the best trusted source amongst all related clusters. If there are two sources with the same priority, the add-on will determine the most trusted based on latest timestamp. If this field is not declared in the 'Field trusted source' table, then the merge for this field is not executed.
Longest	This field value is taken from the record with the biggest field value size (string). The length of a multi-value field is counted as the number of elements in the list.
Max	The field value used is the maximum value amongst all related suspects.
Min	The field value used is the minimum value amongst all related suspects.
Most frequent	The field value used is the value found most frequently amongst all related suspects. Note that this function ignores the value from an auto-created record.
Most recently acquired	The field value is taken from the last timestamps record.
No merge	The field value is not merged

Table 22: Predefined survivorship functions

Record selection policy

Introduction

A record selection policy defines the method used to select a pivot record for the matching procedure and to select a survivorship record.

Pivot selection

A pivot record selection policy is specified when configuring a matching policy.

By default, the add-on uses the new or updated record as the pivot record for computing the match score. This is called the 'newUpdate' policy.

If a policy other than 'newUpdate' is selected, that policy will be enforced after the 'newUpdate' policy has been applied.

Special notation:	
✕	The survivorship will override 'Pivot selection mode'.

Survivorship record selection

When the survivorship process is started, the add-on has to select the record used as the survivorship, or target record. When the record selection policy is 'newUpdate', all suspects to be merged will be merged into the pivot. If another policy is then executed, the pivot is no longer the survivorship record.

Record selection policy

Record selection (logical name: **RecordSelection**).

This table is located under the EBX® Administration Tab in the 'Matching reference data' dataset.

Properties	Definition
Name	The record selection policy name.
Description	The record selection policy description.
Is prebuilt	If 'Yes': This record selection policy is provided by the add-on. If 'No': This record selection policy is not provided by the add-on.
Java class	Specifies a Java reference to provide a custom algorithm.

Table 23: Record selection properties

Predefined record selection policies

The add-on provides these ready-to-use selection policies. When a record selection policy does not provide any records, then the policy 'newUpdate' is used by default except if another one is mentioned in the table below.

Record selection policy	Description
Best score	Record with the best score after the pivot record
Longest	Record with most information (size). All fields (string data type) are taken into account except matching metadata. The length of a multi-value field is the length of longest element in the list.
Most complete	The record with fewest empty fields.
Most complete and longest	Record with the fewest empty fields and with the most information (size). All fields are taken into account except matching metadata. Most complete takes priority over the longest.
Most recently acquired	The record with the latest timestamp managed through the matching metadata tab.
Most trusted source	The record with most trusted source.
New update	The record created or modified.
Was golden	The record is selected amongst 'Was golden' records and current 'Golden'. If many records with the 'Golden' or 'Was golden' identifier exist, the 'Most recently acquired' record is used.

Table 24: Predefined record selection policies

Trusted sources

A source identifies a system that provides data either at the record or field level.

For instance, in a company, the employee's records are provided either by the human resources system (source 1) or a manufacturing system (source 2). For every record, the salary field can be also provided by the accounting system (source 3). Based on these three sources, it is possible to declare trusted levels such as:

- Amongst duplicate records, it is better to select the record stemming from the human resources (source 1) rather than the manufacturing (source 2).
- To select the salary field between duplicate records, it is better to use one stemming from the accounting system (source 3) in priority, and then human resources (source 1) and manufacturing (source 2).

The add-on uses the trusted level mechanism in these procedures:

- When selecting a record (record selection policy is 'Most trusted source').
- When merging a field by the automatic merge (survivorship field policy is 'Most trusted source').

The field stating the source depends on each table under add-on control. The configuration of this field is set in the 'Table' table (logical name: **TableConfiguration**) through the 'Source' parameter field.

Trusted source configuration

> **Note**
>
> If no specific fields are defined as a trusted source, the add-on applies the trust level set for the table to all of its fields.

Source

Source (logical name: **Source**).

Properties	Definition
Name of source	Business source name – any naming convention can be used and the add-on compares this value with a record's source field value.
Description	Description of the source

Table 25: Source properties

Table trusted source

Table trusted source table (logical name: **TableTrustedSource**).

Properties	Definition
Table	A table under the control of the add-on.
Trusted source list	List of sources ordered by trust

Table 26: Table trusted source properties

Field trusted source

Field trusted source table (logical name: **FieldTrustedSource**).

Properties	Definition
Table	A table controlled by the add-on.
Field	A field in the table controlled by the add-on.
Trusted source list	List of sources ordered by trust

Table 27: Field trusted source properties

Language management

Each table under add-on control is associated with one of the languages defined in 'ebx.locales' (the languages managed by the EBX®). The matching process uses this language.

The language configuration is declared in the table configuration.

Synonym configuration

The configuration of synonyms is available in the 'Semantic' data domain.

When configuring a matching field policy, you can refer to a group of synonyms used for the de-duplication process. The add-on provides some sample synonyms.

Synonym

A synonym is any phrase that will be considered as another's synonym. The synonym items are synonyms with each other within a group of synonyms (refer to the table 'Synonym group'). To attach synonyms to a group, the data hierarchy view 'Synonym by group' is used.

Source (logical name: **Item**).

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.

Table 28: Synonym

Synonym group

The synonym items that are synonym with each other are grouped into a Synonym group.

Source (logical name: **Synonym group**).

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 items to a group.
Parent group	Groups of synonyms can be arranged through a parent relationship. At the time you use a synonym group for matching a field, you can configure the matching policy to look for synonyms through the parent relation.

Table 29: Populate synonym group

Populate synonym group

This table shows the relationships between synonym items and synonym groups. A synonym group may contain many synonym items.

Properties	Definition
Synonym group	A record of Synonym group table.
Synonym	A record of Synonym table.

Table 30: Populate synonym group

6.3 Matching operations

In this section, you can find information on matching operations and their contexts for use. Sections are divided into operations that apply to a single record and to those that apply to a set of records.

Operations applied to a single record

List of operations	Type and context of use
Match cluster	Matching operations applied to a single record are not available if the process policy is incorrectly configured or the 'On matching Process' property is set to 'No'. EBX® standard services remain available from the EBX® view. It is possible to duplicate, compare and delete a record. Because the deletion is physical, it can entail integrity constraint failures. For example, if a merged record holds a foreign key to the record that has been deleted, then EBX® will raise integrity errors. You can make the EBX® delete service inaccessible based on user permission levels. This helps to ensure that a logical deletion with the add-on is always used. Note that if a suspect record is manually removed from a cluster using one of these operations and the cluster contains only pivot and merged records, the pivot becomes golden.
Match table	
Match suspicious	
Create new golden	
Set golden	
Set back golden	
Set definitive golden	
Unset golden	
Not suspect	
Merge	
Automatic merge	
Unmerge	
Switch pivot	
Remove from cluster	
Remove and set golden	
Add into cluster	
Set merged (ignore)	
Cancel ignore	
Delete	
Undelete	
Align foreign key of merged record	

List of operations	Type and context of use
Push out suspect (-1)	
Display meta-data (also available at the table view)	UI action applied to single record.
Manage cluster	
Show cluster	
Show cluster for merge	
Display merged records, Display relation records	
Display record	
Hide cluster	
Modify record	
Fix relation records (for pivot and golden when under matching)	
Clean up merged fields log	

Table 31: List of operations applied to single record

On suspicious

Suspicious records are located into the '004' cluster.

Suspicious	
Match cluster	N/A
Match table	Matching against all records in the table with unmatched state (in the '000', not in groups), suspect, pivot and golden. Other states that are not used to execute the match include: to be matched, suspicious, definitive golden, merged and deleted. New scores are computed.
Match best cluster	Matching against Pivot and Golden (with cluster ids > 10). The record will be moved to the cluster that has the best score. The current Pivot or Golden is still prioritized as Pivot and Golden. A **Match table** operation will be initiated if no Pivot or Golden record matches the record.
Match suspicious	Simple matching against all records in the table with unmatched state (in the '000' cluster, not in groups), suspect, pivot and golden. Other states that are not used to execute the match include: to be matched, suspicious, definitive golden, merged and deleted. New scores are computed. The suspicious record is then either associated to one to many potential suspect records or directly considered a golden record.
Create new golden	N/A
Set golden	Record is declared as golden and moves to the '001' cluster.
Set back golden	N/A
Set definitive golden	Same as 'Set golden' but the record moves to the '003' cluster.
Unset golden	N/A
Not suspect	N/A
Merge	N/A
Automatic merge	N/A
Unmerge	N/A
Switch pivot	N/A
Remove from cluster	N/A
Remove and set golden	N/A
Add into cluster	The record becomes a suspect record. It is integrated into a selected cluster along with its merged records and its score is computed against the pivot or golden record. Regardless of the result of this score, the record is kept in the cluster.

Suspicious	
	If there is no pivot or golden record in the selected cluster, then the new record becomes the pivot record. If a pivot already exists, the new record is put into the suspect state. If a golden record already exists, the golden record becomes the pivot record and the new record is put into the suspect state. No automatic merge is executed. It is not possible to add the record into a cluster that is used to group unmatched (to be matched) records.Note that if you move a record other than Merge into a new cluster, its state will be changed to Golden.
Set merged (ignore)	N/A
Cancel ignore	N/A
Delete	Set to deleted and remains in the current cluster.
Undelete	N/A
Align foreign key	N/A
Push out suspect (-1)	N/A

On suspect

Suspect	
Match cluster	N/A
Match table	Matching against all records in the table with unmatched state (in the '000' cluster, not in groups), suspect, pivot and golden. Other states that are not used to execute the match include: to be matched, suspicious, definitive golden, merged and deleted. New scores are computed.
Match suspicious	N/A
Create new golden	N/A
Set golden	The record is declared golden. All other records in the cluster (except merged and deleted) are marked as merged without a merge operation. The score is recomputed for each record in the cluster against the new golden record. Records are kept in the cluster.
Set back golden	Available only if 'Was golden'= 'Yes'. The record becomes golden and is placed into the '001' cluster without any matching or merging. If the **Auto create new golden for single golden** property is active, the add-on creates a new golden record and updates the current record to merged. The remaining records in the impacted cluster will be fixed.
Set definitive golden	Same as 'Set golden' but the record moves to the '003' cluster. The current cluster can have only merged records since the golden record has moved to the '003' cluster.
Unset golden	N/A
Not suspect	Record is no longer a suspect against the pivot or golden record. The record becomes unmatched and is moved to the '000' cluster.
Merge	Record is merged into the pivot using selected fields in the suspect record (selected in the merge UI). One record is merged at a time. No survivorship rules are applied.
Automatic merge	N/A
Unmerge	N/A
Switch pivot	The current record becomes the pivot. The former pivot becomes a suspect record. New scores are recomputed for all records in the cluster against the new pivot. These records are kept in the cluster regardless of the new results (score='-1' is no match). Automatic merge is not executed.
Remove from cluster	The record is removed from the cluster, set to unmatched, and placed into the '000' cluster. The cluster may have a pivot with no suspects.
Remove and set golden	The record is removed from the cluster, set to golden, and placed into the '001' cluster. If the **Auto create new golden for single golden** property is active, the add-on creates a new golden record and updates the current record to merged. The remaining records in the impacted cluster will be fixed.

Suspect	
Add into cluster	N/A
Set merged (ignore)	The record is set to a merged state without performing any merge operations. This is one way of rejecting records (ignore). The link to the target record is set to null.
Cancel ignore	N/A
Delete	The record is set to deleted and remains in the current cluster with its current score. If there are no remaining Suspects in the cluster, the Pivot becomes Golden.
Undelete	N/A
Align foreign key	N/A
Push out suspect (-1)	N/A

On pivot

Pivot	
Match cluster	Matching against suspect records in the cluster. No automatic merge. Suspects are kept in the cluster regardless of the score ('-1' if the minimum threshold to be considered as a suspect).
Match table	Matching against all records in the table with unmatched state (in the '000' cluster, not in groups), suspect, pivot and golden. Other states that are not used to execute the match include: to be matched, suspicious, definitive golden, merged and deleted. New scores are computed.
Match suspicious	N/A
Create new golden	A new pivot record is created if all primary key fields are auto-incremented or done manually from a pop-up. It replaces the existing pivot and the existing pivot becomes a suspect. The Merge view automatically displays.
Set golden	The record becomes golden. All other records in the cluster (except merged and deleted) are set to merged without any merge. Score is recomputed for each record in the cluster against the new golden record.
Set back golden	Available only if 'Was golden'= 'Yes'. Record becomes golden and is placed into the '001' cluster without any matching or merging. The pivot then becomes the most recently updated record. New scoring is executed. All existing suspect records are kept in the cluster whatever the score. If the **Auto create new golden for single golden** property is active, the add-on creates a new golden record and updates the current record to merged. The remaining records in the impacted cluster will be fixed.
Set definitive golden	Same as 'Set golden' but the record moves to the '003' cluster. The current cluster can have only merged records.
Unset golden	N/A
Not suspect	N/A
Merge	N/A
Automatic merge	All suspect records in the cluster are merged into the pivot using defined survivorship policies. The record becomes a golden record.
Unmerge *The EBX® Data history function must be active on the table.*	Value changes revert to that of the pivot record before its last merge, but only if the last merged record's timestamps are still equal to the pivot record timestamp. If these timestamps differ, unmerging is no longer possible. Merged records become suspect records and the pivot record remains in its state.
Switch pivot	N/A
Remove from cluster	N/A
Remove and set golden	N/A

Pivot	
Add into cluster	N/A
Set merged (ignore)	N/A
Cancel ignore	N/A
Delete	The record's state is set to deleted, the record remains in the current cluster. The cluster can have no pivot. Every suspect is set to '-1'. If **On suspect record retention** is activated: • And there is only one Suspect, the Suspect becomes Golden. • And there are multiple Suspect records, the add-on selects a new Pivot and executes a **Match cluster** operation to recalculate the score.
Undelete	N/A
Align foreign key	Realign foreign keys of merged records of the pivot in the dataspace.
Push out suspect (-1)	All suspect (-1) records of the cluster move to the unmatched cluster. If there are no remaining Suspect record, the Pivot becomes Golden.

On golden

Located either in non-predefined clusters or in '001' (golden) or '003' (definitive golden).

Golden	
Match cluster	N/A
Match table	Matching against all records in the table with unmatched state (in the '000' cluster, not in groups), suspect, pivot and golden. Other states that are not used to execute the match include: to be matched, suspicious, definitive golden, merged and deleted. New scores are computed. If matching does not find a potential duplicate record, the golden record remains in its current cluster.
Match suspicious	N/A
Create new golden	N/A
Set golden	N/A
Set back golden	N/A
Set definitive golden	Available only if the record is not in the '003' cluster. The record moves to the '003' cluster. The current cluster (if different from '001') can have only merged records since the golden has moved to the '003' cluster.
Unset golden	Record becomes pivot. All other merged records in the cluster have 'Target record' field points to the Golden are set to suspect and the target record is set to 'null'. For a golden record in the '001' and '003' clusters, the record moves to the '000' cluster as unmatched. For golden from '003' former merged records are set to suspect with a score set up to '-1'.
Unset golden recursively	Record becomes pivot. All other merged records in the cluster are set to suspect with scores before being merged.
Not suspect	N/A
Merge	N/A
Automatic merge	N/A
Unmerge *The EBX® Data history function must be active on the table.*	Changes the value back to that of the golden record before its last merge, but only if timestamps of the last merged records are still equal to the timestamp of the golden record. If these timestamps differ, unmerging is no longer possible. Merged records become suspect records and the golden record becomes a pivot record.
Switch pivot	N/A
Remove from cluster	N/A
Remove and set golden	N/A

Golden	
Add into cluster	Record becomes the new pivot in the cluster, match cluster is executed, all suspects are kept and merge is not executed. Its merged records are also moved to the new cluster without changing states. It is not possible to add the record in a cluster that is used to group unmatched (to be matched) records. Note that if you move a golden record into a new cluster, its state will remain the same.
Set merged (ignore)	N/A
Cancel ignore	N/A
Delete	The record's state is set to deleted and the record remains in the current cluster with its current score. The cluster can have merged records only.
Undelete	N/A
Align foreign key	Realigns foreign keys of the merged records in the dataspace.
Push out suspect (-1)	N/A

On merged

A merged record is located in a cluster different from reserved clusters ('000' to '010').

Merged	
Match cluster	N/A
Match table	N/A
Match suspicious	N/A
Create new golden	N/A
Set golden	N/A (even if Was golden = yes)
Set back golden	N/A
Set definitive golden	N/A
Unset golden	N/A
Not suspect	N/A
Merge	N/A
Automatic merge	N/A
Unmerge	N/A
Switch pivot	N/A
Remove from cluster	N/A
Remove and set golden	N/A
Add into cluster	The record is integrated into a selected cluster a long with its merged records without changing states and scores.
Set merged (ignore)	N/A
Cancel ignore	Available only if the link to the target record is null. The record becomes a suspect record and remains in the current cluster with its current score.
Delete	N/A
Undelete	N/A

Merged	
Align foreign key	N/A
Push out suspect (-1)	N/A

On unmatched

An unmatched record is located in the '000' (unmatched) predefined cluster or located in a normal cluster through the 'Group at once unmatched' operation.

Unmatched	
Match cluster	N/A
Match table	Matching against all records in the table with unmatched state (in the '000' cluster, not in groups), suspect, pivot and golden. Other states that are not used to execute the match include: to be matched, suspicious, definitive golden, merged and deleted. New scores are computed. Depending on the record scores in the table, the unmatched record may move to another cluster, including the '001' cluster if it directly becomes a golden record. Not available for unmatched records that have been grouped by the 'Group at once unmatched' operation.
Match best cluster	Matching against Pivot and Golden (with cluster ids > 10). The record will be moved to the cluster that has the best score. The current Pivot or Golden is still prioritized as Pivot and Golden. A **Match table** operation will be initiated if no Pivot or Golden record matches the record.
Match suspicious	N/A
Create new golden	N/A
Set golden	The record becomes golden and moves to the '001' cluster.
Set back golden	N/A
Set definitive golden	The record becomes golden and moves to the '003' cluster.
Unset golden	N/A
Not suspect	N/A
Merge	N/A
Automatic merge	N/A
Unmerge	N/A
Switch pivot	N/A
Remove from cluster	N/A
Remove and set golden	N/A

Unmatched	
Add into cluster	The record becomes a suspect record. It is integrated into a selected cluster a long with its merged records and its score is computed against the pivot or golden record. Regardless of the result of this score, the record is kept in the cluster. If there is no pivot or golden record in the selected cluster, then the new record becomes the pivot record. If a pivot already exists, the new record is put into the suspect state. If a golden record already exists, the golden record becomes the pivot record and the new record is put into the suspect state. No automatic merge is executed. It is not possible to add the record to a cluster that is used to group unmatched (to be matched) records. Note that if you move a record other than Merge into a new cluster, its state will be changed to Golden.
Set merged (ignore)	N/A
Cancel ignore	N/A
Delete	The record's state is set to deleted; the record remains in the current cluster.
Undelete	N/A
Align foreign key	N/A
Push out suspect (-1)	N/A

On deleted

A deleted record can be located in any cluster.

Deleted	
Match cluster	N/A
Match table	N/A
Match suspicious	N/A
Create new golden	N/A
Unmatch	N/A
Set golden	N/A
Set back golden	N/A
Set definitive golden	N/A
Unset golden	N/A
Not suspect	N/A
Merge	N/A
Automatic merge	N/A
Unmerge	N/A
Switch pivot	N/A
Remove from cluster	N/A
Remove and set golden	N/A
Add into cluster	N/A
Set merged (ignore)	N/A
Cancel ignore	N/A
Delete	N/A
Undelete	The record is set to unmatched state in the '000' cluster with the score '-1'.

Deleted	
Align foreign key	N/A
Push out suspect (-1)	N/A

On to be matched

A record that is ready to be matched is located in the '002' predefined cluster, or located in a normal cluster through the 'Group at once to be matched' operation.

To be matched	
Match cluster	N/A
Match table	Matching against all records in the table with unmatched state (in the '000' cluster, not in groups), suspect, pivot and golden. Other states that are not used to execute the match include: to be matched, suspicious, definitive golden, merged and deleted. New scores are computed. Depending on the scores of the records in the table, the record in the 'To be matched' state may move to another cluster including the '001' cluster if it directly becomes a golden record. Not available for 'To be matched' records that have been grouped by the 'Group at once unmatched' operation.
Match best cluster	Matching against Pivot and Golden (with cluster ids > 10). The record will be moved to the cluster that has the best score. The current Pivot or Golden is still prioritized as Pivot and Golden. A **Match table** operation will be initiated if no Pivot or Golden record matches the record.
Match suspicious	N/A
Create new golden	N/A
Unmatch	N/A
Set golden	N/A
Set back golden	N/A
Set definitive golden	N/A
Unset golden	N/A
Not suspect	N/A
Merge	N/A
Automatic merge	N/A
Unmerge	N/A
Switch pivot	N/A
Remove from cluster	N/A
Remove and set golden	N/A

To be matched	
Add into cluster	N/A
Set merged (ignore)	N/A
Cancel ignore	N/A
Delete	The record's state is set to deleted and it remains in the current cluster.
Undelete	N/A
Align foreign key	N/A
Push out suspect (-1)	N/A

UI operation applied to single record

UI operation applied to single record	Description
Display metadata	Opens a pop-up window showing record metadata values. This service is also available at the table view level.
Manage cluster	Opens the cluster with all its records.
Show cluster	Displays all records contained in the cluster in the bottom window. The bottom window is a cumulative view. One to many clusters can be displayed.
Show cluster for merge	Displays all records contained in the cluster in the bottom window so that the 'Merge' button is readily available.
Clean up the list of 'Not suspect with'	Removes all records in the metadata tab's 'not suspect with' property. You can only see this service when at least one record holds the 'not suspect' designation with the current record (except Merged and Deleted).
Display merged records	Displays all merged records.
Fix relation records	Allows you to fix the relation records
Clean up merged field log	Removes all merged field information logged in Metadata. This service is only visible when at least one record merged into the current record (except Merged and Deleted).
Align foreign key of merged records	Updates foreign key references from Merged records to Pivot or Golden records. This service can be used to align the foreign key of a recursive merged record if the 'Was golden' field value is set to 'True' on the target record.
Hide cluster	Empties the bottom window.
Modify record	Opens the record tabular view.
Display relation records	When a relation match is configured, this service allows you to display the 'Relation table' records.

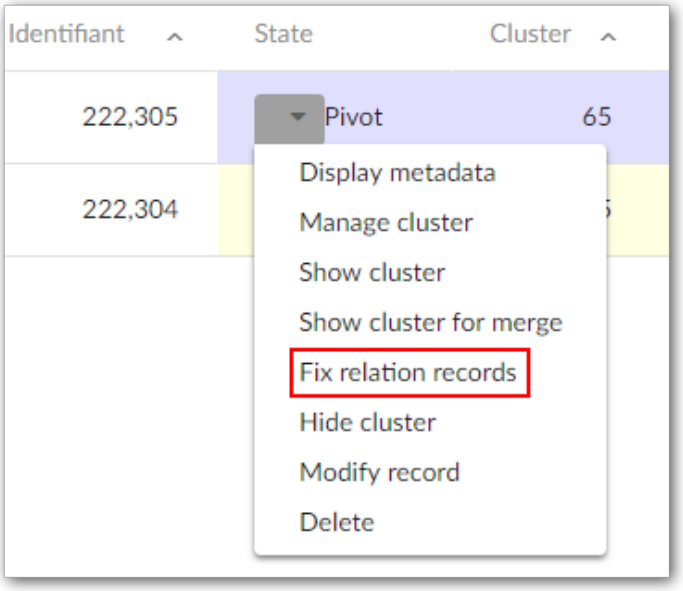
Fix relation records

The sections below show how to fix related records and which operations to apply.

Enabling the fixing of relation records

When a table holds the matching metadata type, the 'Fix relation records' service is available. This service allows you to arrange and clean records that are considered 'relation records' to this table across dataspaces and datasets. These records hold a direct or indirect (through a join table) relation to this table.

For instance, a join table 'EmployeeCompany' is used to define the relations between employees and companies. Once the 'Employee' table has been extended with the matching metadata type, the 'Fix relation records' service becomes available in the light and full matching view menu (see below, Stewardship UI).



When the table is under the matching process the 'Fix relation records' service is available for records with 'Pivot' and 'Golden' matching states only. The add-on process is then fully applied. When the table is not under the matching process, the 'Fix relation records' service is available for all matching states. In this situation the add-on process is no longer applied.

Operations to fix the relation records

You can apply one of these actions to the "relation record":

- 'Delete' — Physically removes the "relation record" when the relation table does not have matching metadata. If the relation table has matching metadata, a logical deletion is performed (the "relation record" moves to the 'deleted' state).
- 'Align' — Forces the "relation record" to link with the parent record. This situation can happen when a add-on operation has collected a set of "relation records" that are not linked to the "parent record".
- 'Set target' — allows you to select one "relation record" as the pivot record on which a merge process will be applied. The 'Merge relation' button appears if you have selected one record as 'Set target' and at least one other "relation record" exists.
- 'Reset' - Reverses action applied on the record.

Not suspect with

The 'Not suspect' service is only available for suspect records.

Once this service is executed on a record, the record is no longer considered as a suspect compared to its pivot or golden record. This has an effect not only for the current match procedure, but for future ones as well.

The add-on keeps a list of non-suspect records for each pivot and golden record. The 'Not suspect with' matching metadata is populated automatically by the add-on.

Records can manually be removed from this list. Records must only be added using the 'Not suspect' service.

This is an example of the 'Not suspect' mechanism.

oid (PK)	Social number	Last name	First name	Cluster	State	Score	Not suspect with
1	987-65-4320	Wood	Johnny	130	Pivot	100%	3,4,5
2	787-14-2277	Ewood	John	130	Suspect	80%	
3	987-65-4320	Woody	Nathalie	000	Unmatched	78%	
4	439-45-4309	Moody	Paul	133	Pivot	100%	
5	667-76-4547	Moowy	Thim	133	Suspect	80%	

The records 3, 4 and 5 will never be considered as a suspect against record 1 during matching. Record 4 is now a pivot, but was previously a suspect against record 1.

When record 1 subsequently changes states, it maintains its 'Not suspect with' list.

'Not suspect with' lists are not modified when referenced records are modified or logically deleted. When a physical deletion is executed then the list is updated to avoid the risk of breaking integrity.

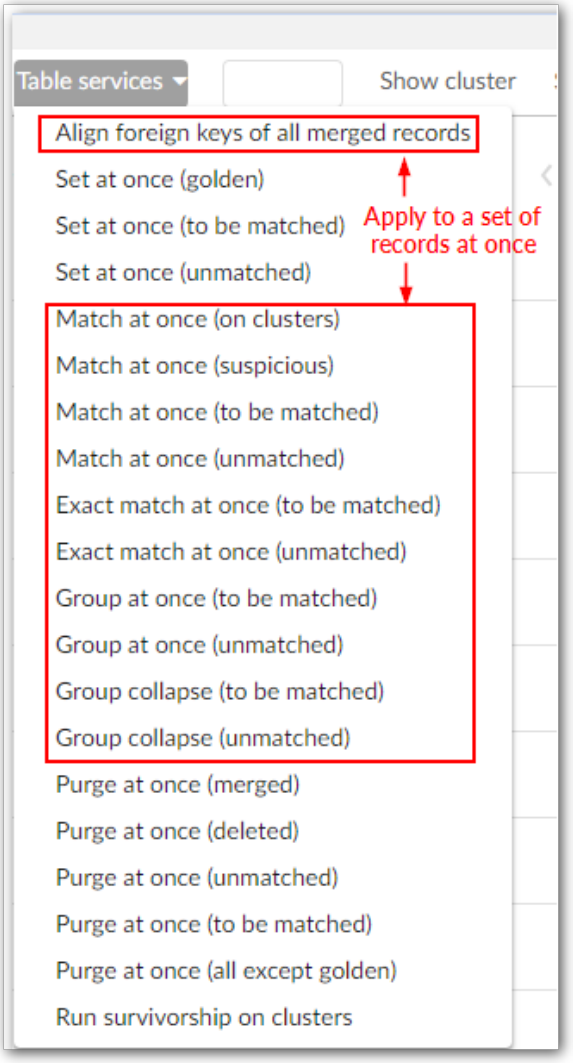
The score of the record against its 'Not suspect with' record is kept by the add-on so that future scoring can be compared with previous scores. If the difference between a former score and a new score is higher than the 'notSuspectComeBackSuspectThreshold' value (Process Policy) then the record is removed from the 'Not suspect with' list and its state returns to suspect.

Operations applied to a set of records at once

Types of operations

Type and context of use	List of operations
These operations are not available when a process policy has an incorrect configuration. However, they remain usable when the 'On matching Process' is set to 'No'.	Match at once (suspicious)
	Match at once (to be matched)
	Match at once (unmatched)
	Exact match at once (to be matched)
	Exact match at once (unmatched)
	Group at once (to be matched)
	Group at once (unmatched)
	Group collapse (to be matched)
	Group collapse (unmatched)
	Align foreign key of all merged records

Table 32: List of operations applied to a set of records at once



'Unmatched' records are created either when matching is deactivated for a table through the 'On matching process' parameter in the table configuration, or when the 'Remove from cluster' operation is performed.

'To be matched' records are created when importing bulk data. Then, a policy should be configured with the 'Is import mode' property set to 'Yes'. This allows the add-on to set every imported record to 'To be matched'.

'Suspicious' records are created when the simple matching configuration is used.

Match at once

Four operations are available for launching matches on a set of records at once:

- Match at once (on clusters)
- Match at once (unmatched)
- Match at once (to be matched)
- and Match at once (suspicious).

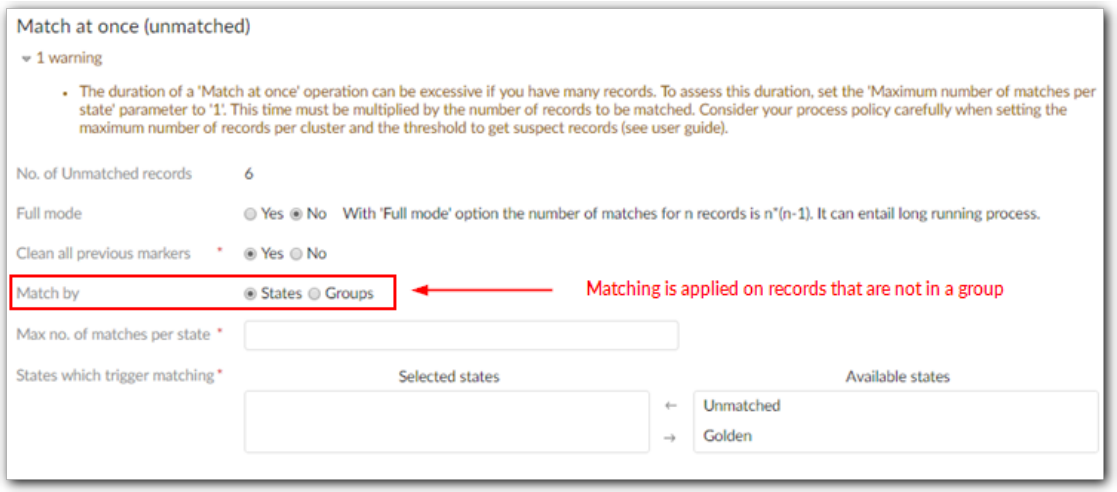
For the 'unmatched' and 'to be matched' records, two execution levels are available:

- either on the whole scope of a table for certain states,
- or a subset of records collected in 'groups' by the 'group at once' operations.

Special notation:	
✔	It is highly recommended to configure a special matching policy to use for the 'Match at once' operation ('For match at once' property in the 'Matching policy' table). If this policy is not configured, then the matching policy will be selected based on the context of records (the field values configured in table 'Matching policy context'), it can entail defect in the matching results.
	In case the 'Record creation matching state' and 'Record modification matching state' indicators of the EBX® Insight Add-on are configured, their automatic execution may affect 'Match at once' performance.

Match by states

By default, matching is applied solely on the scope unmatched records (to be matched, unmatched or suspicious depending on the operation).



Using a 'base' configuration allows you to define from which record states the 'Match at once' operation is driven. For example, if the base is 'Golden', then golden records are matched against the unmatched (or to be matched, suspicious) records. The order of the selected states is significant since matching is executed for the first state, then the second, etc.

The 'Maximum number of match per state' parameter is used to limit the number of records matched for each state configured in the base. It allows you to avoid executing the 'Match at once' operation on all existing records in the table.

Special notation:	
✔	The 'Match at once' operation ignores the records that are located in the groups.

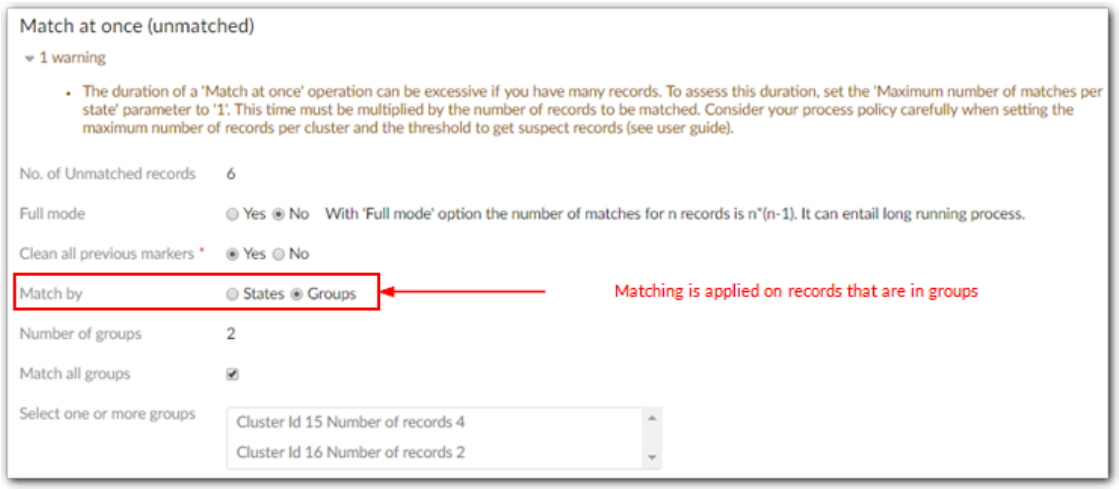
Example of match at once	List of operations
Match at once (unmatched) with base = golden, unmatched	All golden records are matched against all unmatched records. Once this match is terminated, the rest of the unmatched records are matched with each other.
Match at once (unmatched) with base = golden	Same as the previous example but the rest of unmatched records are no longer matched.
Match at once (unmatched)	All unmatched records are matched with each other.

Table 33: Examples of match at once operations

Special notation:	
✔	The 'Match at once' operations handle the records to be matched as a stack of records. In other words, a record is matched only one time during the whole process.
✔	**To streamline the process of the 'Match at once' operation, it is recommended to:** • Use only one matching algorithm level, meaning that the 'Second matching algorithm' property in the Matching policy used for the match at once operation, should be set to 'not defined'. • Deactivate the automatic merge process by setting the Process policy's 'On survivorship' property to 'No'. • Deactivate the 'Not suspect with' feature by setting the Process policy's 'On not suspect with' property to 'No'. • Use 'Full mode' with caution due to the exponential match operation execution: n*(n-1) match execution for n records.

Match by groups

It is possible to execute the 'Match at once' operation on a subset of unmatched records (or to be matched) that have been collected by the 'Group at once' operations.



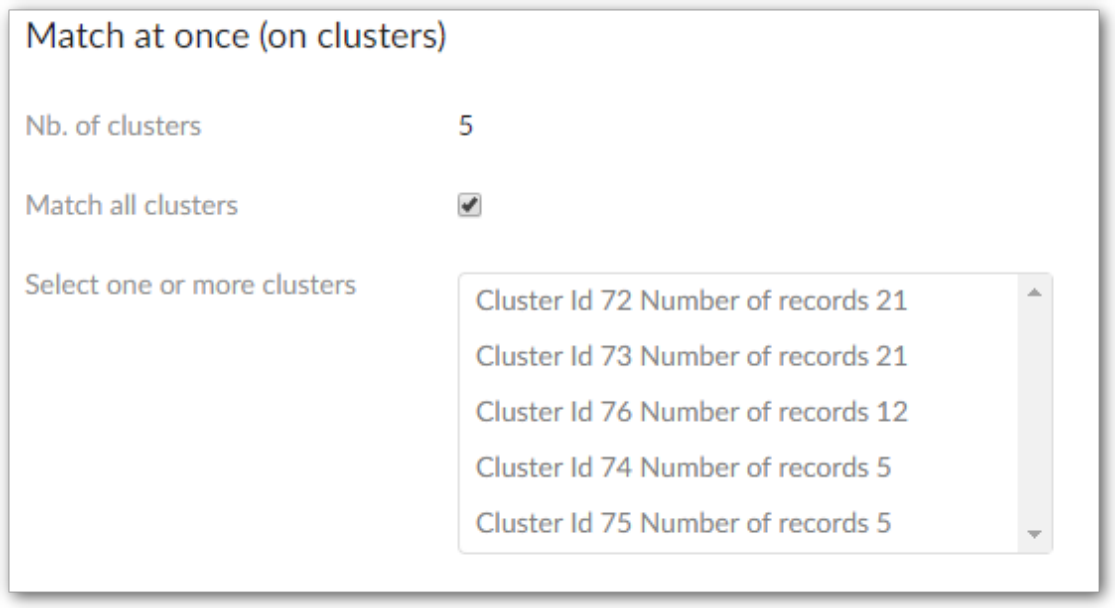
Special notation:	
✔	The match by groups option is available if there are existing groups (see the 'Group at once' operation to create groups of records).

When a large set of records must be matched, it is recommended to apply a rough matching policy to create groups. This requires a shorter execution response time. For example, using an Exact algorithm on a 'postal code' field to group customers by department. After that, the 'Match at once' operation is executed on the groups as explained above.

Special notation:	
✘	Parallel matching on several groups.

Match at once (on clusters)

You can execute the 'Match at once' operation on a single cluster or a set of clusters. Only clusters that have a ClusterID greater than 11 and do not contain a golden record display in the list of clusters.



Special notation:	
✕	Matching policy for 'Match at once' is not applied when executing 'Match at once (on clusters)'.

Match with the 'Full mode' option

When you execute the 'Match at once' service and enable the 'Full mode' option, a match is run against every record in the table, regardless of any existing 'Match at once' configuration settings. This means that any property values which make up a "base" configuration are ignored. The number of matches executed is n*(n-1) for n involved records. This improves matching result relevance, but increases execution time.

In 'Full mode', the 'Force orphans golden' option is active as default. In this mode, all lone suspect or lone pivot records in a cluster are forced to golden.

When this option is not active, a record is matched only if its state value is within the 'Match at once' base configuration's parameters.

Using match at once without the 'Full mode' option	
First step of the service A, Unmatched B, Unmatched C, Unmatched D, Unmatched	Match at once on A. The A record is a pivot and matched with B, C, and D. If a record becomes Suspect it is no longer used for the match. This is considered a "stack mechanism" to facilitate rapid matching. Applying an additional match on a Suspect record could entail new duplicates that are ignored in the current 'Match at once' operation.
Second step of the service (executed automatically) A, Pivot B, Suspect C, Unmatched D, Unmatched	The next match is done on C. The B record is not used to launch a match. This stack process streamlines the response time and results in a fast matching process.

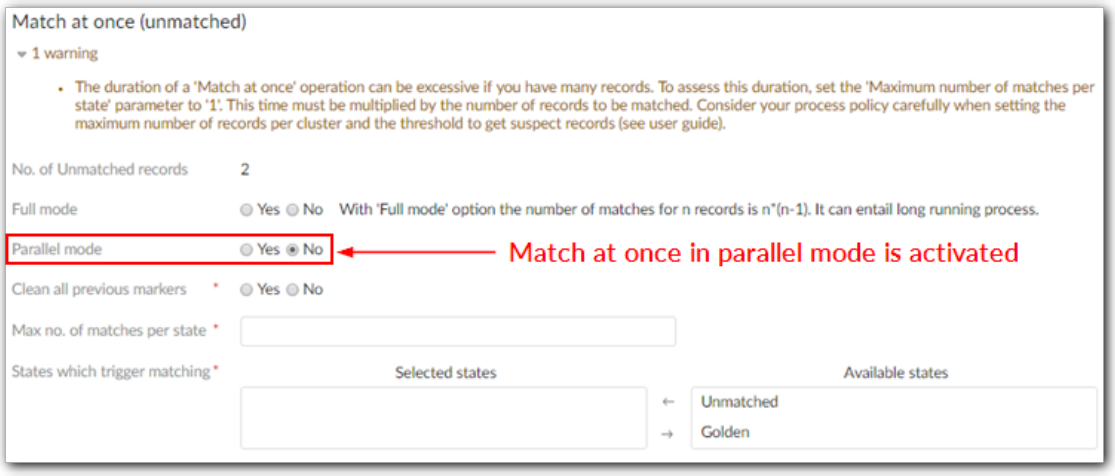
Using match at once with the 'Full mode' option	
First step of the service A, Unmatched B, Unmatched C, Unmatched D, Unmatched	Match at once on A. In 'Match at once' with the 'Full mode' option enabled all records in the table are matched one time and only one time, even if they are suspect, pivot or golden. The base configuration is not used.
Second step of the service (executed automatically) A, Pivot B, Suspect C, Unmatched D, Unmatched	Next match is done on B by applying the service 'Match table' on the suspect.

Special notation:	
✕	The 'Unmatched' records created during match at once with 'Full mode' option are matched against other records in the new cluster.

Match at once in 'Parallel mode'

When you activate Parallel mode, the add-on divides records into groups and executes matching on these groups simultaneously. This method can reduce matching response time. The options when running Match at once in parallel mode are defined in 'Process policy' table's 'Options for Parallel

Match at once' tab. Please refers to *Process policy* section for more details. Note that the Parallel mode is hidden when replication is activated (only in commit mode).

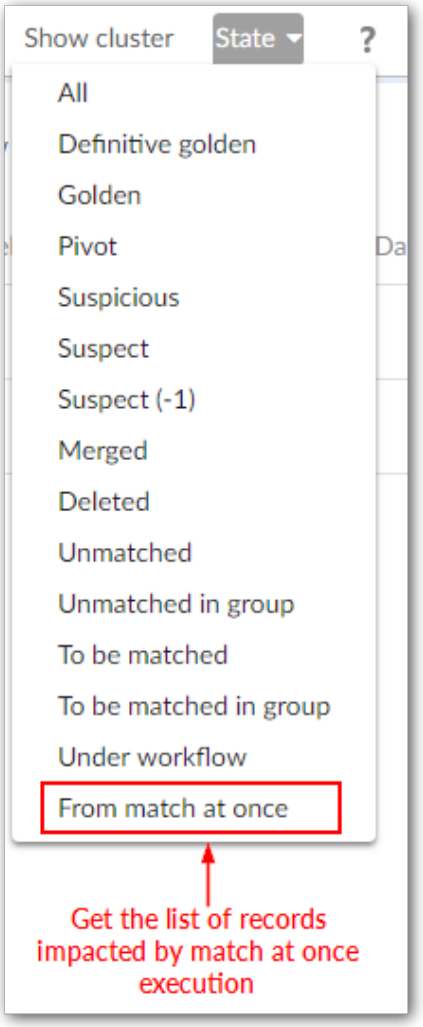


> **Note**
>
> The parallel matching option does not display on a historized table, or an existing replication table on submit mode.

Special notation:	
✗	Match at once parallel does not work on a historized table or a table which is configured a 'On submit' mode replication.
✔	Parallel matching is in beta mode, it is highly recommended that you work on a working dataspace instead of real data.

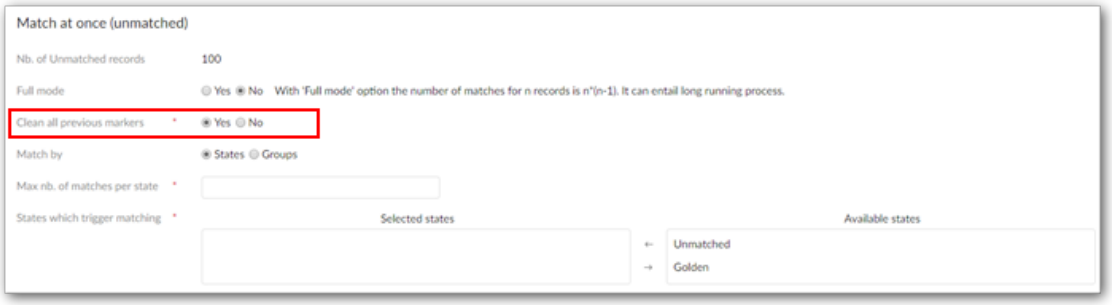
Follow the result of a match at once execution

The 'Clean all previous markers' parameter is used to retrieve records that have been handled by the match at once execution.



When a record's state is modified during a 'match at once' operation, the add-on flags it. Then, it is possible to get the list of all records impacted by match at once operation execution(State button).

A record is considered as stemming from a 'match at once' operation until it is modified by any other operations or if the 'Clean all previous makers' option is set to 'Yes' when executing a 'match at once' operation (figure below).



Response time

When a table contains a large set of records, it is recommended to apply an incremental matching procedure (see *Incremental matching procedure* section). The following table describes three properties that have an impact on a match at once operation's response time.

Property	Description
Max number of records in cluster (Table configuration)	When a set of records contains many suspect records, setting the 'Max number of records in cluster' property to a higher value improves response time. When the 'Max number of records in cluster' property is set lower and the 'Maximum number of match table for each state' is higher, a match operation creates more clusters containing fewer records.
Stewardship min score (Process policy)	The lower this threshold is, the faster the match at once operation will be. The pace of feeding clusters depends on the ability to make a record suspect as quick as possible. If this threshold is too high it means that likelihood of a record to be a suspect is lower. This configuration can raise many matches of the same record against many other records in the table because of a lack of suspect detection.
Maximum number of match table for each state ('Match at once' operation)	The match at once operation is executed as many times as records are available to match. When an unbounded value is used for this property, it means that all records in the table will be impacted by the operation.

Table 34: Properties having an impact on the response time of match at once operations

This table gives some examples of possible response times for the match at once operation. The assumptions are as follows:

- 100,000 records are set to unmatched.
- Execution of the 'match at once unmatched' operation.
- Match on one field.

Note: obviously the response time depends on the quality of the execution platform as well. For this reason the following table remains just an example.

Property	Example
Max number of records in cluster=5 Stewardship min score=15% Maximum number of match table for each state=10	Quick match to get an idea for response time on a limited matching coverage Result in: 5 seconds
Max number of records in cluster=5 Stewardship min score=15% Maximum number of match table for each state=100	Higher number of records to match. The response time should be about 10 times the previous one Result in: 40 seconds
Max number of records in cluster=5 Stewardship min score=15% Maximum number of match table for each state=10,000	For 10,000 records it should take about one hour (40 seconds * 100)
Max number of records in cluster=50 Stewardship min score=15% Maximum number of match table for each state=100	With the 'Max. number of records' property set to a higher number, suspect record collection time improves. Example: 3 minutes
Max number of records in cluster=50 Stewardship min score=15% Maximum number of match table for each state=10,000	The same configuration as the previous one but applied to 10,000 records should take about 5 hours (3 minutes * 100)

Table 35: Example of possible response time for a table with 100,000 records

Incremental matching procedure

Because there are many parameters that influence the 'Match at once' service execution time (see *Response time* section), it is recommended to run matching on successive subsets of records. This is achieved on all records directly in the table, or by creating groups of records (operations Group at once).

No use of groups of records

The following table shows an example for the execution of a 'match at once' operation applied on a set of 400,000 records.

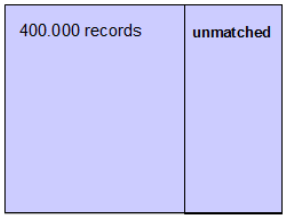
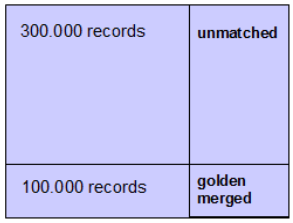
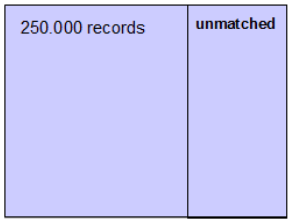
Records	Matching Execution
	**Matching stage #1** All table records are set to 'unmatched' when importing data for the first time. On this table, a 'match at once (unmatched)' operation is executed with a limit of records set to 100,000. The stewardship for these records is then applied. Some of these records move to merged and can be deleted. Other are set to golden, pivot and suspects.
	A second 'match at once (unmatched)' operation is executed on the next 100,000 records. Only records with an unmatched state are used, meaning that all records that have been analyzed during the first matching execution are ignored.
	**Matching stage #2** After the execution of four 'match at once (unmatched)' operations by sets of 100,000 records, there are 150,000 records holding merged and deleted states. To execute a second matching stage, all other records must be set to 'unmatched' (see the 'set at once' service). When applied to the 250,000 remaining records, the second matching execution stage is performed. The 'match at once (unmatched)' operation is directly executed on the scope of all 250,000 records, or on a smaller set depending on the matching strategy.

Table 36: Example of an incremental matching procedure

Use of groups of records

Rather than matching 100,000 records directly (see *No use of groups of records* section) with a fine grained matching policy, it is recommended to first apply a rough matching policy that groups all unmatched (or to be matched) records without changing their states. This rough policy can rely on the fastest matching procedure such as an Exact algorithm with one field only.

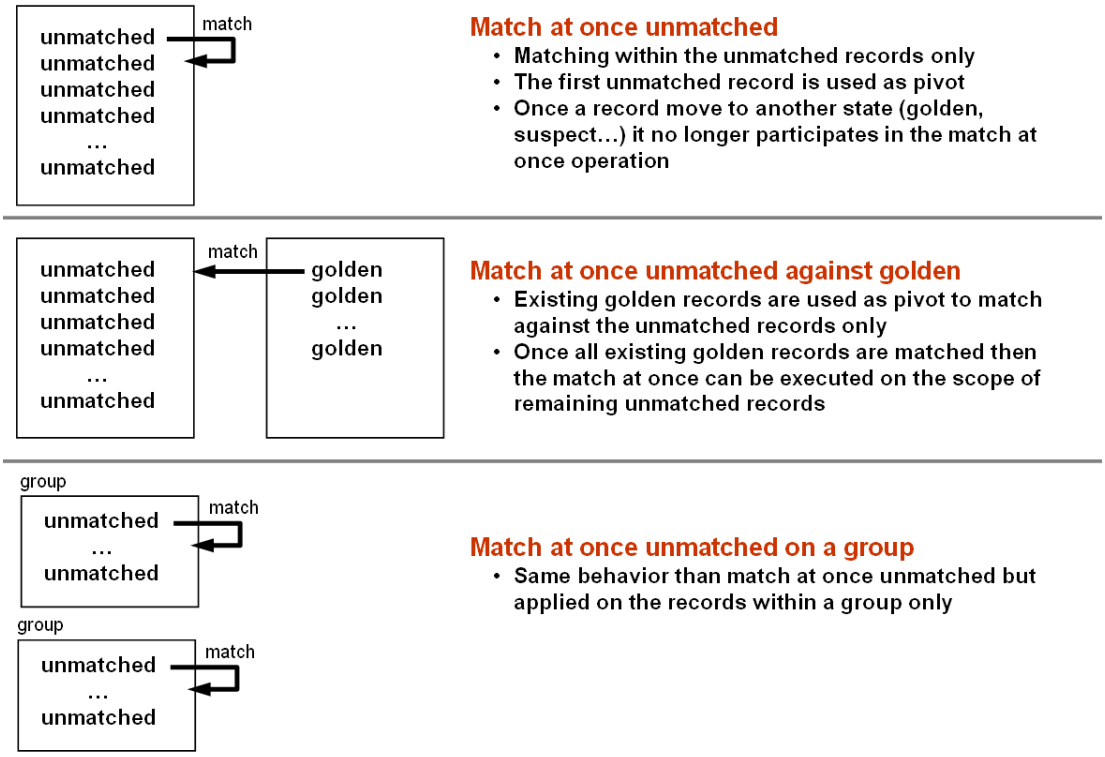
Once groups are created, then the match at once operation can be executed on every group to keep the response time to a minimum.

To create a group the 'Group at once' operations are available to apply to 'unmatched' or 'to be matched' records (see next section).

Synthesis of possible approaches to match large volumes of data

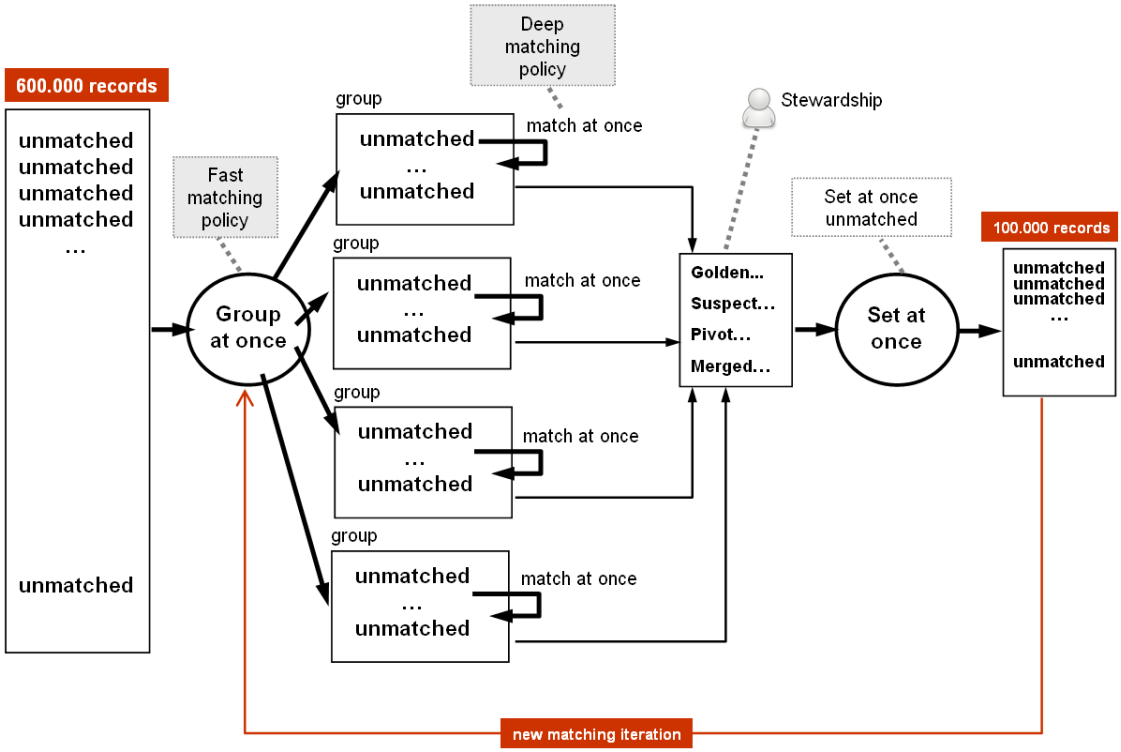
Different types of match

The same process can be applied to 'to be matched' and 'Suspicious'.



The whole process of matching large set of data

The same process can be applied to 'to be matched' and 'Suspicious'.



Special notation:	
✔	When you are dealing with a large volume of data, it is also possible to use the 'Exact match at once' operation to apply a direct data query in order to get groups of records based on exact data values (see *Exact match at once* section).
✔	You can filter data at the time the matching policy is executed. Fuzzy matching is then applied to the subset of records retrieved through the filter. Please see the 'Filter by' property.

Exact match at once

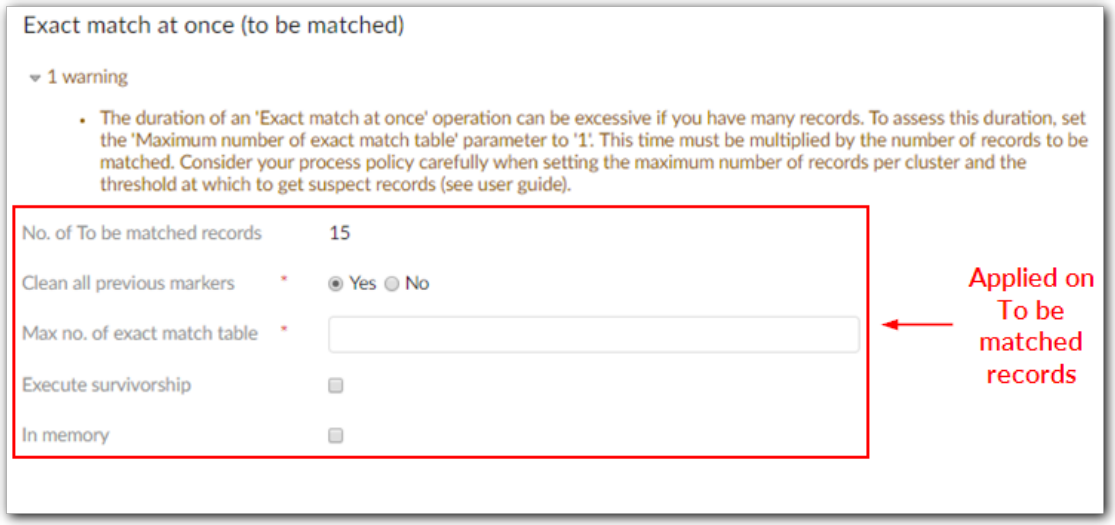
The 'Exact match at once' operation is executed on 'To be matched' or 'Unmatched' records.

This operation applies a full 'exact' matching based on the matching fields that are configured in the active matching policy. This means that the algorithms configured in the matching policies are bypassed to apply an exact matching.

Similar to the 'Exact' algorithm in the matching field configuration, the 'Exact match at once' operation, runs a direct data query that is not based on a fuzzy algorithm. Therefore the response time is dramatically improved but the matching results cannot be mixed with a fuzzy search. When many fields are configured to use matching then the 'Exact match at once' operation computes the record's score by using a direct 'and' operator.

The 'Exact match at once' operation creates clusters of records that match.

The records used to match come from the cluster reserved for the 'To be matched' records (or 'Unmatched') and the records that are pivot, suspect and golden. In other words, 'To be matched' and 'Unmatched' records that are already grouped are not used. The suspicious records are not used.



The 'In memory' option forces the execution of the matching in a full memory mode for which all data is managed within a unique transaction. It improves the response time dramatically (more than ten times faster) but requires more memory use. This option is strongly recommended for large volume of data with million of records.

At the end of the execution, you can still have 'To be matched' or 'Unmatched' records when they have empty values for the exact matching fields or they are located already in existing groups.

Special notation:	
✔	'Exact match at once' is a fast matching operation that can be used on large set of records to easily create initial clusters of records that share exact values on certain fields.
✔	The 'Exact match at once' operation does not work in conjunction with the 'Relation match' operation. With a direct foreign key relationship, the exact match is applied using the foreign key raw value not its label.
✔	Even if a matching policy includes a 'Filter by' configuration, the 'Exact match at once' operation ignores it. This is because 'Exact match at once' already behaves as a filter.
✘	The 'Exact match at once' operation does not support matching on multi-valued fields.

Group at once

The 'Group at once unmatched' operation ('to be matched') allows you to group unmatched or 'to be matched' records. Contrary to matching execution, records that are grouped do not move to another state such as golden, pivot or suspect. They keep their states unmatched (to be matched).

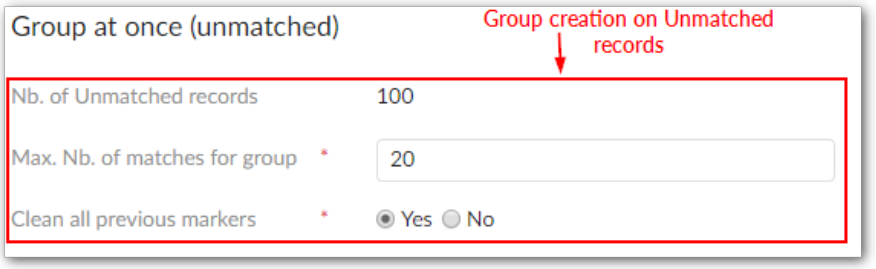
Special notation:	
✔	The maximum number of records in a group is defined by the 'Max number of records for group' property in the 'Table' configuration. It is recommended that this property is configured to maintain manageable groups sizes that depend on the context of use. For example, to have a chance to group 100,000 records in a minimum of 100 groups, the 'Max number of records for group' property must be set to '1,000'.

Groups creation

It is possible to configure a special matching policy for group at once execution ('For group at once' property in the Matching policy table). If this policy is not configured, then the default is used.

Most of the time, the group at once operation is used to create groups of records on which a further execution of the match at once operation will be applied. Then, the groups can be created from a rough matching policy (Exact algorithm, one matching field) that will streamline the execution response time.

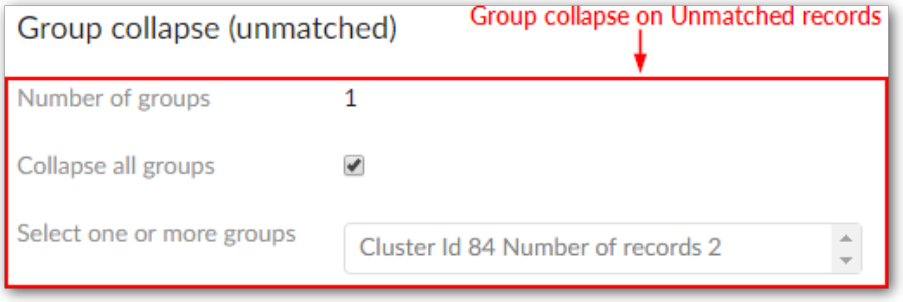
The group at once operation takes into consideration all unmatched records (to be matched) in the table, whatever their cluster localization ('000' for unmatched, '002' for to be matched), including records already grouped. In other words, the execution of the group at once operation reshapes the existing groups for unmatched or to be matched.



The 'Maximum number of matches for group' property allows you to limit group at once operation match execution.

Groups collapse

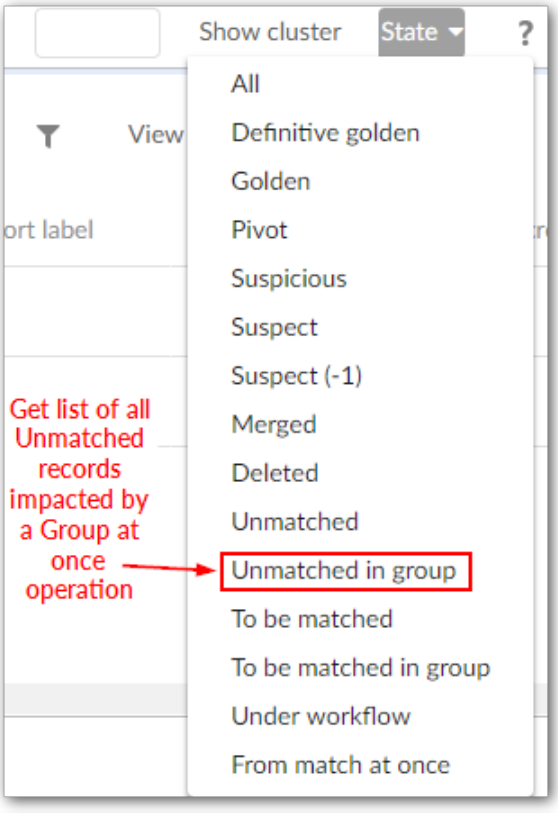
It is possible to collapse a group by using the 'Group collapse (unmatched)' operation (to be matched). Then, all group records move to the reserved cluster '000' (unmatched) or '002' (to be matched) and the group dissolves.



Note: the number of records in a group is computed at the time of group creation. In order to keep the original information about the group size, it is not updated after the creation.

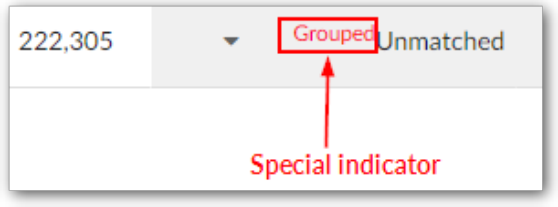
Follow the result of a group at once execution

The 'Clean all previous markers' parameter is used to retrieve records that have been handled by the match at once execution.



When a record is grouped during a 'group at once' operation, the add-on flags it. This allows you to retrieve a list of all records impacted by a group at once operation (button State, operations 'Unmatched in group' and 'To be matched in group').

A record is considered as stemming from a 'group at once' operation until it is modified by any other operation, or if the 'Clean all previous markers' option is set to 'Yes' when executing a 'group at once' operation (figure below). Then, a special indicator displays as highlighted below.



Align foreign keys of all merged records

Any record that still has a relationship with another record after it has been merged has an integrity issue. Since merged records still remain in their tables, built-in validation performed by EBX® cannot be used to resolve this issue.

The EBX® Match and Merge Add-on provides two services to realign foreign keys:

- 'Align foreign key of merged record' applied to a golden or pivot record uses all merged records related to the pivot or golden record to realign (see *Operations applied to single record* section).
- 'Align foreign key of all merged records' applied to a table uses all merged records in the table to realign.

These services check every foreign key in relation to the table managed by the add-on, including ones in the table on which the service is executed (in case of self-referencing relationships). The scope of these services includes all related dataspaces and datasets.

When a foreign key is linked to a merged record, the add-on automatically updates it with the correct target record in the appropriate dataspace and dataset. To achieve this process, the service relies on the matching metadata to keep track of the target record for each merged record. Note that the setting of a table's **Relationship management** property can affect this behavior as follows:

- When the property is set to **None**, no action is taken.
- When the property is set to **Manual**, alignment follows the selections in the **Merge view**.

To verify whether the orphan relationships have been fully repaired in a repository, all obsolete merged records must be purged. Once these records are no longer physically present in the tables, built-in validation performed by EBX® can be relied upon to identify remaining orphan relationships.

As the 'Align foreign key' services can no longer be used once merged records have been purged, you can employ a child dataspace for EBX® validation checking. In the child dataspace, it is then possible to purge the merged records in isolation and correct any validation errors reported by EBX®. Once the results have been validated in the child dataspace, the corrections alone can be merged back into the parent dataspace, thereby leaving merged records untouched in the parent dataspace allowing 'Align foreign key' to be run again in the future.

Logical deletion of records

The add-on relies on a logical deletion of records (deleted state). From the add-on's point of view, it is better to keep the record deleted until a definitive decision for its physical deletion is made. At any time, a deleted record can be reset to unmatched if needed (undelete).

When a record moves to the deleted state, integrity is a concern if other records hold a link to this record. Since the record is not yet physically deleted, the EBX® validation procedure does not raise errors.

To enable EBX® to enforce the validation you need to physically remove all records with the deleted state. This can be achieved either by using the usual EBX® 'delete' service, or by executing the 'Purge at once (deleted)' operation supplied by the add-on.

Running survivorship on all clusters

From the **Table services** menu, you can execute the **Run survivorship on clusters** service to run survivorship on selected clusters. Only clusters that have pivot, golden and merged records will be taken into account. When you run this service, the add-on executes survivorship to merge data from merged records to golden, or pivot records.

Running a match from a table

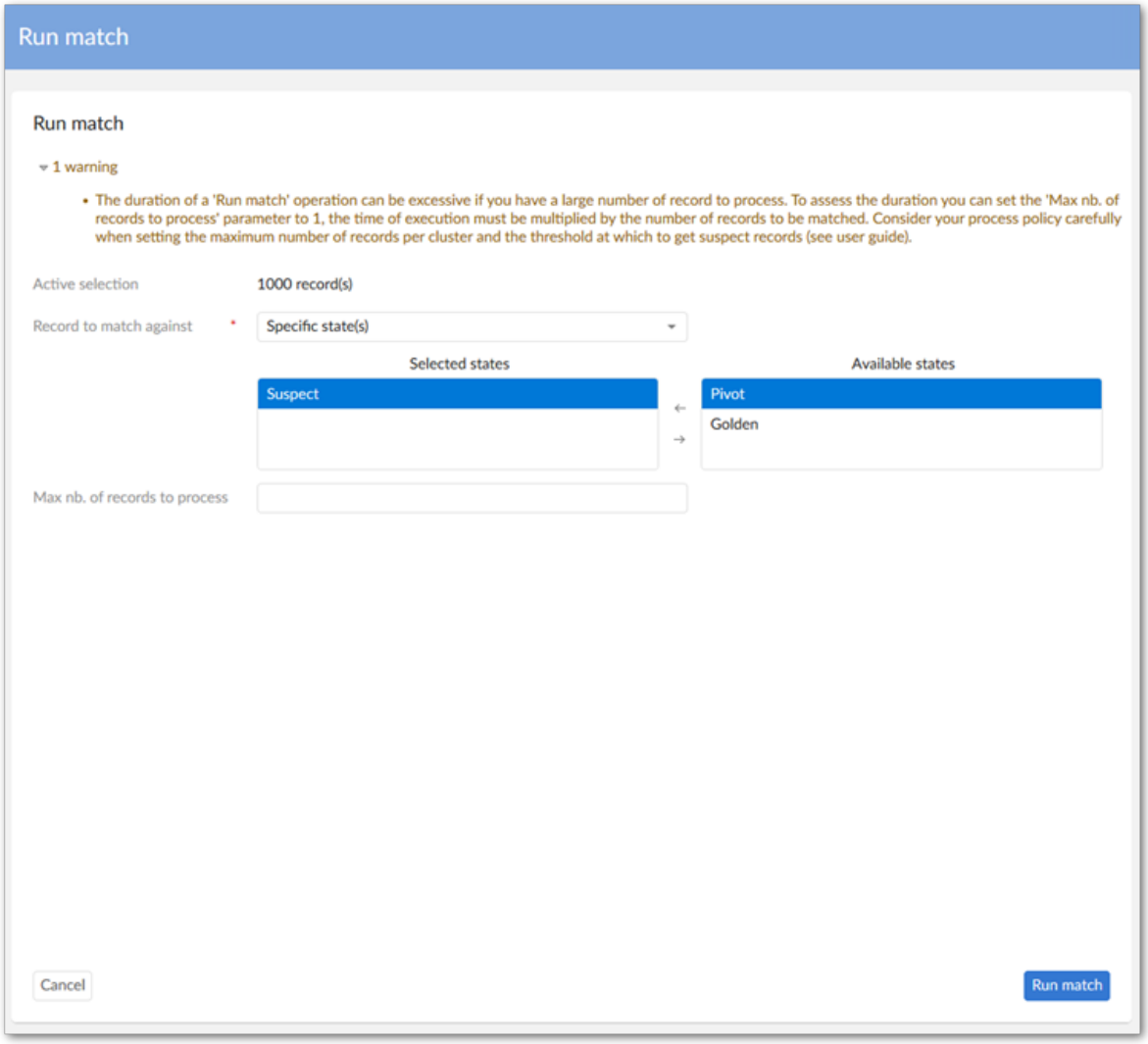
You can execute a match from a table's **Actions** menu by selecting the **Run match** service. This service allows you to select one or more records to include in a matching operation. If you run the service without making any selections, matching runs on the entire table.

When you run the service and:

- a single record is selected, the add-on executes a match against the selected record and immediately displays the **(Light) Data quality stewardship view**.
- multiple records are selected, the add-on displays the **Run match** screen where you can specify how you want the match to execute. The match executes only on the selected records and the add-on redirects you to the **(Full) Data quality stewardship** view.
- no records are selected, the **Run match** screen displays and you can choose how you want the match to run. When you run the match, it executes on the entire table. Once the matching process completes, the add-on opens the **(Full) Data quality stewardship** view.

In the **Run match** screen, the **Active selection** field shows the number of currently selected records. You can use the **Record to match against** field to define which records the add-on will process against the active selection. The **Specific states** option allows you to select from available states. The matching operation then runs on records with the selected state. When you choose the **Active selection** option, the match runs only actively selected records (against each other). The **Max no. of selected records** property limits the number of records from the active selection that will be processed.

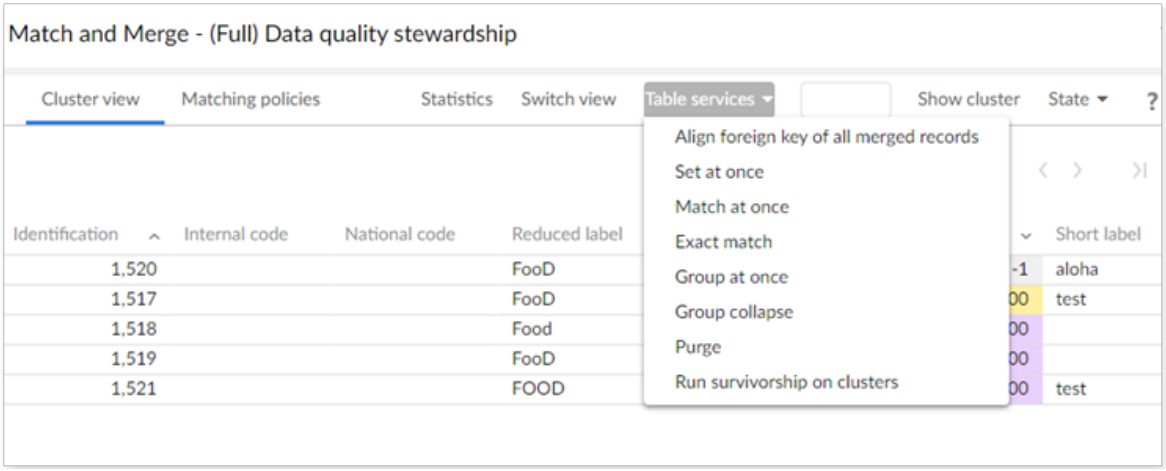
The following image shows the **Run match** screen:



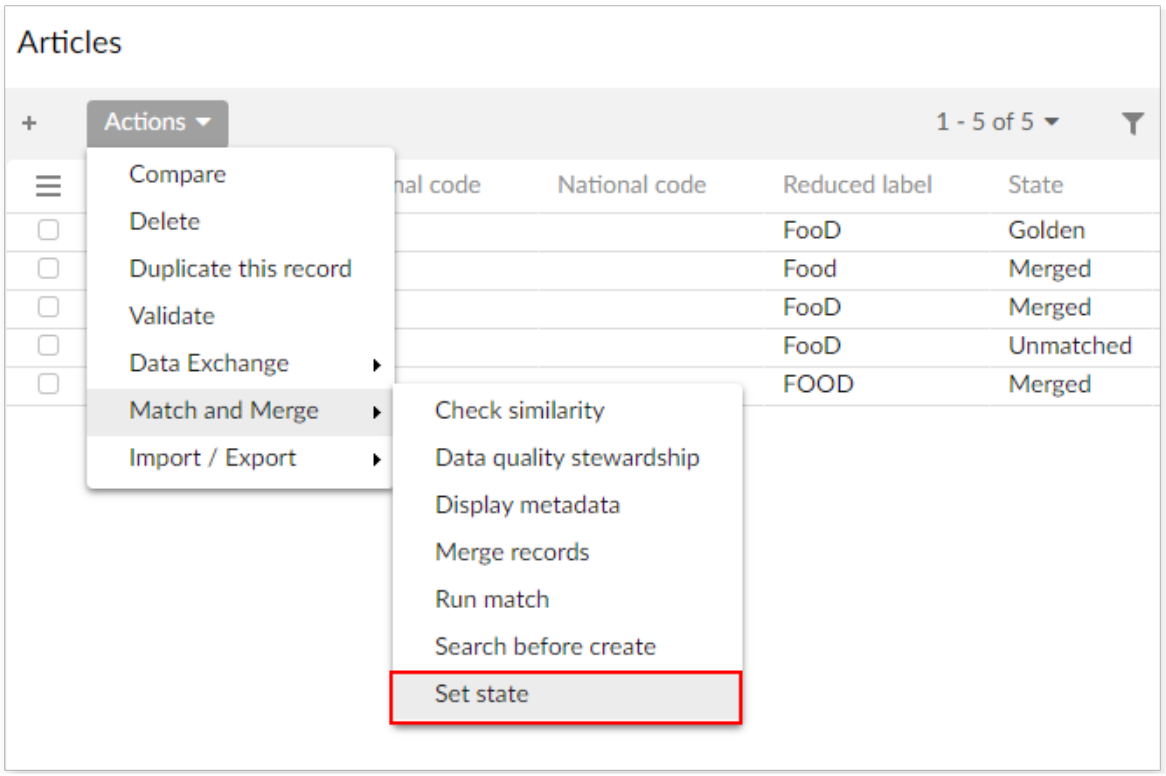
Setting record state

The add-on allows you to set the matching state for records as follows:

- When in the **Data quality stewardship** screen, you can set all records in a table using the **Set at once** service options.



- When viewing a table, you can select records and use the **Set state** service. Note that if you do not select any records, the service applies to all records in the table.



CHAPTER 7

Searching before record creation

This chapter contains the following topics:

7.1 Overview

If an administrator has enabled the option, you can use the add-on to search for potential duplicates prior to creating a record. You can include each field configured to use matching when searching for duplicates. The add-on will present you with a list of possible duplicate records and you can choose whether to update an existing record, or create a new one.

7.2 Enabling search before create

If you are an administrator you can enable the **Search before create** option at the process policy level. If multiple matching policies are applied to a table, you can also specify on which policy the add-on applies the feature.

To enable the option to search for potential duplicates before record creation:

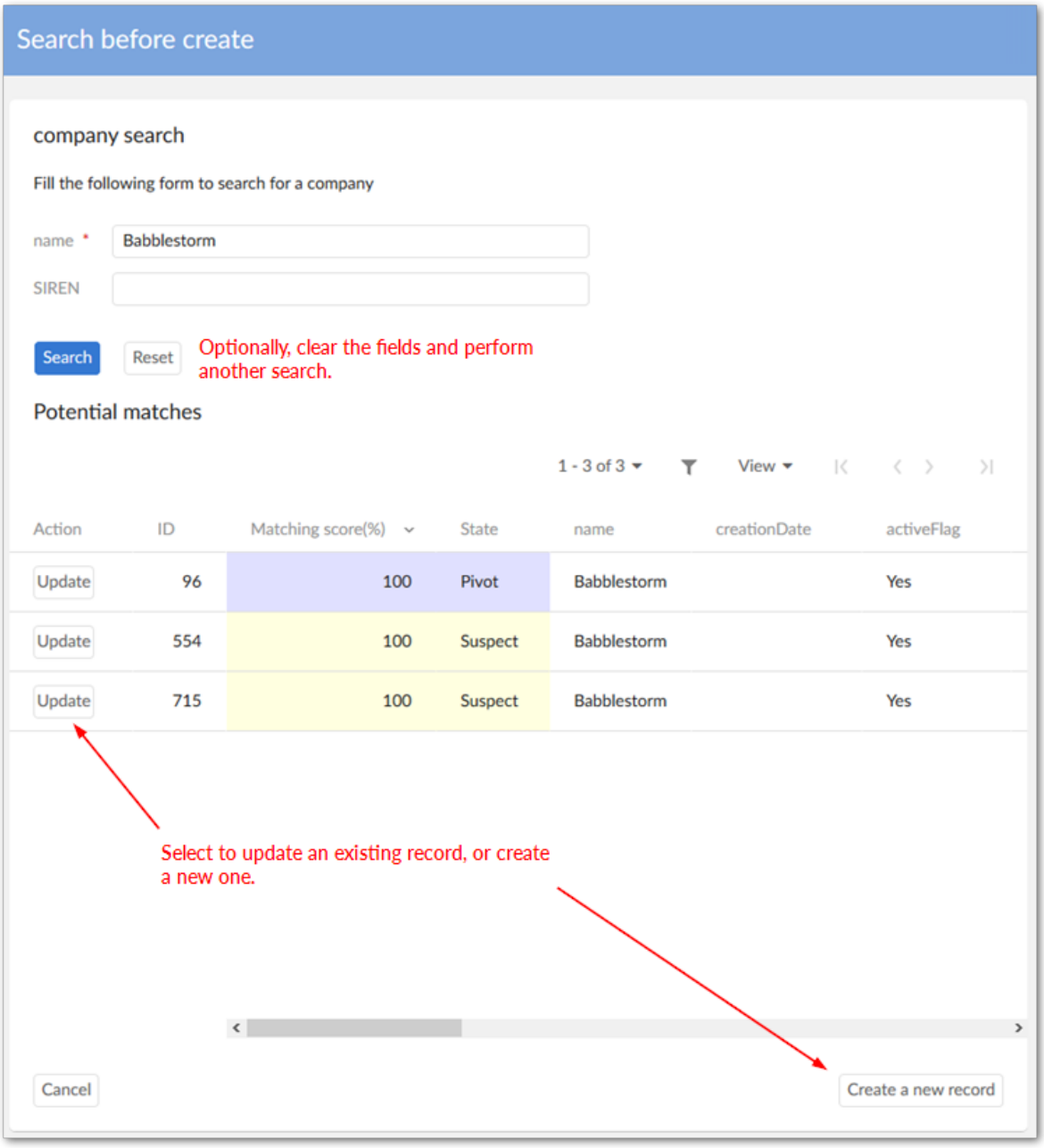
1. Navigate to *Administration > Data quality & analytics > TIBCO EBX® Match and Merge Add-on* and locate the desired policy in the **Process policy** table.
2. Set **Search before create** to **Yes**.
3. If the table you want to enable this feature for has multiple matching policies applied, use the **Matching policy** table's **For search before create** option to specify which matching policy the add-on will use for the feature.

7.3 Using search before create

Once enabled by an administrator, you can search for potential duplicates prior to creation using the following steps:

1. From a table's **Actions** menu, select *Match and Merge > Search before create*.
2. Fill in any required fields and any optional fields with information from the record you want to create and select **Search**.

3. After the search completes you have the options to select:
   - **Reset** to clear the search form's fields.
   - **Update** in the **Potential matches** panel to change values in an existing record.
   - **Create a new record** to add a new record to the table. In the creation form that displays, any fields that included searched values will be populated with these values.



CHAPTER 8

Data Cleansing

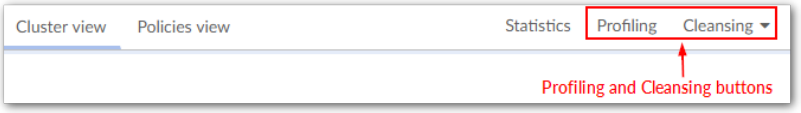
This chapter contains the following topics:

8.1 Introduction

The cleansing process involves detecting and correcting inaccurate or missing data, and deleting obsolete records. You can activate profiling mode to create reports on data quality defects. Based on this report, you can execute the appropriate cleansing procedures to improve data quality.

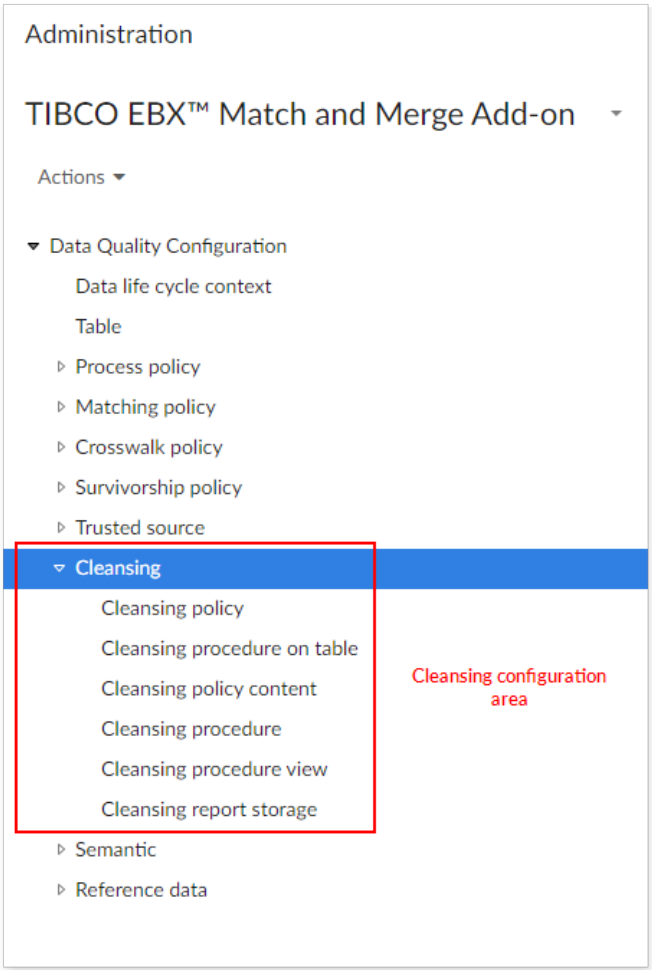
8.2 Enabling cleansing

To enable the data cleansing feature for a table, its **Use cleansing** property needs to be set to **Yes**. This property is available in the 'Table' domain located in the **TIBCO EBX® Match and Merge Add-on** dataspace under the **Administration** tab. You also need to configure a **Cleansing policy** containing one or many **Cleansing procedures**. Once you have completed configuration, the **Profiling** and **Cleansing** buttons display in the UI, as illustrated below.



8.3 Configuration

The 'Cleansing' domain is used to configure cleansing procedures and policies applied to tables and is available from the **TIBCO EBX® Match and Merge Add-on** dataspace located under the EBX® 'Administration' tab.



Cleansing policy

The 'Cleansing policy' table displays a list of cleansing policies and their corresponding tables. From this table, you can specify whether or not a cleansing policy is active on a given table. Additionally, you can associate a cleansing policy with a context by specifying a dataset and a dataspace.

The same table can be configured multiple times with different cleansing policies and contexts. But, only one cleansing policy can be active per context.

The 'Cleansing policy content' table lets you choose which cleansing procedures are attached to a cleansing policy.

Source (logical name: **CleansingPolicy**).

Properties	Definition
Code	Any naming convention is possible.
Name	Any name is possible.
Short description	Description of the cleaning policy
Long description	Long description of the cleaning policy
Table	Table on which the cleansing policy is configured.
Is active	If 'Yes': the cleansing policy is used by the add-on. If 'No': the cleansing policy is not used by the add-on.
Is context	If 'Yes': the cleansing policy requires a context and you must use the 'Data life cycle context' property to specify the context applied to this cleansing policy. If 'No': the cleansing policy does not require on a context.
Data life cycle context	The context used by this cleansing policy. When 'Is context' is set to 'Yes', then you provide a context by specifying a dataspace and dataset.

Table 37: Cleansing policy

Cleansing procedure on table

This feature allows you to associate cleansing procedures with specific tables and fields.

Source (logical name: **CleansingProcedureOnTable**).

Properties	Definition
Cleansing procedure definition	A drop-down list that contains the available cleansing procedures. Any cleansing procedures listed with the prefix of '[ON]' are provided by the add-on as ready-to-use procedures. You can develop and customize your own cleansing procedures via the API.
Table	Table on which the cleansing procedure applies.
Field	Field in the specified 'Table' on which the cleansing procedure applies.
Additional field(s)	When the cleansing procedure is able to manage more than one field, then the additional fields can be specified here.

Table 38: Cleansing procedure on table

Cleansing policy content

The 'Cleansing policy content' table specifies which cleansing procedures are attached to cleansing policies.

Source (logical name: **cleansingPolicyContent**).

Properties	Definition
Cleansing policy	Reference to a cleansing policy.
Cleansing procedure on table	Reference to a cleansing procedure that has been defined on a table.

Table 39: Cleansing policy content

Cleansing procedure

The 'Cleansing procedure' table lists fully defined cleansing procedures. Cleansing procedures can be defined by referencing a cleansing procedure implementation that has previously been published. Although certain predefined cleansing procedures are provided with the EBX® Match and Merge Add-on, you can use the API to implement your own bespoke cleansing procedures.

Source (logical name: **cleansingProcedureDefinition**).

Properties	Definition
Code	Any naming convention can be used except the prefix '[ON]' which is reserved by the add-on for predefined cleansing procedures.
Name	Any naming convention can be used.
Cleansing procedure	List of implemented cleansing procedures that have been published through the API provided by the add-on. You can create multiple cleansing procedures based on the same cleansing procedure implementation. For instance, the '[ON] Missing fields values' cleansing procedure can be used to create two definitions of cleansing procedures with different values for the input parameters.
Input parameters	A list of customized parameters. The available parameters depend on the cleansing procedure configuration.
D.E.C. type	D.E.C. on which the cleansing procedure is applied. For example, this can be a 'Table' or a 'Field'.
Can be used in profiling procedure	If 'Yes': the cleansing procedure can be used in the profiling operation. If 'No': the cleansing procedure is not usable for the profiling operation.
Can be used for many tables	If 'Yes': the cleansing procedure can be attached to many tables at one time. If 'No': the cleansing procedure cannot be attached to many tables at one time. The 'cleansing procedure on table' feature allows you to attach a cleansing procedure to a table.
Can be used for many fields	If 'Yes': the cleansing procedure can be attached to multiple fields simultaneously. If 'No': the cleansing procedure cannot be attached to multiple fields simultaneously. The 'cleansing procedure on table' feature allows you to attach a cleansing procedure to a field.
Save last value only	You can either delete the existing results in the Cleansing report tables or keep them over time to make possible data analytics processes. If 'Yes': All existing Cleansing results for this Cleansing procedure and current table are deleted before the procedure is executed. If 'No': Keep the previous results in the Cleansing report tables. The execution of the procedure enriches the results over time and allows you to apply data analytics processes.
Cleansing operation definition	

Properties	Definition
Every cleansing procedure can publish several 'Cleansing operation(s)' that are used to fix data quality defects. For instance, the cleansing procedure '[ON] Missing fields values' publishes two cleansing operations that enable management of missing data values. The "Replace all" operation allows you to enter a value that will be used to fill all missing fields simultaneously. The "Replace manually" operation allows you to enter missing values one at a time.	
Name	Name of the cleansing operation that displays in the UI.
Cleansing operation	Name of the cleansing operation coming from the implemented procedure.
Input parameters	A list of customized parameters. The available parameters depend on the cleansing procedure configuration.
Is matching active	If 'Yes': matching is executed each time a record has been fixed. If 'No': matching is not executed after a record has been fixed.

Table 40: Cleansing procedure

Cleansing procedure view

Every cleansing procedure is associated to a data view name that is used to filter the 'Big data report cleansing' table.

Source (logical name: **cleansingProcedureView**).

Properties	Definition
Cleansing report storage	dataspace and dataset inside which the cleansing report is stored
Cleansing procedure	Reference to the cleansing procedure.
Data view	Name of the data view (publication name) used to filter the results of this cleansing procedure in the table 'Big data report cleansing'.

Table 41: Cleansing procedure view

Cleansing report storage

dataspace and dataset where the cleansing report data is stored

Source (logical name: **cleansingReportStorage**).

Properties	Definition
Code	Code of the cleansing report storage.
dataspace	dataspace where is located the cleansing report storage
dataset	dataset where is located the cleansing report storage
Is active	Use by default when 'Is active' = 'Yes'. Only one storage can be active at a same time

Table 42: Cleansing report storage

It is possible to declare other custom cleansing report storage environments by following this procedure:

- Create a data model with any names.
- Include the data model 'ebx-addon-daqa-cleansing-report-metadata.xsd' from module 'ebx-addon-daqa'.
- Create a table with the physical name 'BigDataCleansingReport' (this name cannot be changed).
- Inside the BigDataCleansingReport table, create two groups 'CleansingReportGenericFields' and 'CleansingProcedureFields'. The name of these two groups cannot be changed. All the fields inside these two groups are reused from the included data model 'ebx-addon-daqa-cleansing-report-metadata.xsd'.
- Create another table with the physical name 'BigDataProfilingReport' to store profiling data (this name cannot be changed).
- Inside the table BigDataProfilingReport, create two groups 'CleansingReportGenericFields' and 'ProfilingFields'. The name of these two groups cannot be changed. All the fields inside these two groups are reused from the included data model 'ebx-addon-daqa-cleansing-report-metadata.xsd'.
- Publish this data model to any dataspace.
- Go to the **TIBCO EBX® Match and Merge Add-on** dataset in the Administration part, Cleansing report storage table, and declare the new created dataset as new storage.

8.4 Predefined cleansing procedures

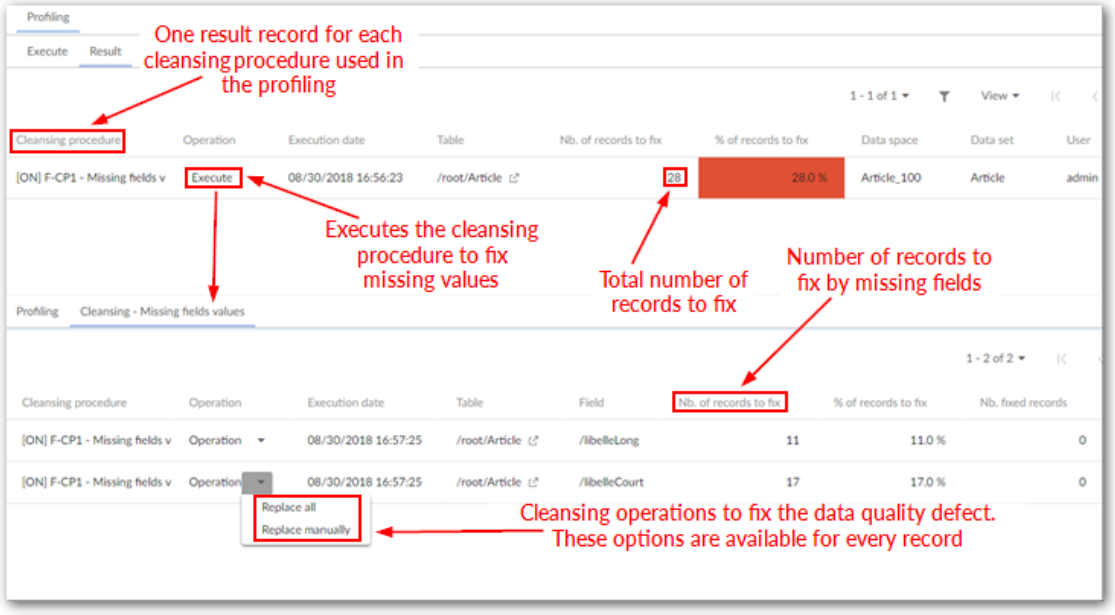
The ready-to-use cleansing procedures are listed in the following table. You also can create your own bespoke cleansing procedures using API.

Name	Description
Deprecated records	The 'Deprecated records' procedure allows you to get the list of the deprecated records (based on a time configuration) and decide to delete or keep the records. The way to detect a deprecated record is based on the following policies (time configuration): • Duration between the last modification date of the record and the current date (in days) • Comparison from the last modification date of the record and a selected date (Input parameters → To this date). • Comparison from the last modification date of the record and a computed date using a Java class (Input parameters → Java class).
Unused records	The 'Unused records' procedure provides you with a set of rules to detect the unused records and process them (delete, keep, etc.). An unused record is not referenced by any other record in the repository.
Foreign key fixing	The broken foreign keys are automatically identified and a UI allows you to fix the defective relationships.
Missing fields values	The 'Missing fields values' identifies all missing values for the configured fields. A missing value is either an empty value or any other string values that are configured depending on the need. To clean up the missing values, two cleansing operations are provided: 'Replace all' by a same value and 'Replace manually' to decide the value for every missing field

Table 43: Predefined cleansing procedures

8.5 **Profiling**

Clicking the 'Profiling' button executes cleansing procedures-in profiling mode-and generates a report containing table data quality defects. From this report, you can execute cleansing procedures to fix these defects. The following figure illustrates this process:



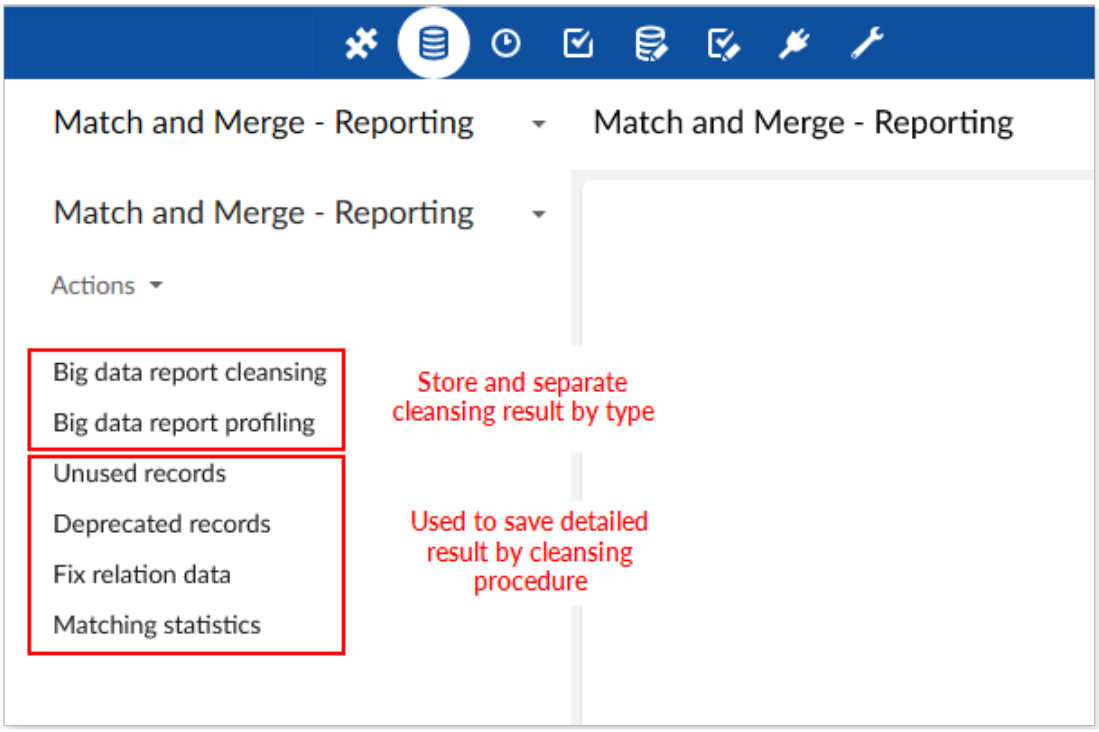
8.6 **Operations**

Clicking the 'Cleansing' button displays a list of the available cleansing procedures enabled for the current table.

Special notation:	
✔	In the current version of the add-on the following cleansing procedures are provided: 'Missing fields values', 'Foreign key fixing', 'Unused records', 'Deprecated records'. You can add your own cleansing procedures by using the API provided with the add-on.

8.7 **Result storage**

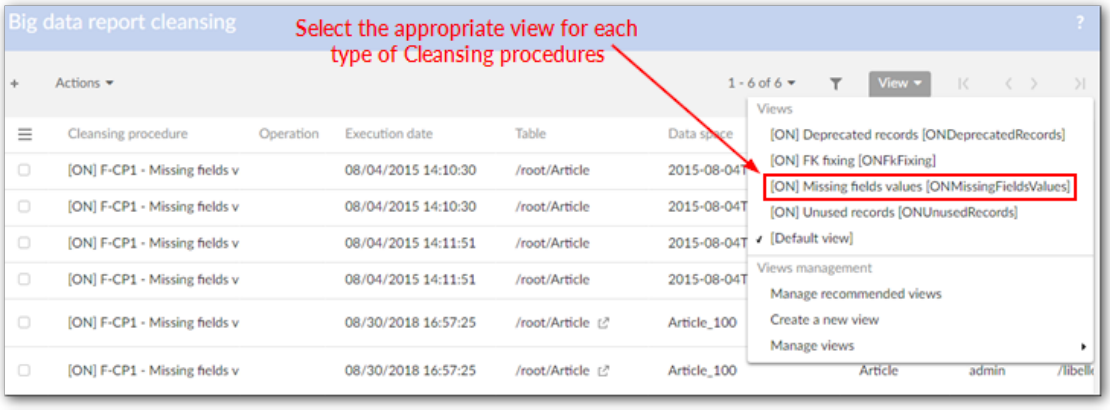
A dedicated dataspace is used to store and separate cleansing results by type (profiling and cleansing).



These tables are pre-pended as 'Big data' with flat and generic data structures. You can reuse them no matter what cleansing procedure output parameters are used.

Other tables are used to save detailed results by cleansing procedure, such as the list of unused records.

A data view can be configured to display data that is relevant to a specific cleansing procedure. An example of this using the 'Missing fields values' cleansing procedure is shown below.



Special notation:	
✔	You can set the 'Big data report cleansing' and 'Big data report profiling' tables to keep only the most recent results, or to store a history of results. This behavior is determined by the 'Save last value only' property contained in the 'Cleansing procedure' table.

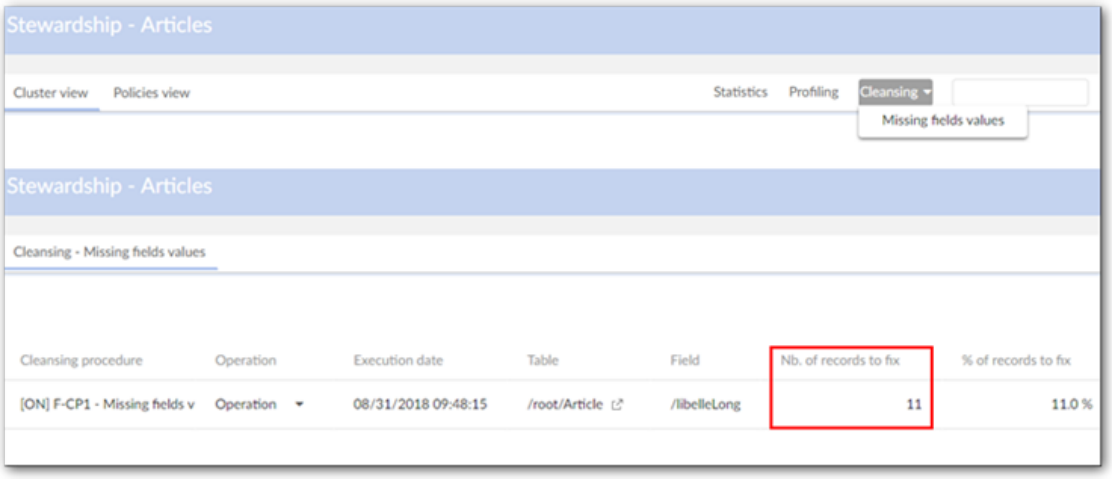
8.8 **Example**

In this section, you will find the examples of cleansing operations.

Cleansing - missing field values

In the following example, the pre-defined cleansing procedure '[ON] F-CP1 - Missing fields values' was configured to fix records with an empty value in the 'Long label' field.

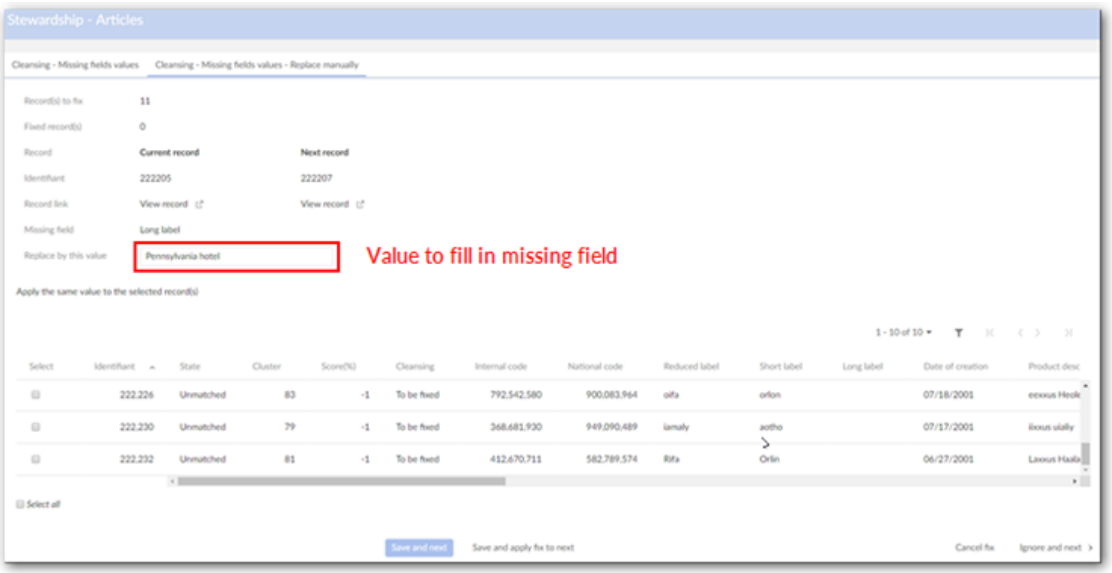
The 'Missing fields values' option displays in the matching view under the 'Cleansing' drop-down list and allows you to execute the cleansing procedure. Once you enter the cleansing screen, the add-on generates a report containing table data quality defects with a total number of records to fix.



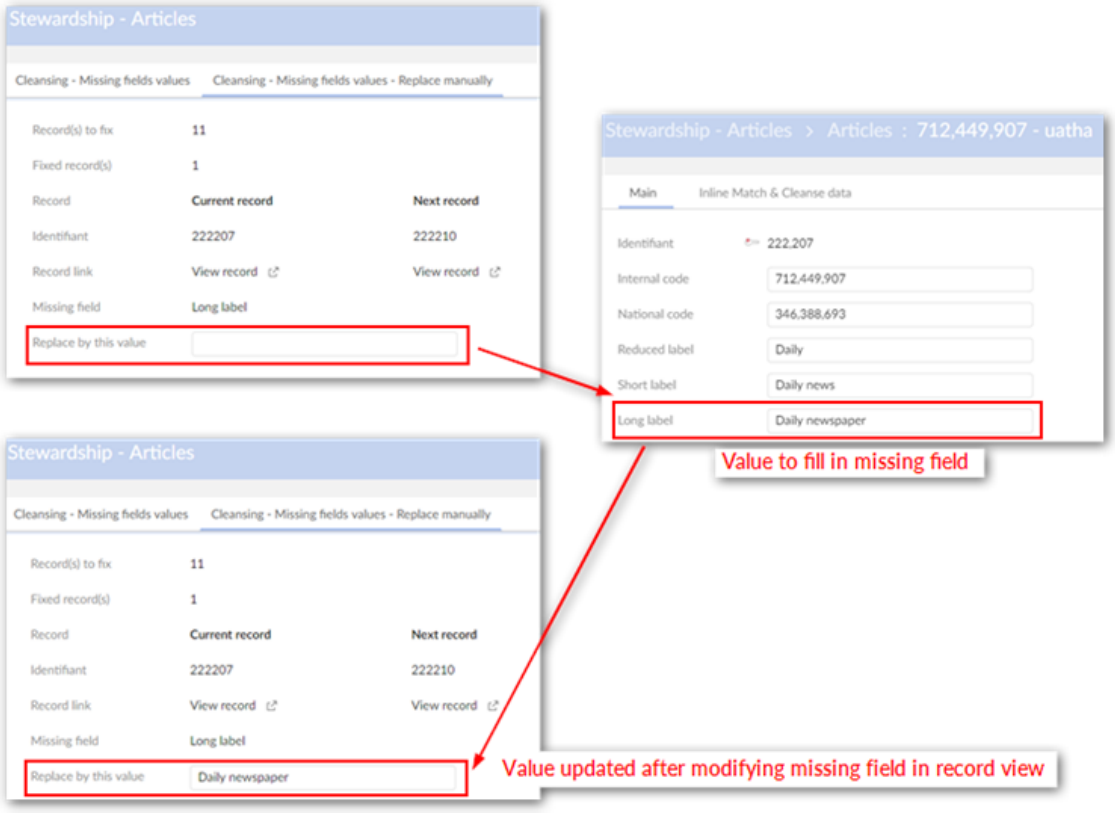
There are two ways to fix the records:

- **Replace all** - fill in all missing fields with the same value
- **Replace manually** - fill in missing fields with the same or different values

By executing the 'Replace manually' option, you can narrow your focus by selecting records in the set of records to fix. You can remove the filter and come back to the screen to work on all records using 'Deselect all'.



A preview button, available at the end of 'Record link' field, allows you to update the missing field directly in the record view. The 'Replace by this value' updates with the newly added value for the missing field.



Instead of entering a value to replace one at a time for each record, you can click on 'Save and apply same on next' button to open the next record with the 'Replace by this value' populated with the previous entry.

CHAPTER 9

Crosswalk (Records linking analysis)

This chapter contains the following topics:

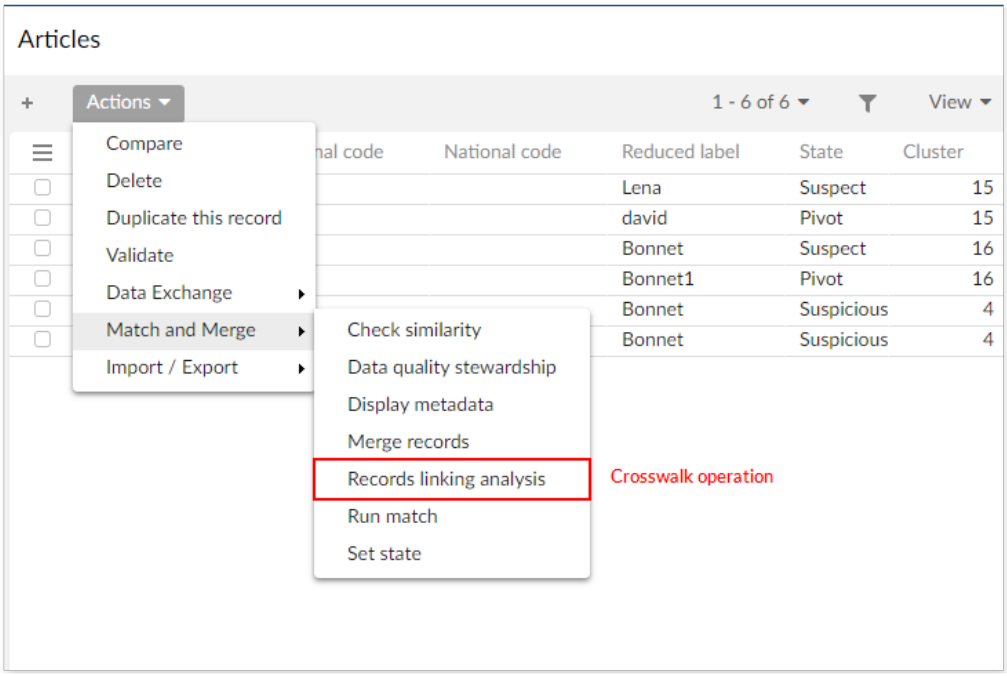
9.1 Introduction

A 'Crosswalk' operation allows you to create a cross-reference table based on a matching policy executed over multiple tables. For instance, the 'Client', 'Party' and 'Customer' tables have duplicate records and you want to create a cross-reference table that shows how the same record is represented in multiple tables. This feature is not used to merge the records but provides a convenient place to view the linked records. The 'Records linking analysis' service executes this function.

9.2 Enabling 'Records linking analysis'

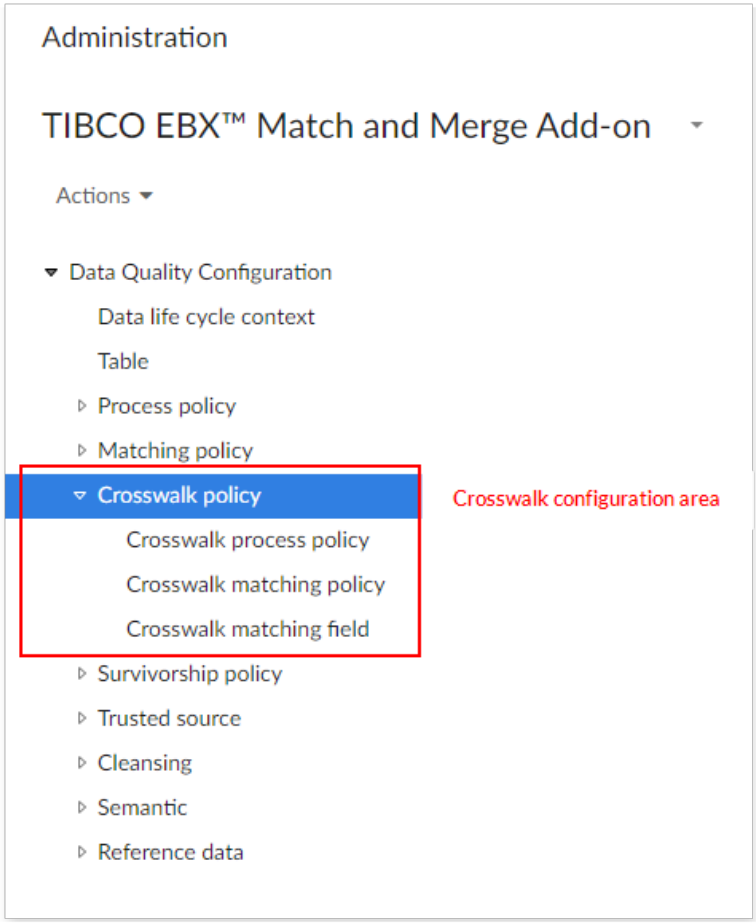
To enable the **Crosswalk** feature for a table, its **Use crosswalk** property needs to be set to **Yes**. This property is available in the **Table** domain located in the **TIBCO EBX® Match and Merge Add-on** dataspace under the **Administration** tab. You also need to configure a **Crosswalk process policy** and the related matching policies and matching fields. Once configuration is complete, you can execute the **Records linking analysis** service. This service is available from the **Actions** drop-down menu

when viewing a table that has the **Crosswalk** feature enabled—as highlighted in the image below. The service can be applied either on the whole table or on selection.



9.3 Configuration

The 'Crosswalk Policy' domain is used to configure the 'Records linking analysis' function applied to a table. It is available from the **TIBCO EBX® Match and Merge Add-on** dataspace located under the EBX® 'Administration' tab.



Records linking analysis workflow script task

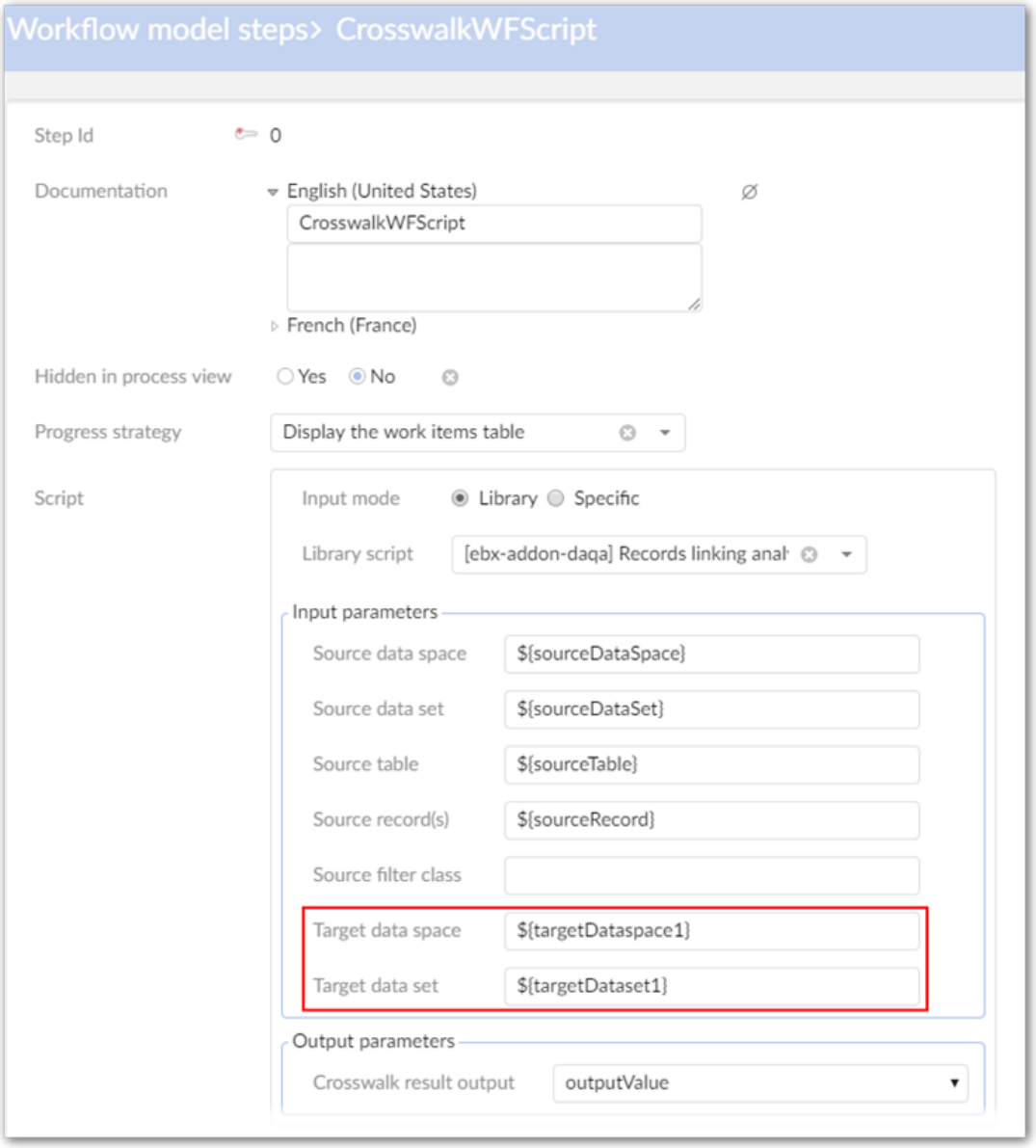
Without having to modify a Crosswalk matching field, you can use a workflow script task to declare a target dataspace for the 'Records linking analysis' service. There, you can define such parameters as: source dataspace, source dataset, source records, source filter class, target dataspace, target dataset, and an output parameter to store the results. The operation uses Tables and Fields that were earlier defined in the configuration. If the target dataspaces and datasets have not been registered with the add-on, the corresponding dataspaces and datasets in the configuration will be retrieved and used instead.

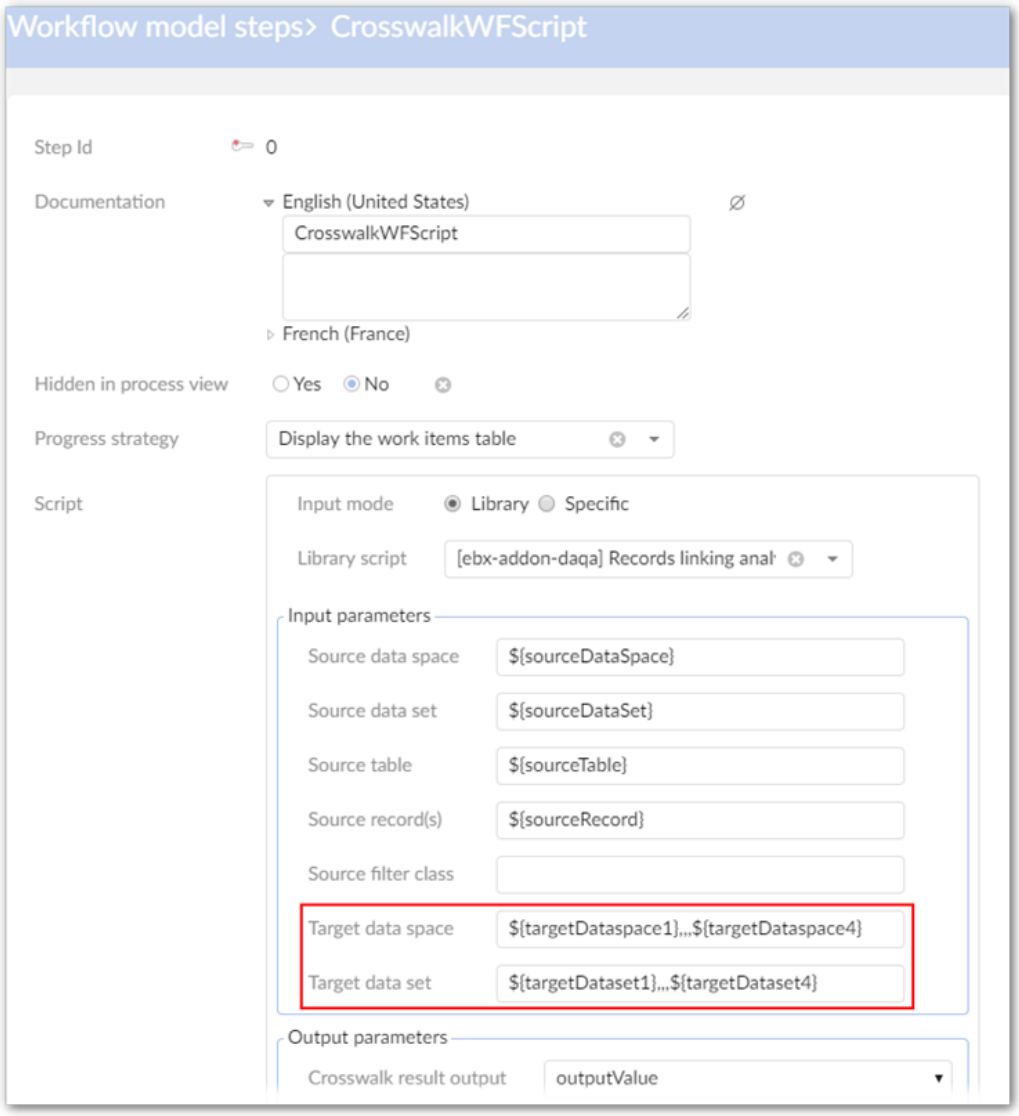
To specify a target dataspace using workflow script task, you must follow the following steps:

- Configure a Crosswalk matching field.
- Configure workflow script task **[ebx-addon-daqa] Records linking analysis**.
- Define data context.
- Then publish and launch workflow to finish.

The 'Target dataspace' field allows you to specify one, or enlist all the dataspaces that have been registered in the configuration. If you declare more than one dataspace, use a comma to separate them. Additionally, you must define the dataspaces in the same order as the configuration. For example: DataSpace1, DataSpace2, DataSpace3, etc. If you want to skip one, or more dataspaces, do not declare the dataspace in its intended position. The following example skips DataSpace2: DataSpace1, ,DataSpace3.

If you leave the 'Target dataspace' field undefined, the script uses the dataspace registered in the configuration. Please refer to the images below for examples.





If both the 'Source record(s)' and 'Source filter class' fields are not defined, the 'Records linking analysis service' runs on all records in the table. Note that the 'Source record(s)' has higher priority than 'Source filter class'.

Crosswalk process policy

A crosswalk process policy defines a set of parameters used to execute the 'Records linking analysis' service. Multiple crosswalk policies can be defined on a table, but only one can be active at a time.

Source (logical name: **CrosswalkProcessPolicy**).

Properties	Definition
Crosswalk process policy code	Any naming convention without white space can be used.
Description	The description of crosswalk process policy.
Source table	Table on which the crosswalk process policy is defined.
Active	If set to 'Yes': This crosswalk process policy is used. If set to 'No': This crosswalk process policy is not used. Default value: 'Yes'
Match in source table	Determines if the crosswalk policy can match records within the source table or not. In an example configuration, a 'Client' table is the source table and the 'Records linking analysis' service is configured to match against the 'Customer' table. In this instance, if the 'Match in source table' property is set to 'No', the returned results will show only records considered duplicated between the two tables. If you want to also check for duplicates within the 'Client' table itself, then the 'Match in source table' property must be set to 'Yes'.
Keep previous result	If set to 'Yes': The previous results of the 'Records linking analysis' service are kept in the result tables ('Crosswalk' and 'Crosswalk additional result' tables). If set to 'No': The previous results of the 'Records linking analysis' service are replaced by the new execution results (located in the 'Crosswalk' and 'Crosswalk additional result' tables). Default value: 'No'
Maximum number of results	The maximum number of results saved in the 'Crosswalk additional result' table. When no value is defined, all results are saved. Default value: '20'
Stewardship min scope (%)	When the matching score is lower than this threshold then the record is not considered a match.
Threshold second matching (%)	When the score computed by the first algorithm is between 0% and the value specified by the 'Threshold second matching(%)' property, then the second algorithm is applied to detect potential false negative records. For example, if 'Threshold second matching' is set to '5%' then the second algorithm executes each time the score of the first algorithm is lower than '5%'. Constraint: 'Threshold second matching' is lower than or equal to 'Stewardship min score'. This property is used when 'Second matching algorithm' in the crosswalk matching policy is not empty. Default value: '0'

Properties	Definition
2nd level stewardship min score (%)	When the second matching level is configured to find false negative records from a first level matching, then this threshold is used to evaluate the matching score and whether or not it provides a suspect record. This property is used when 'Second matching algorithm' in the crosswalk matching policy is not empty.

Table 44: Crosswalk process policy

Crosswalk matching policy

A crosswalk matching policy is used to configure how matching executes to detect duplicate records and then issues the records linking analysis.

Source (logical name: **CrosswalkMatchingPolicy**).

Properties	Definition
Crosswalk matching policy code	Any naming convention without white space can be used.
Description	The description of crosswalk matching policy.
Source table	Table on which the crosswalk matching policy is defined.
Active	If set to 'Yes': This crosswalk matching policy is used. If set to 'No': This crosswalk matching policy is not used. Default value: 'Yes'
Main matching algorithm	The default algorithm for all crosswalk matching fields that use this crosswalk matching policy. If required, the main matching algorithm can be changed at the field-level (see table 'Crosswalk matching field'). Selection of a main algorithm is mandatory.
Second matching algorithm	The algorithm used to manage records that could be false negative results after main algorithm execution. Used after the first matching procedure, this algorithm ensures that no matching records have been missed. You cannot override this matching algorithm at the field-level.
Threshold matching	These threshold matching values are not mandatory. They are used instead of the corresponding values that are defined at the crosswalk process policy level. If one field is provided, then the add-on will use all of them rather than the values specified at the crosswalk process policy level. In this case, the three values become mandatory.
Stewardship min score (%)	If undefined, the value configured at the Crosswalk process policy level is used. When the matching score is lower than this threshold, the add-on does not consider the record a match.
Threshold second matching (%)	If undefined, the value configured at the Crosswalk process policy level is used. When the score computed by the first algorithm falls between 0% and the value specified by the 'Threshold second matching(%)' property, then the second algorithm is applied to detect potential false negative records. For example, if 'Threshold second matching' is set to '5%' then the second algorithm is executed each time the score of the first algorithm is lower than '5%'. Constraint: 'Threshold second matching' must be lower than or equal to 'Stewardship min score'. This property is used when the 'Second matching algorithm' property in the crosswalk matching policy is not empty.
2nd level stewardship min score (%)	If undefined, the value configured at the Crosswalk process policy level is used.

Properties	Definition
	When the second level of matching is configured to seek false negative records from the first matching, then this threshold is used to evaluate if the matching score provides a matched record or not. This property is used when the 'Second matching algorithm' property in the Crosswalk matching policy is not empty.

Table 45: Crosswalk Matching Policy

Crosswalk matching field

Configuration of the fields used for the crosswalk matching execution time. A 'Crosswalk matching field' is attached to a 'Crosswalk matching policy'.

Source (logical name: **CrosswalkMatchingField**).

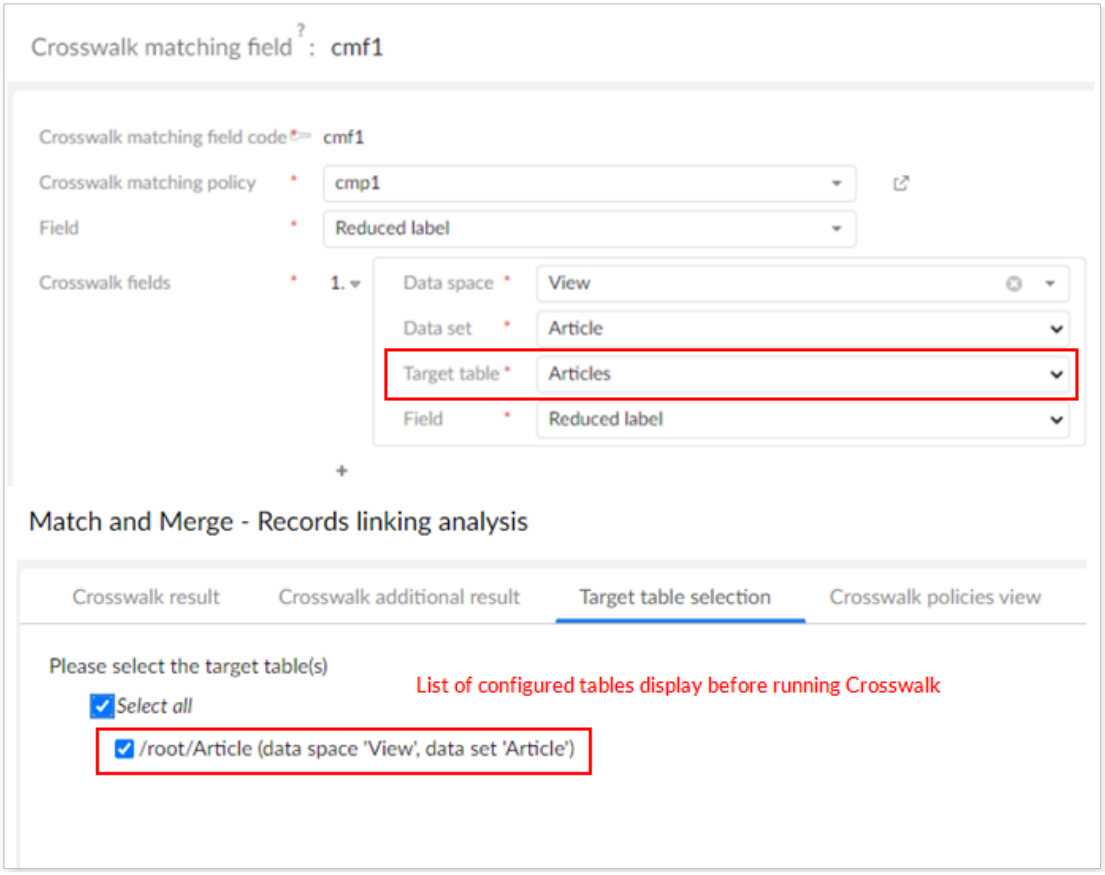
Properties	Definition
Crosswalk matching field code	Any naming convention without white space can be used.
Crosswalk matching policy	Reference to the 'Crosswalk matching policy' table for which the crosswalk matching field is defined.
Field	A field in the table that is declared by the referenced Crosswalk matching policy. This field is used during the matching procedure. If the field value is empty at execution time, then matching is not enforced. It is possible to match inside a list. Two lists of values will match when at least one value is exactly the same in the two lists. For instance (John, Carl, Paul) will match with (Frank, Theo, John).
Crosswalk fields	This part of the configuration allows you to declare the fields of the target tables that must be matched with the 'Field' declared in the previous property. The records linking analysis will provide the cross reference between the record of the main table and the tables of the crosswalk fields declared here. When using a table in more than one Crosswalk matching field, it must occupy the same location in each field. For example, a field from a Party table may be defined first in one Crosswalk matching field. If you use a different field from the Party table in another Crosswalk matching field it must also hold the first position.
dataspace	dataspace where the target table is located. After selecting your desired dataspace, you must click 'Save' to make options available in 'dataset', 'Target table' and 'Field' below.
dataset	Date set where the target table is located.
Table	The target table.
Field	The target field, also known as 'Crosswalk field'.
Matching algorithm	By default, the algorithm used is the one defined as the main algorithm in the 'Crosswalk matching policy' table. This default value can be changed to select another algorithm.
Score weight	When several fields are used for matching, the score weight property gives a weight to each field score that the add-on uses to compute a weighted average. Example: '0.5' means the score of the field is worth half (50%). If a matching field is empty then the weighted average does not take into account this field. Default value: '1'
Use synonym group	A group of synonyms can be selected. If a field value in the suspect record does not match the pivot record, then a new matching is achieved with the synonym values rather than the

Properties	Definition
	initial value in the suspect field. The first positive match is used to get the final score.
Check synonym in all groups	If set to 'true': When matching uses the synonym mechanism, all child synonym groups are used to look for a synonym. If set to 'false': When matching uses the synonym mechanism, the synonyms are looked up within each separate child synonym group.

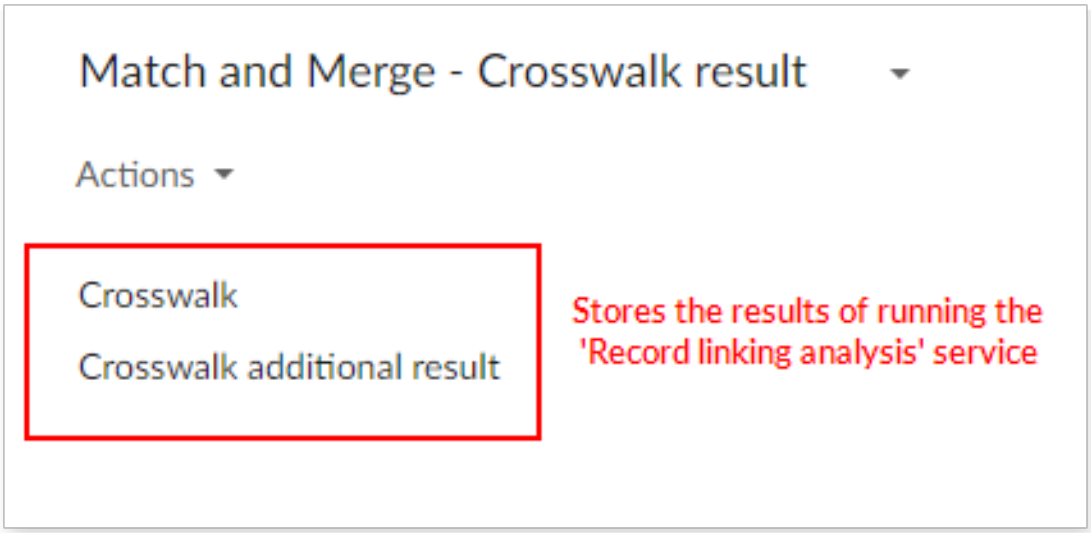
Table 46: Crosswalk Matching Field

9.4 **Target table selection**

To select target tables in the list of configured tables under 'Crosswalk fields' group. You can configure a set of crosswalk fields at once and then select the target table from 'Record linking analysis' screen before running crosswalk. The add-on matches the source table against the selected target tables.



9.5 **Result storage**



A dedicated dataspace **Match and Merge - Crosswalk result** is used to store the results of running the 'Records linking analysis' service.

The first twenty linked records are saved in a flat data structure in the 'Crosswalk' table that is easy to query. Beyond these results, an additional table is used to save one record per result: 'Crosswalk additional result'.

Crosswalk

This table is used to save the first twenty records linking results in a flat data structure that is easy to query.

Source (logical name: **Crosswalk)**.

Properties	Definition
Execution date	Date of the 'Records linking analysis' execution.
User	User in charge of the execution.
Crosswalk process policy	The crosswalk process policy's identifying 'Code' that has been used for these results.
Crosswalk matching policy	The crosswalk matching policy's identifying 'Code' that has been used for these results.
Source table	Link to the source table used for the 'Records linking analysis' service.
Source record	Link to the source record for which the 'Records linking analysis' service has been executed. The primary key value is displayed.
Record linking 01	
Table	Link to the target table having a cross-reference with the source table.
Record	Link to the target record having a cross-reference with the source record. The primary key value is displayed.
Score	Matching score of the target field against the source table field.

Table 47: Crosswalk

Crosswalk additional result

This table is used to save any additional results—beyond the initial twenty—from running the 'Records linking analysis' service. Every result is saved in one record.

Source (logical name: **CrosswalkAdditionalResult**).

Properties	Definition
dataspace	dataspace where the target table is located.
dataset	dataset where the target table is located.
Table	Table that has a cross-reference with the source table .
Record	Link to the target record having a cross-reference with the source record. The primary key value is displayed.
Score	Matching score of the target field against the source table field.
Source record	Foreign key to corresponding source record in 'Crosswalk' table.

Table 48: Crosswalk additional result

CHAPTER 10

Simulating a match using REST

This chapter contains the following topics:

10.1 Overview

The EBX® Match and Merge Add-on allows you to use a REST service to simulate a match operation. You can simulate a match on existing records, or before creating records to avoid duplicating data. Responses are returned in the JSON format shown below:

```
{
 rest_simulateMatch_results:[
  {
   "primaryKey": "XXXX",
   "recordLabel": "XXXX",
   "score": "XX.XX"
  },
  etc.
 ]
}
```

Simulating a match

The following table provides an example of asset URL retrieval:

Method type:	`POST`
URL pattern	`http://localhost:8080/ebx-addon-daqa/rest/v1/simulate-match/<dataspaceKey>/<datasetName>/<aTablePath>/<primarykey>?login=<user>&password=<password>`
Sample URL	`http://localhost:8080/ebx-addon-daqa/rest/v1/simulate-match/BReference/Article/_2E_2Froot_2FArticle/1?login=admin&password=admin` `http://localhost:8080/ebx-addon-daqa/rest/v1/simulate-match/BReference/Article/_2E_2Froot_2FArticle?login=admin&password=admin`
Request parameters	`dataspaceKey`: An encoded dataspace key of the user dataspace. `datasetName`: An encoded dataset name of the user dataset. `aTablePath`: An encoded table path (in the schema). `aRecord`: An encoded primary key.
Sample body	`{"content": { "name": "phap", "midlename": "quang", "generic": {"ratio": 2.5}, "offices": ["Ha Noi", "Sai Gon"] } }`
Response parameters	`primaryKey`: The primary key for a record returned as a possible match. `recordLabel`: The record's label. `score`: The record's matching score.
Sample response	`{ "rest_simulateMatch_results": [ { "primaryKey": "3", "recordLabel": " - yellow", "score": 100 }, { "primaryKey": "2", "recordLabel": " - green", "score": 100 } ] }`
Note	If you provide a record as input, the matching operation uses its value and not the request body (if one is defined). When inputting a date/time value, use the following pattern: yyyy-MM-dd'T'HH':mm:ss When inputting a number value, use the following pattern: XX.xx

CHAPTER 11

Migrating, backing up, and restoring settings

This chapter contains the following topics:

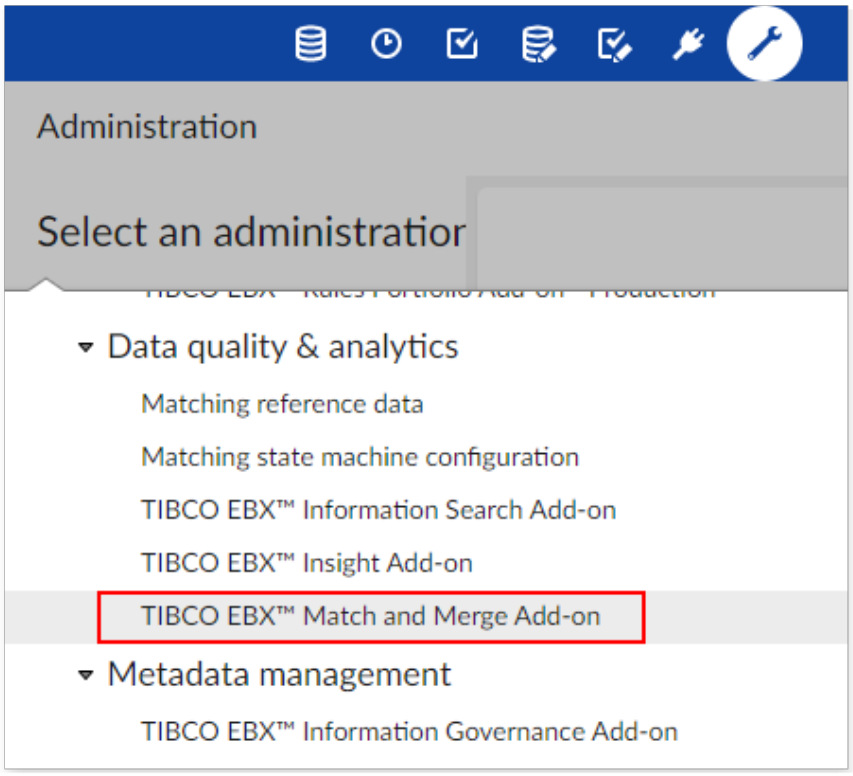
11.1 Overview

You can export an EBX® archive file to back up configuration settings. By importing this file, you can restore configuration settings or migrate them to another environment. Note that when you import, you can use the add-on's **Import configuration** service or the EBX® archive import service. The section below on importing describes the differences between these methods.

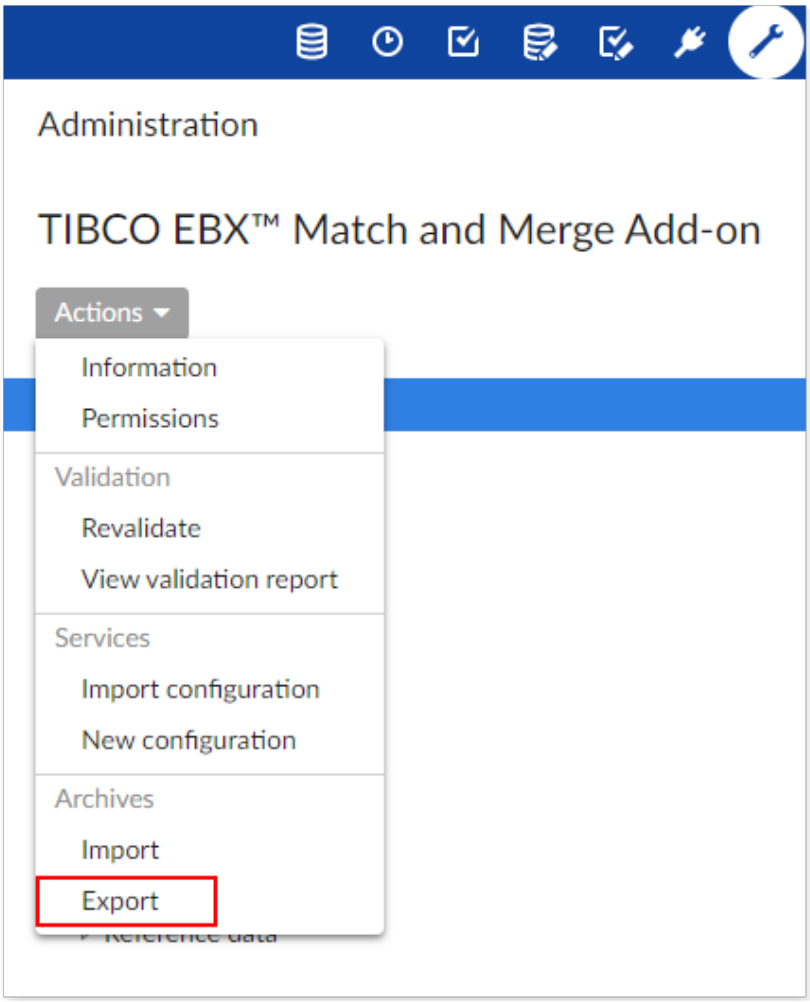
11.2 Backing up configuration settings

To export add-on configuration settings:

1. Open the **Administration** panel and select **TIBCO EBX® Match and Merge Add-on**.

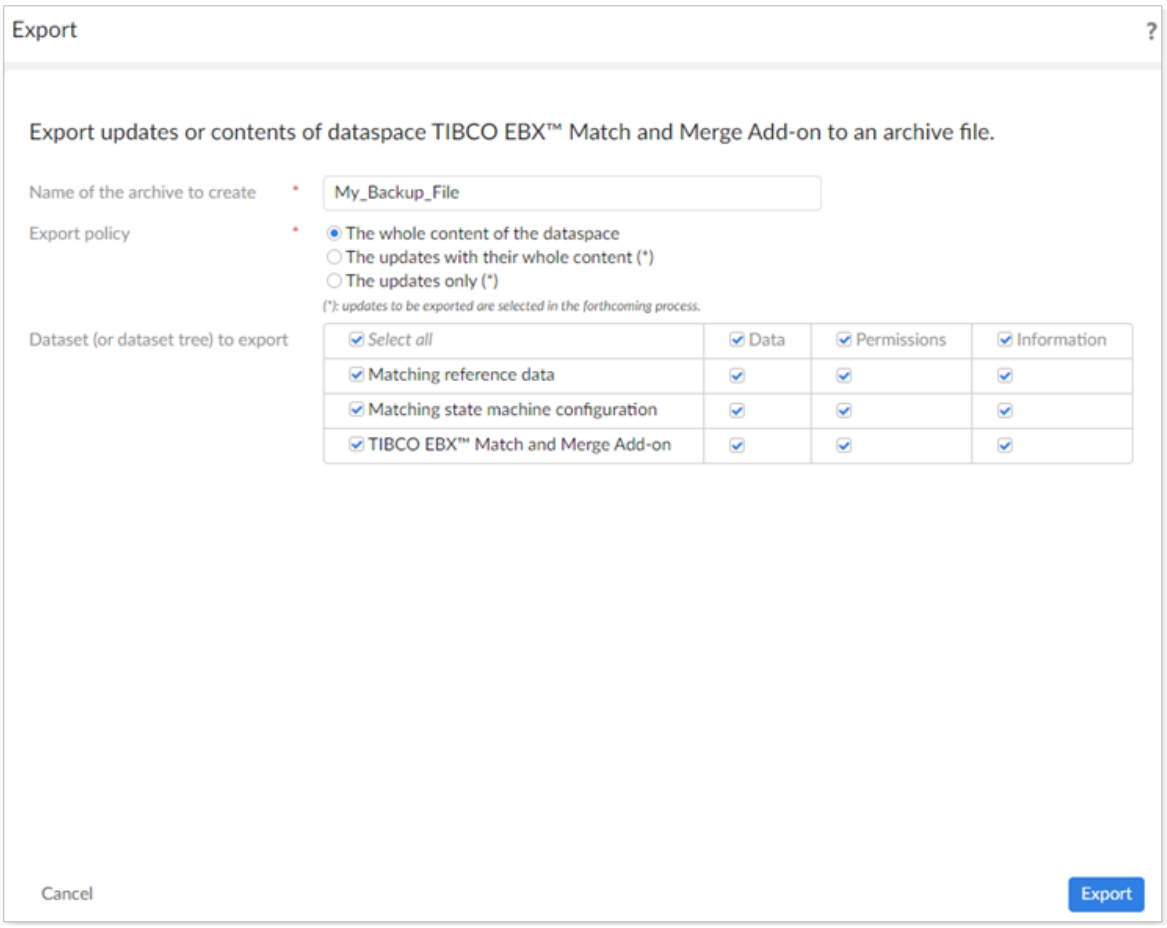


2. From the **Actions** menu, select **Export archive**.



3. Enter a name for the archive. In this example we keep the default options. If you would like to learn more about the archive export service, select the help icon. Alternatively, you can view the

EBX® product documentation's description of dataspace archives. See *User Guide* > *Dataspaces* > *Working with existing dataspaces* > *Dataspace archives*.



> **Note**
>
> After successful completion of the export, you can find the archive file in the `/ebx_home/ebxRepository/archives` directory.

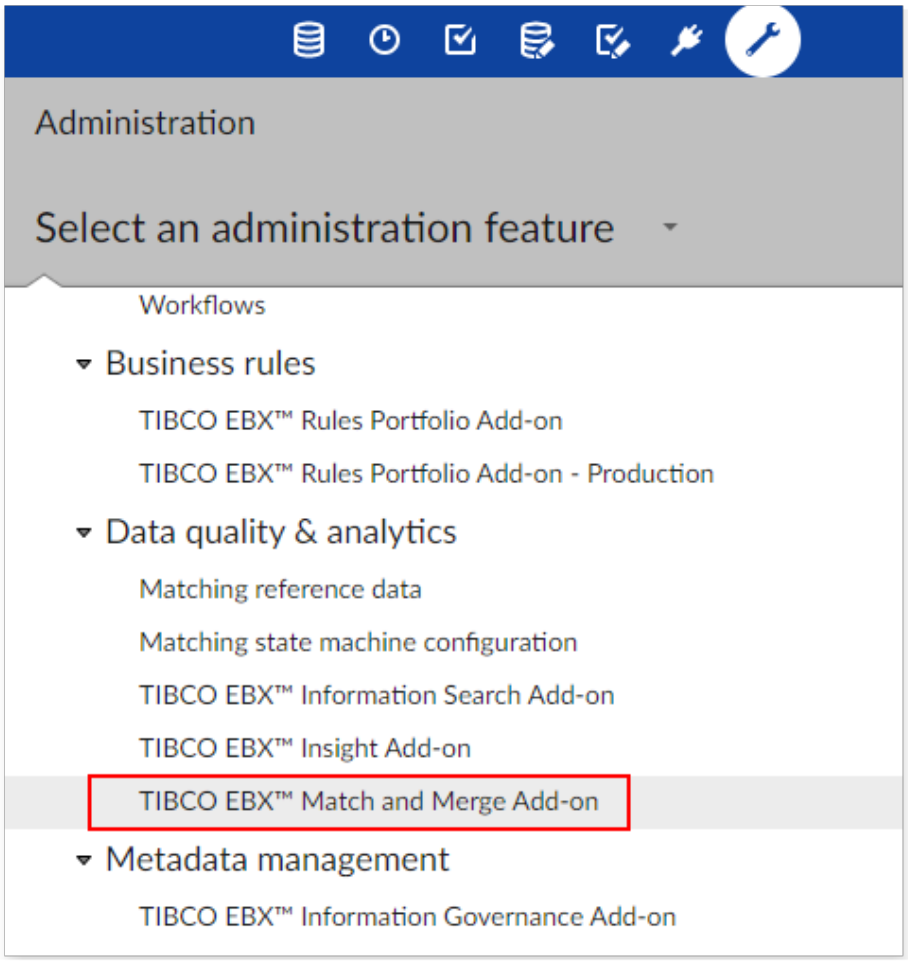
11.3 Restoring and migrating settings

When you import an archive to update configuration settings using the EBX® **Import** service, all of the latest cluster IDs will be overwritten. To prevent this behavior and keep the cluster IDs, use the add-on's **Import configuration** service.

To import configuration settings:

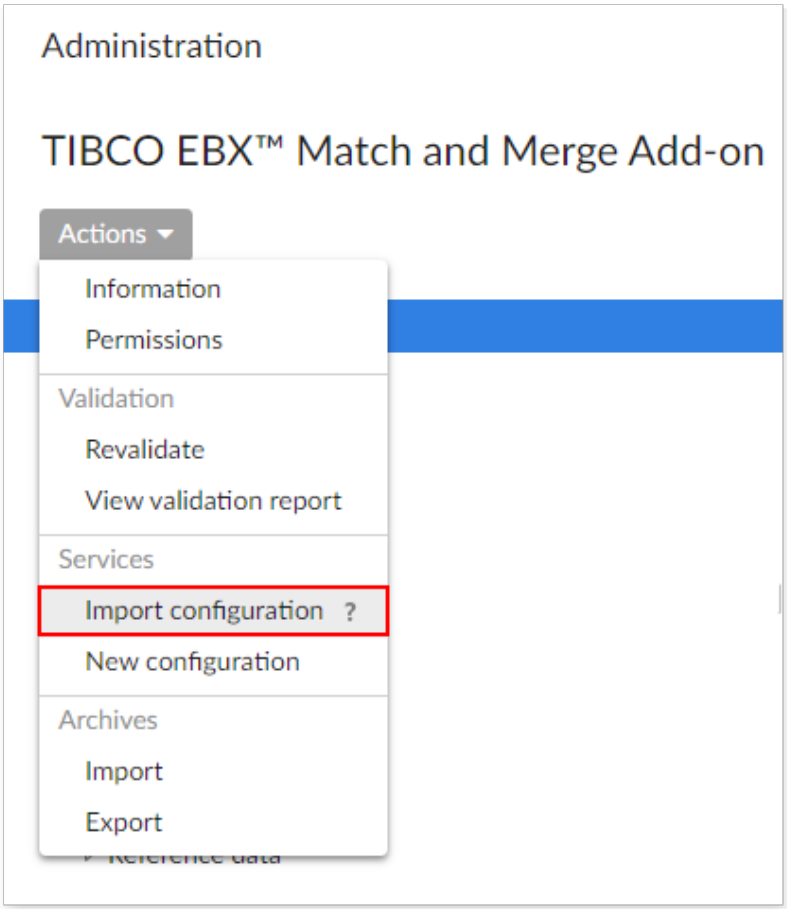
1. Ensure the exported archive file is in the correct location. If you are importing:
   - into the same environment you exported from, no action is required.
   - into a new environment, copy the archive file to `/ebx_home/ebxRepository/archives`.

2. Open the **Administration** panel and select **TIBCO EBX® Match and Merge Add-on**.



3. From the main **Actions** menu, select **Import configuration**.

> **Attention**
> You can also import an archive using the **Actions** menu's **Import** service. Please be aware that this will overwrite all of the latests cluster IDs.



4. You can find all configuration archive files available to import display in the page's **Archive to import** menu. If the desired archive does not display here, please make sure the archive file is in the correct location described in the first step. After choosing the desired archive, select **Import** to complete the process.

Reference Manual

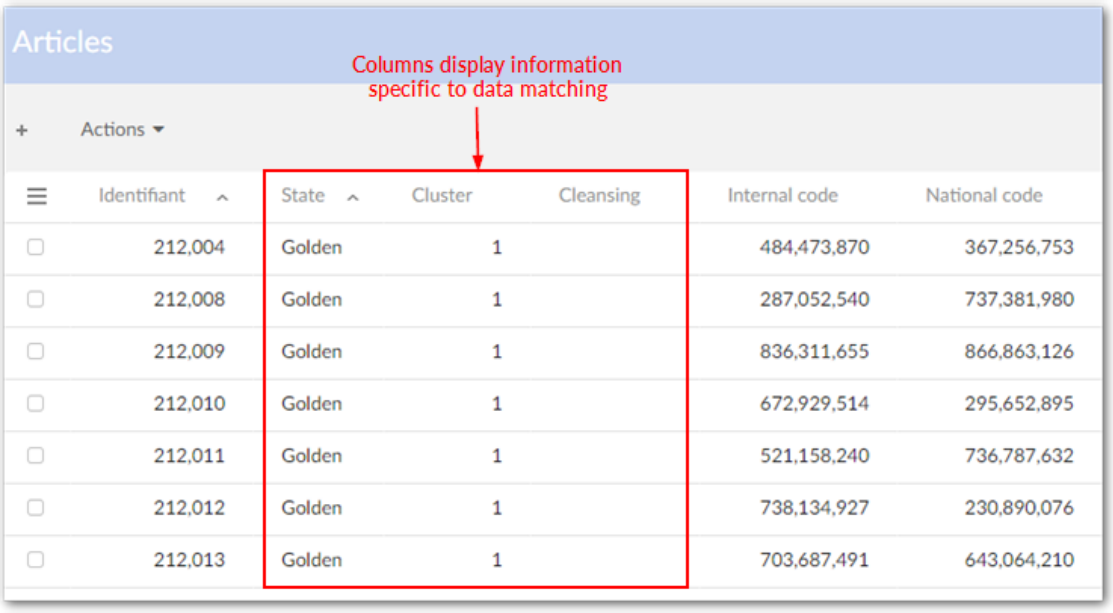
CHAPTER 12

User interface overview

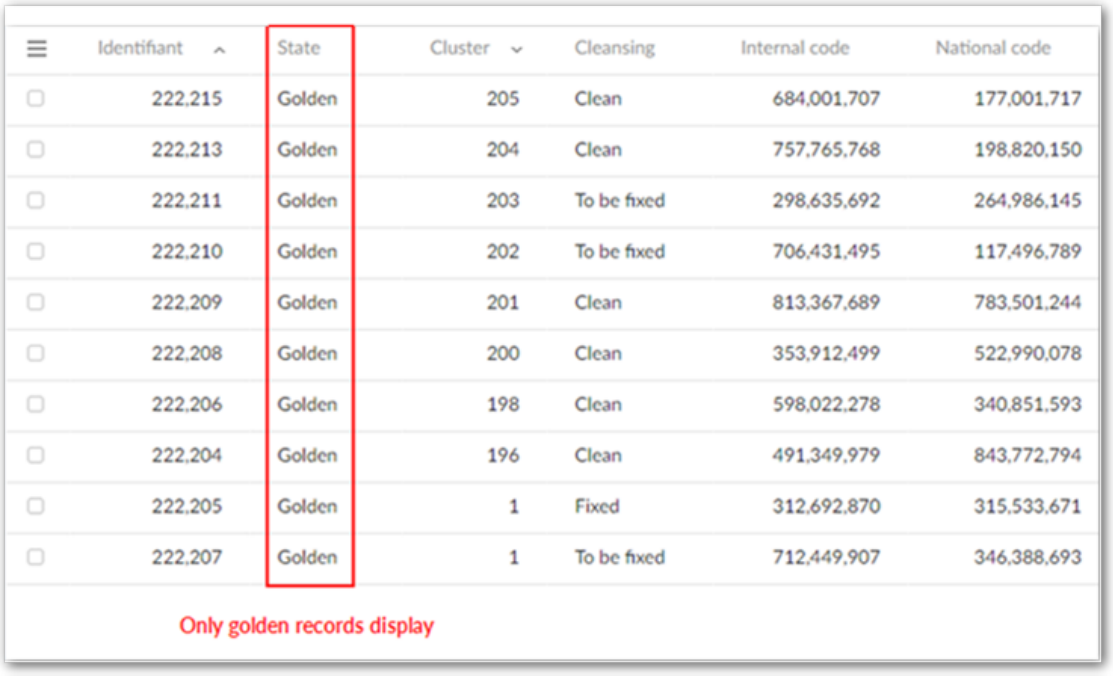
This chapter contains the following topics:

12.1 EBX® view

The default EBX® view includes three columns that display information specific to the add-on for every table record under add-on control. These three columns are **State**, **Cluster** and **Cleansing**.



By filtering the view, it is possible to hide data and/or display specific record states, such as showing golden records only.

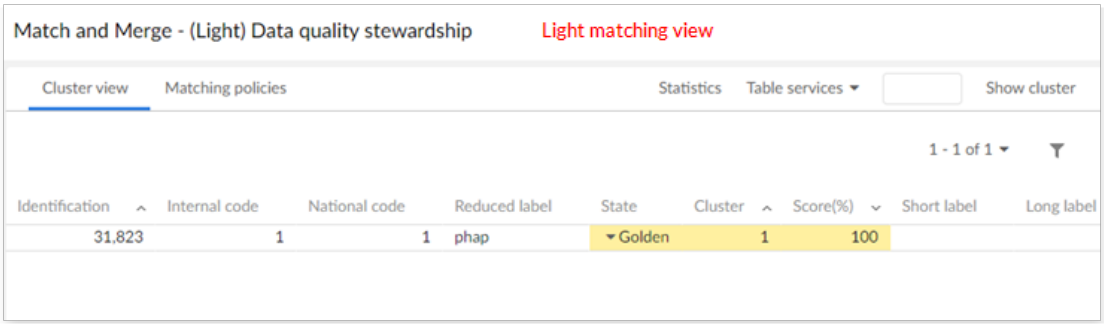


Accessing data quality views

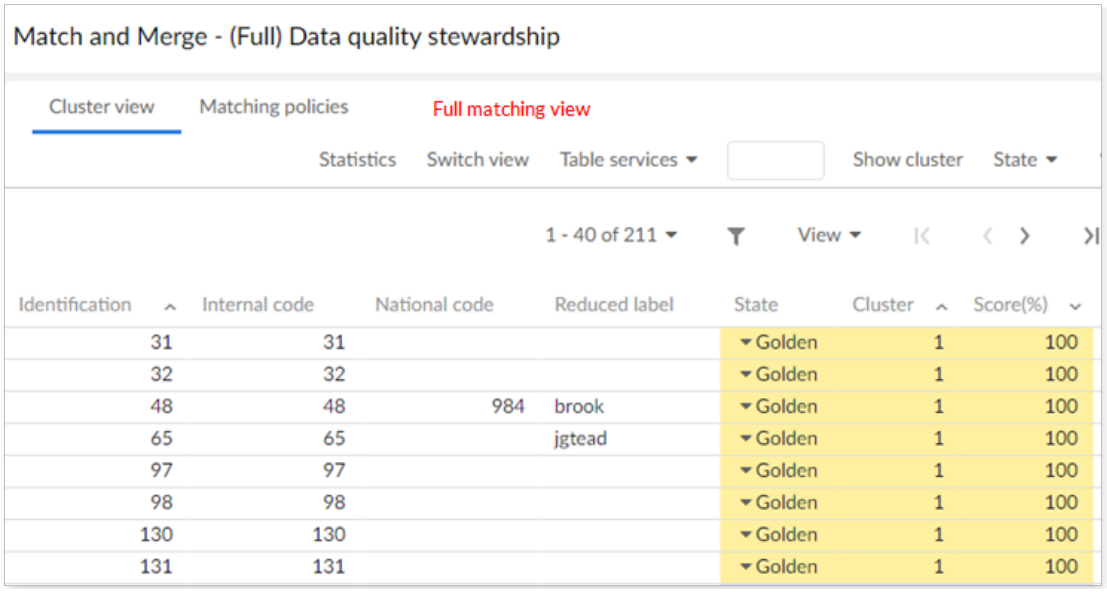
You use the **Data quality stewardship** service to access the light and full matching views. The view that displays depends on whether you have one, or no records selected. With a single active selection, the add-on displays **(Light) Data quality stewardship** view. When you do not have an active selection, the **(Full) Data quality stewardship** view displays.

To access the stewardship views, open the table's **Actions** menu and select *Match and Merge* > *Data quality stewardship*. The following images provide examples of both views:

The light view is shown below:



The full view is shown below:

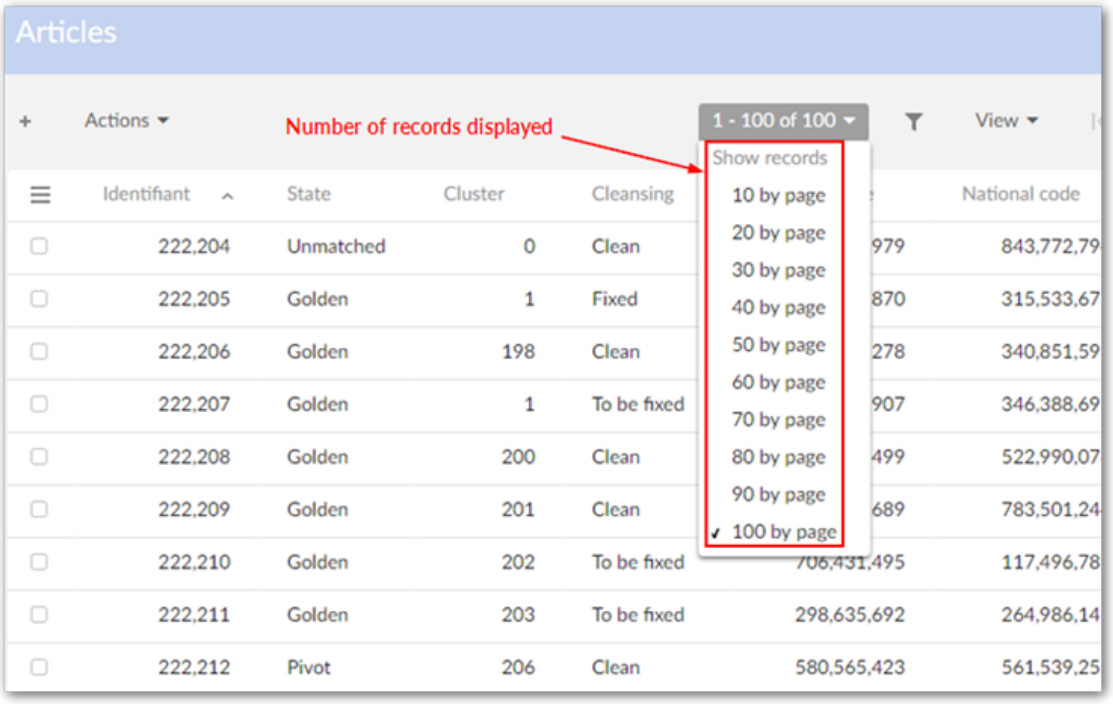


12.2 **Usability improvement**

To enhance the add-on's response time, you can use one of the following options:

- Choose the number of records displayed
- Switch to light view instead of full view

Number of records displayed

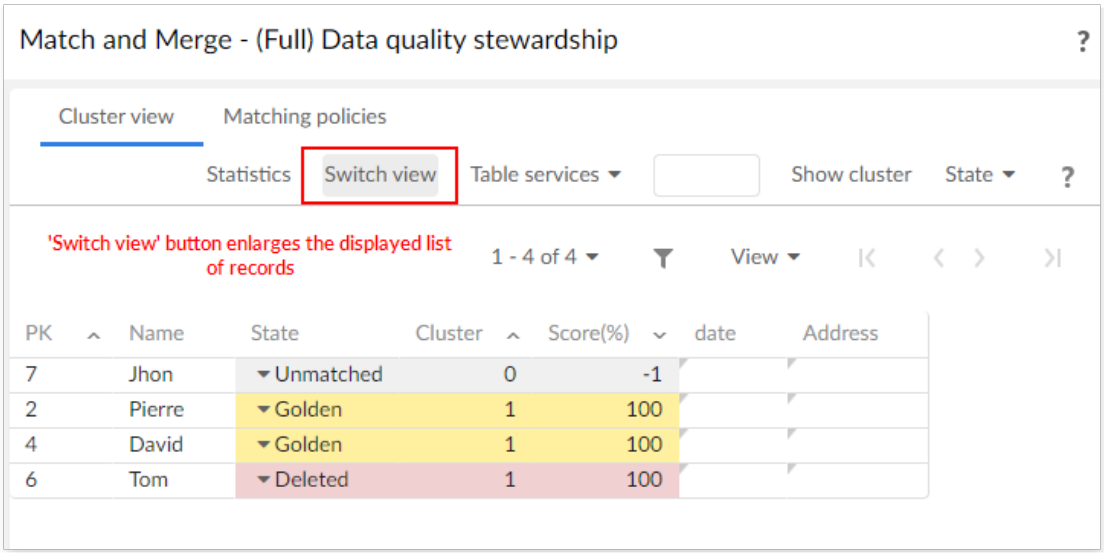


In matching views, the number of records shown in the lists can be adapted to improve response time. If the execution platform and/or network are limited, decreasing the number of records (In case of

network issue, it is also recommended to use a zip protocol to convey EBX® pages—HTML, scripts, etc.) shown can improve response time.

Switch to the light view

When you are in the full-view, you can use the **Switch view** button at any time to enlarge the displayed list of records.



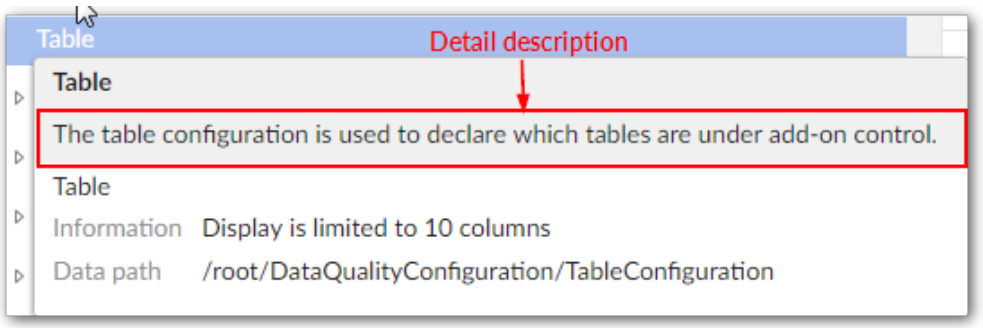
Query optimization

Special notation:	
✔	To guarantee better response time when displaying list of records that are sorted by clusters and scores, it is important to declare the right indexes on the table under add-on control. Please refer to the *Installation and first configuration* section for more detail.

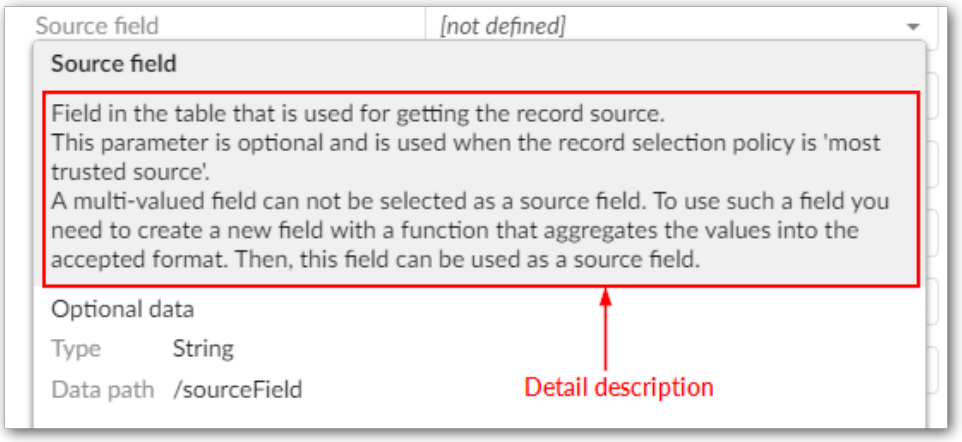
12.3 Online help

You can access context sensitive online help by clicking the **?** icon next to UI components (as shown below), or by clicking the question mark icon in the top-right corner of the page.

Table

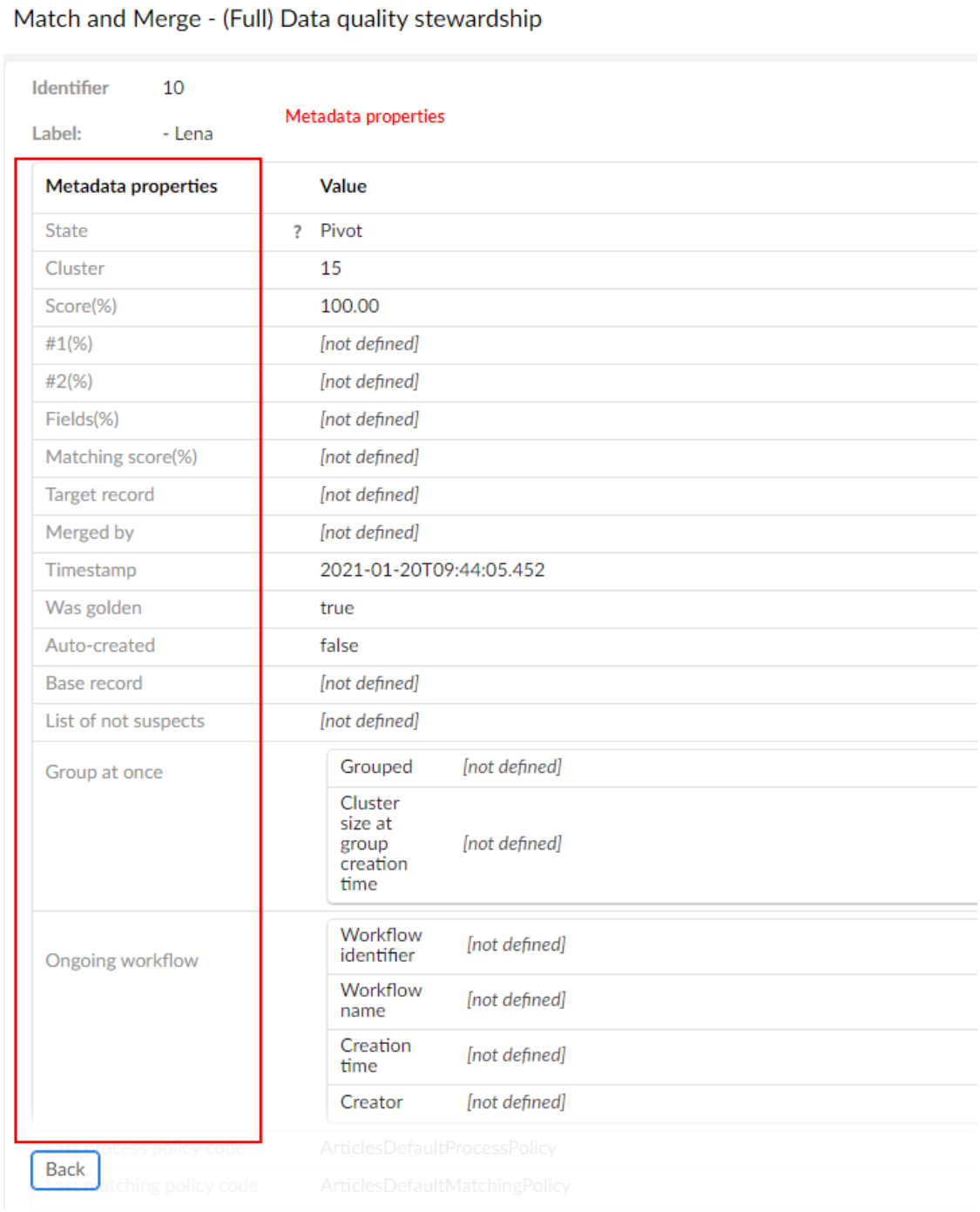


Field

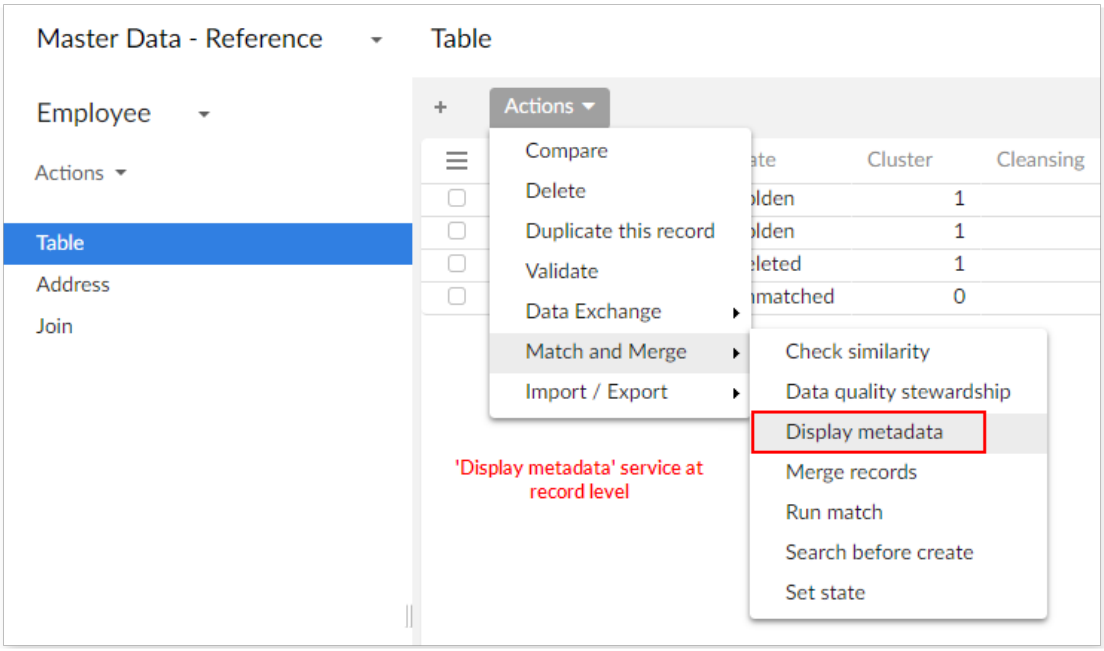

Source field
[not defined]
Source field
Field in the table that is used for getting the record source.
This parameter is optional and is used when the record selection policy is 'most trusted source'.
A multi-valued field can not be selected as a source field. To use such a field you need to create a new field with a function that aggregates the values into the accepted format. Then, this field can be used as a source field.
Optional data
Type String
Data path /sourceField
Detail description

Matching metadata

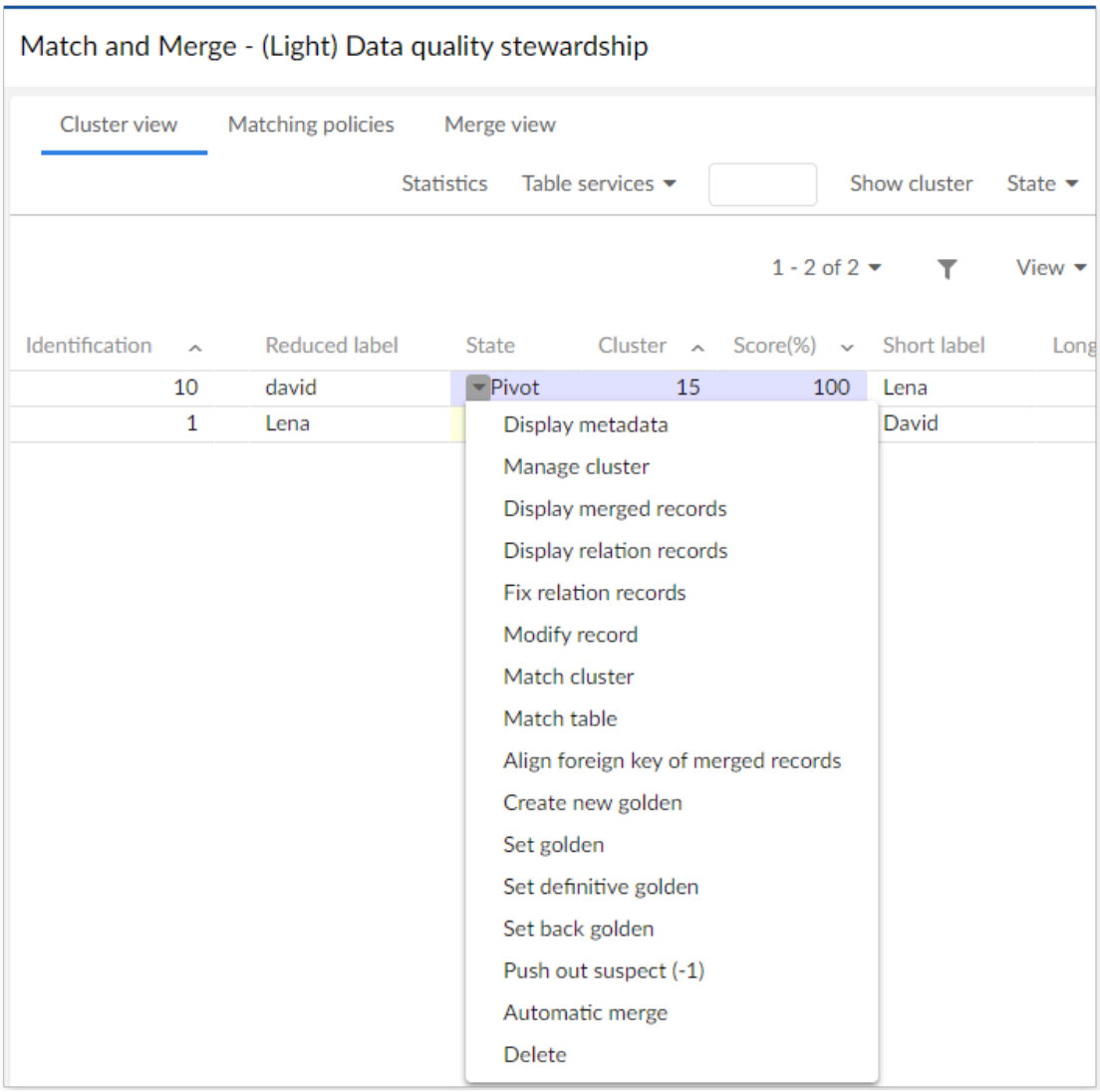
Invoking the **Display metadata** operation (through the stewardship light, full-view, and through the display metadata in tabular view) on a record opens a window that shows the record metadata values (tool tips are available to get help). This is illustrated below.



As an alternative to using the cluster view, you can display a table's metadata using an add-on service available in the tabular view. This service is available when a table has TIBCO EBX® Match and Merge Add-on metadata and displays when you select *Actions → Match and Merge*.



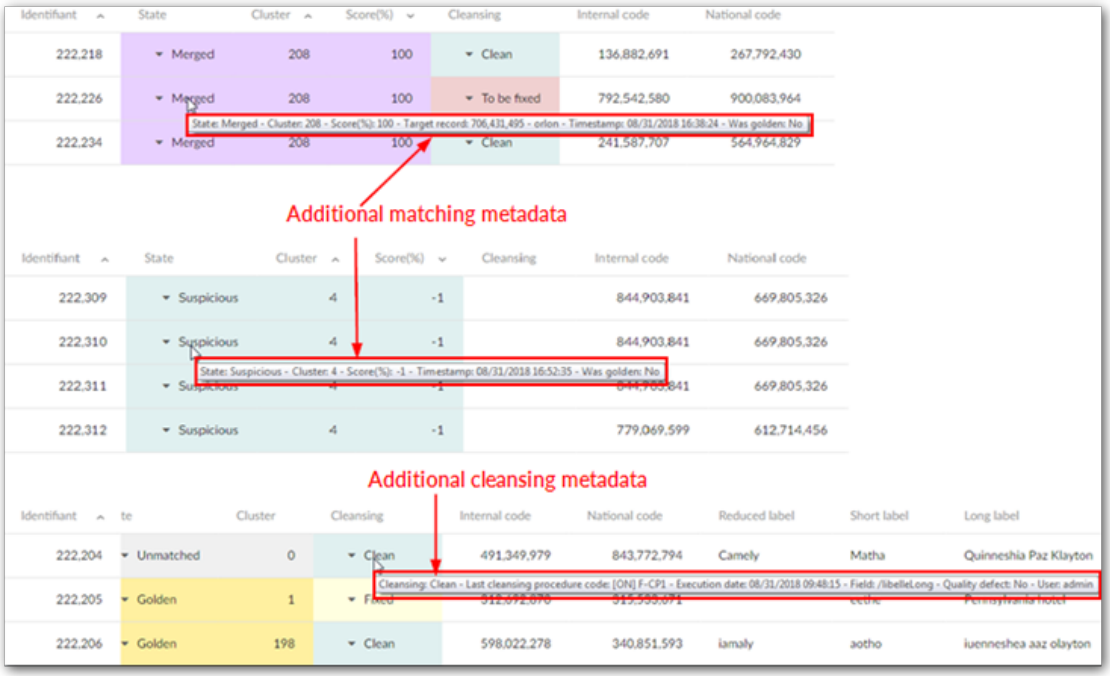
The **Display metadata** service presents full matching and cleansing metadata. This is exactly what would be shown when accessing this service from **Cluster** view.



When using the **Display metadata** service accessed when viewing a table in the standard tabular view, the Matching metadata and Cleansing metadata are shown in distinct tabs. They are displayed in separate windows in the cluster view.

Additional metadata on a record

In the EBX® Match and Merge Add-on views, you can display additional matching metadata for a record by setting the focus on the **State** and **Cleansing** fields. Available matching data is highlighted in *Matching Metadata* section.

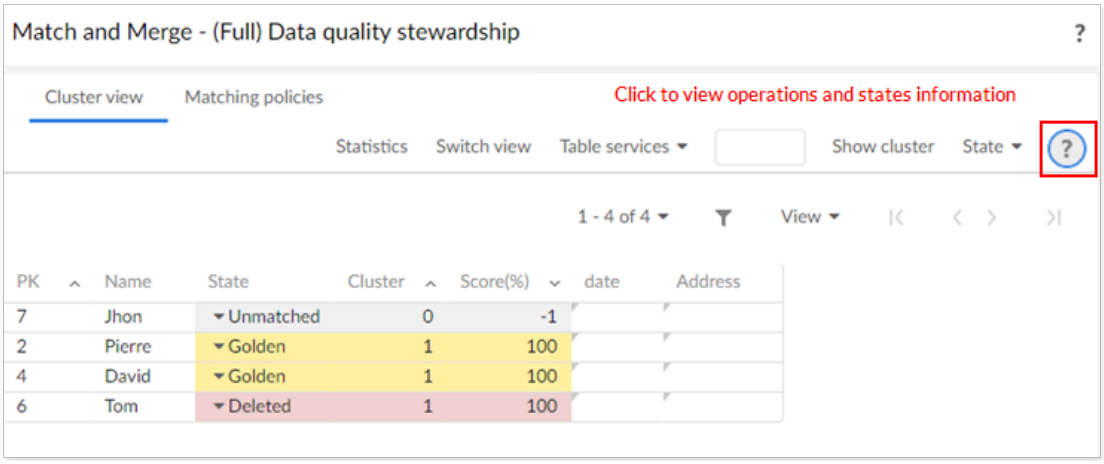


The **Target record** references the record that benefited from the merge.

When a record is under workflow control, the following additional data is displayed: workflow name, creation time and creator.

Operations and states

For a description of operations available on each state, you can click the question mark button at the top-right corner of **Cluster view** to access **Matching Operations applied to a single record** in the user guide.



Operations applied to a single record

List of operations	Type and context of use
Match cluster	Matching operations applied to a single record are not available if the process policy is incorrectly configured or the 'On DQ Process' property is set to 'No'.
Match table	EBX™ standard services remain available from the EBX™ view. It is possible to duplicate, compare and delete a record. Because the deletion is physical, it can entail integrity constraint failures. For example, if a merged record holds a foreign key to the record that has been deleted, then EBX™ will raise integrity errors. You can make the EBX™ delete service inaccessible based on user permission levels. This helps to ensure that a logical deletion with the add-on is always used. Note that if a suspect record is manually removed from a cluster using one of these operations and the cluster contains only pivot and merged records, the pivot becomes golden.
Match suspicious	
Create new golden	
Set golden	

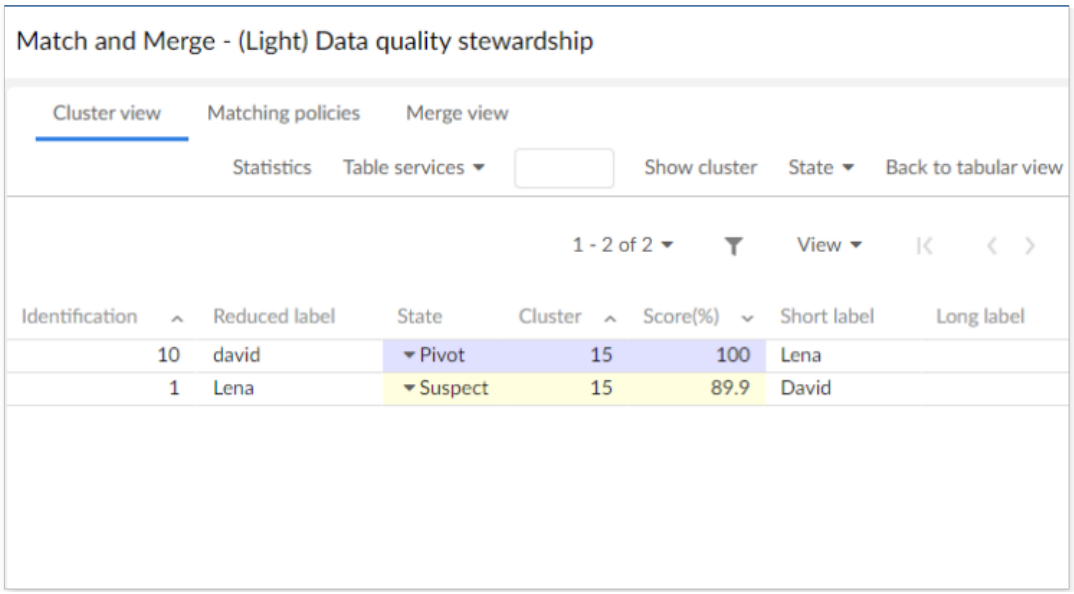
12.4 **Light matching view**

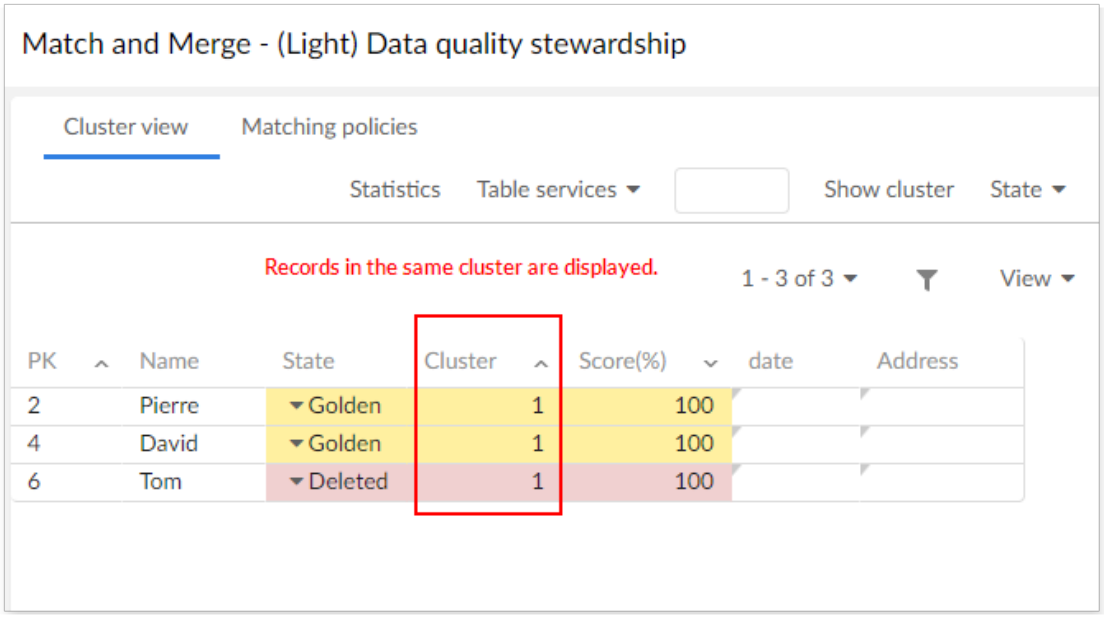
The light matching view displays records contained in a cluster. Every record offers a pop-up menu with available matching operations.

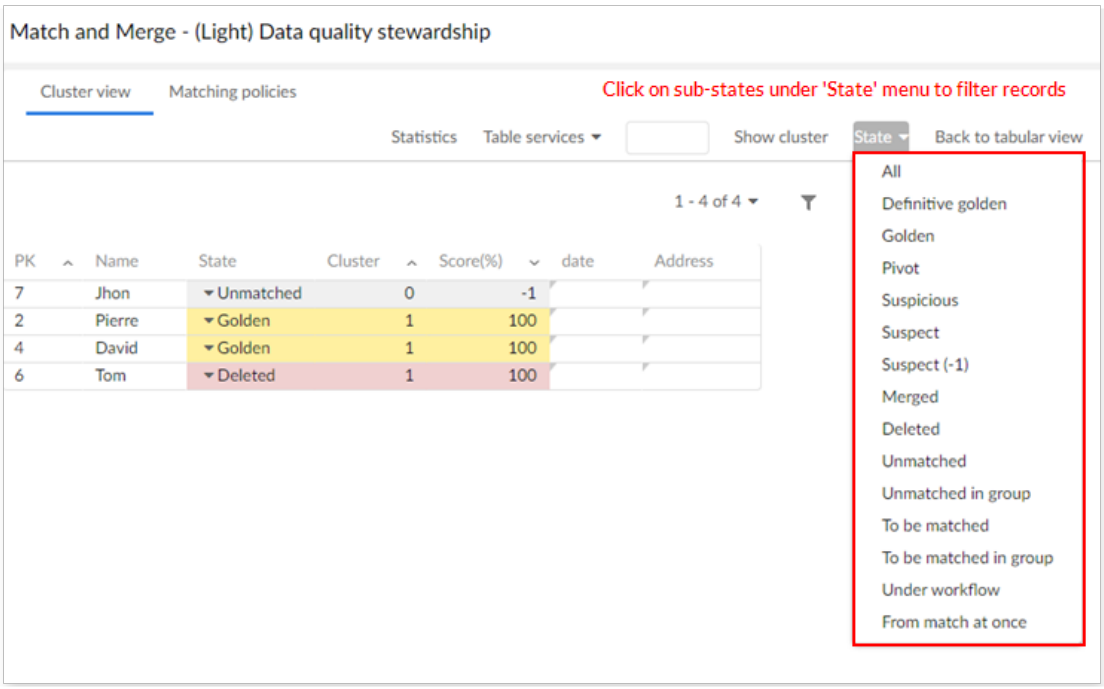
The record's matching score displays alongside the state and the cluster identifier.

The **Table services** menu button offers matching operations that can apply to the table (all records). The **Search** menu button allows you to search records based on matching policies. This button is not

displayed when there are no matching policies defined and if a matching field is not a type of string/ text.





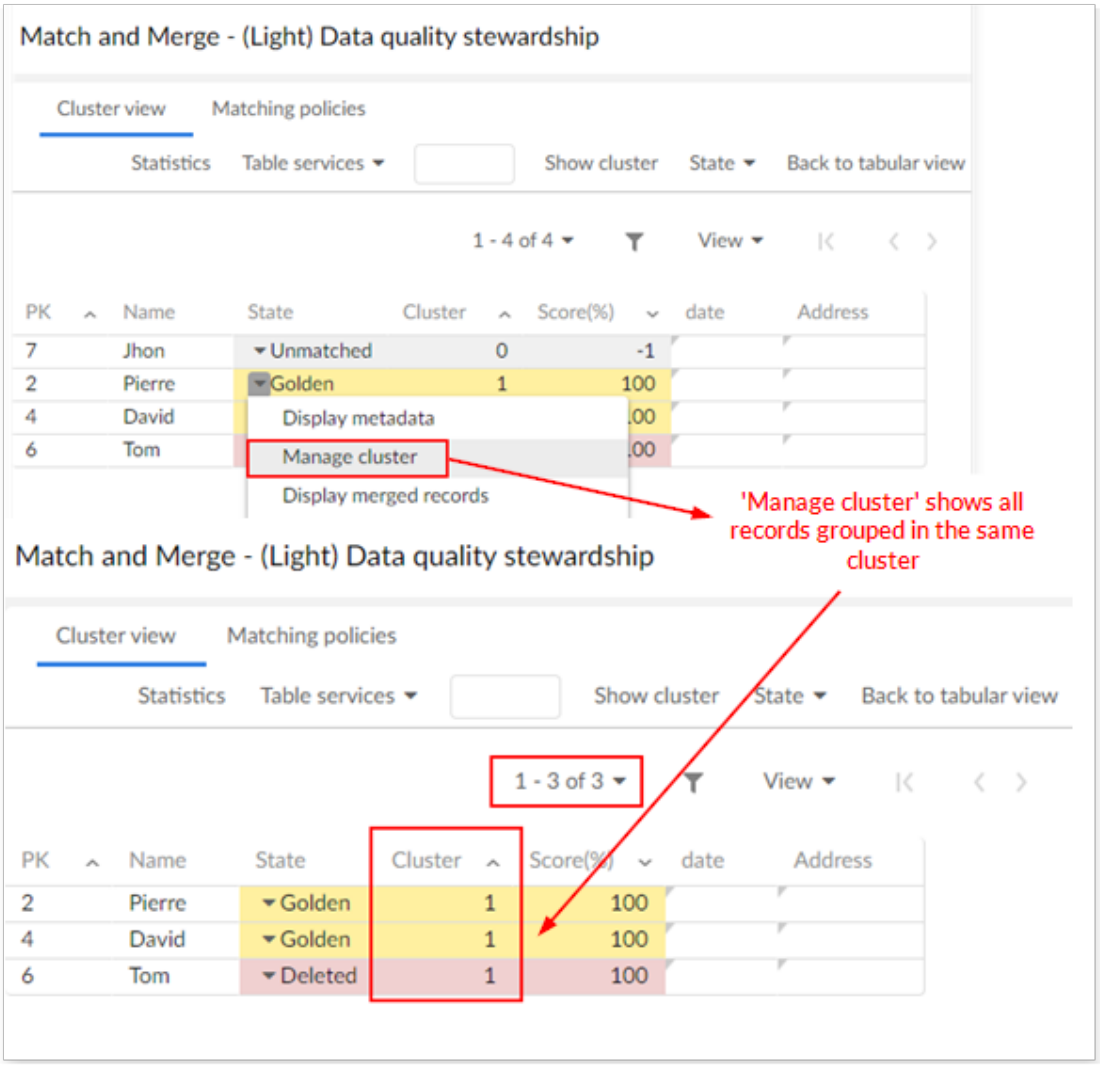


You can filter records by their current matching state using the **State** menu button. This button is also used to get records related to sub-states, such as **Under workflow** and **From match at once**.

The **Under workflow** filter gives all records that are under the control of a workflow.

The **From match at once** filter gives all records where the latest modification has been done with a **match at once** operation (an operation applied to a set of records).

From the list of records, using the **Manage cluster** operation opens all the records that are grouped in the cluster.

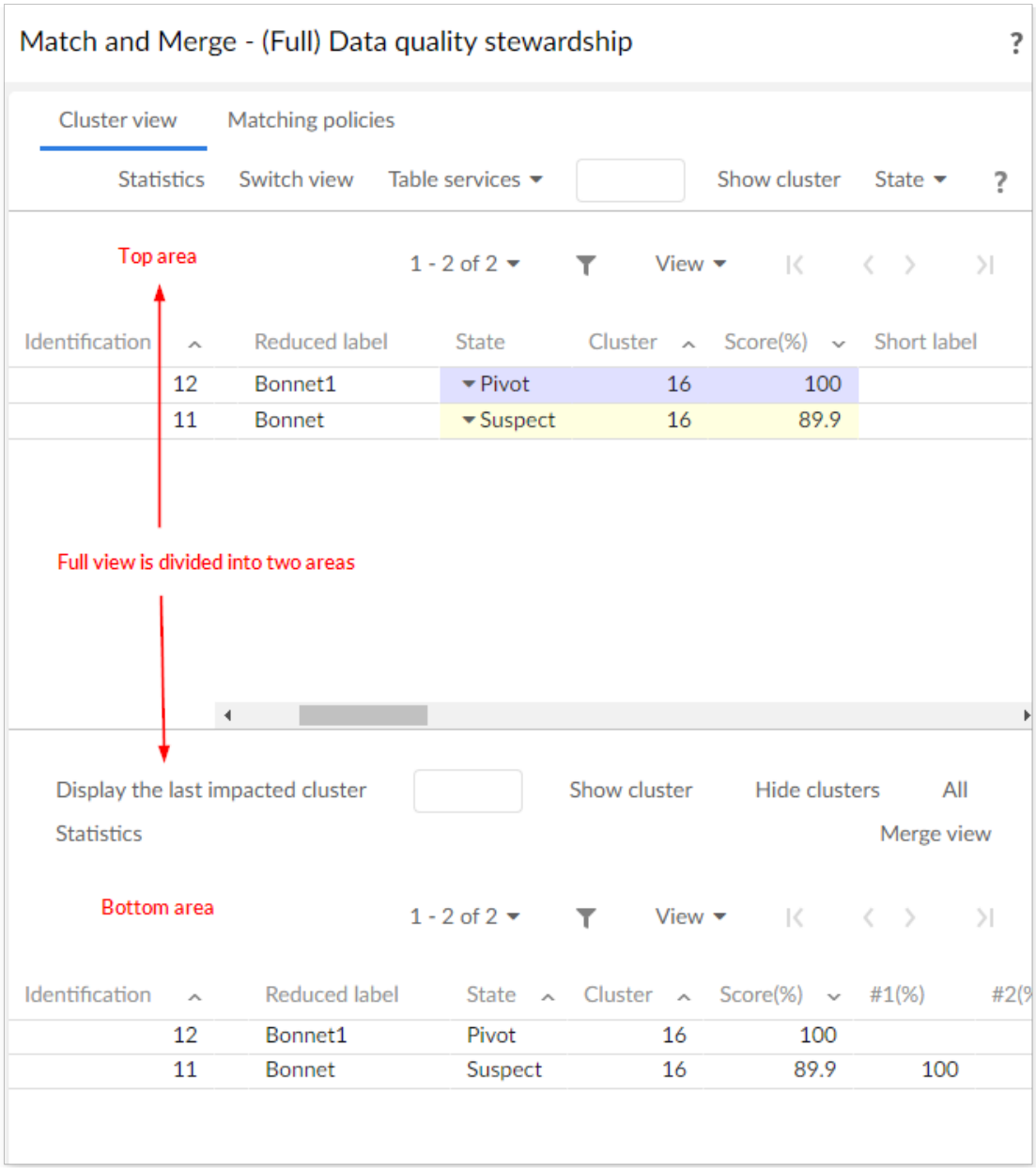


12.5 User services views

Full matching view

The full matching view is divided into two areas. The first area (top) displays a cluster's records or query results (**Search** and **State** menu buttons).

The second area (bottom) displays the cluster in which the last matching operation has been executed by the add-on.



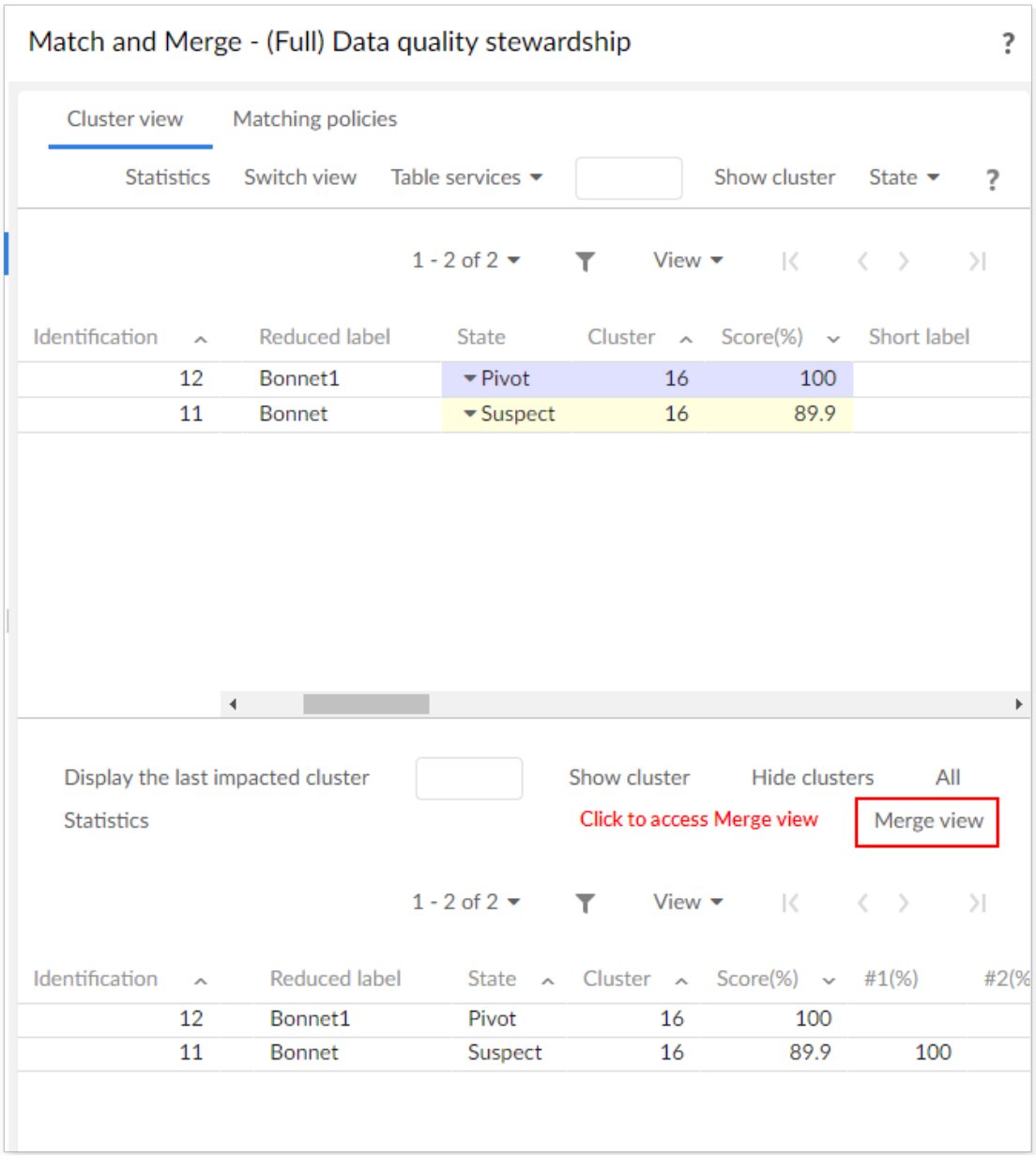
This second area is cumulative, meaning that it is possible to display as many clusters as needed. Only the first area holds the pop-up menu with matching operations and the **Table services** menu button that contains operations that can be applied to the table. The second area is used to display information and is helpful for analysis work.

Additional information displayed in the second includes: score of the first algorithm '#1(%)', score of the second algorithm '#2(%)', target record (used for merged record to keep a link to the record

benefiting of the merge), timestamp of last modification and was golden (life cycle of golden records over time).

Special notation:	
✔	To access to the merge UI, a cluster containing a pivot record must be displayed in the second area. **Note**: **only one cluster must be displayed to allow the merge**. The 'Show cluster' operation enables you to display the desired cluster. The 'Hide clusters' button can be used to clear this second area at any time.

The **Merge view** button appears on the right side below the first frame of the user interface (next figure).



When using the magnifying glass and applied EBX® search filters on the table, you may want to first use the **Switch view** button to display the first UI area in full page mode. This type of EBX® search is different from the matching view **Search menu** button. This button simulates matching

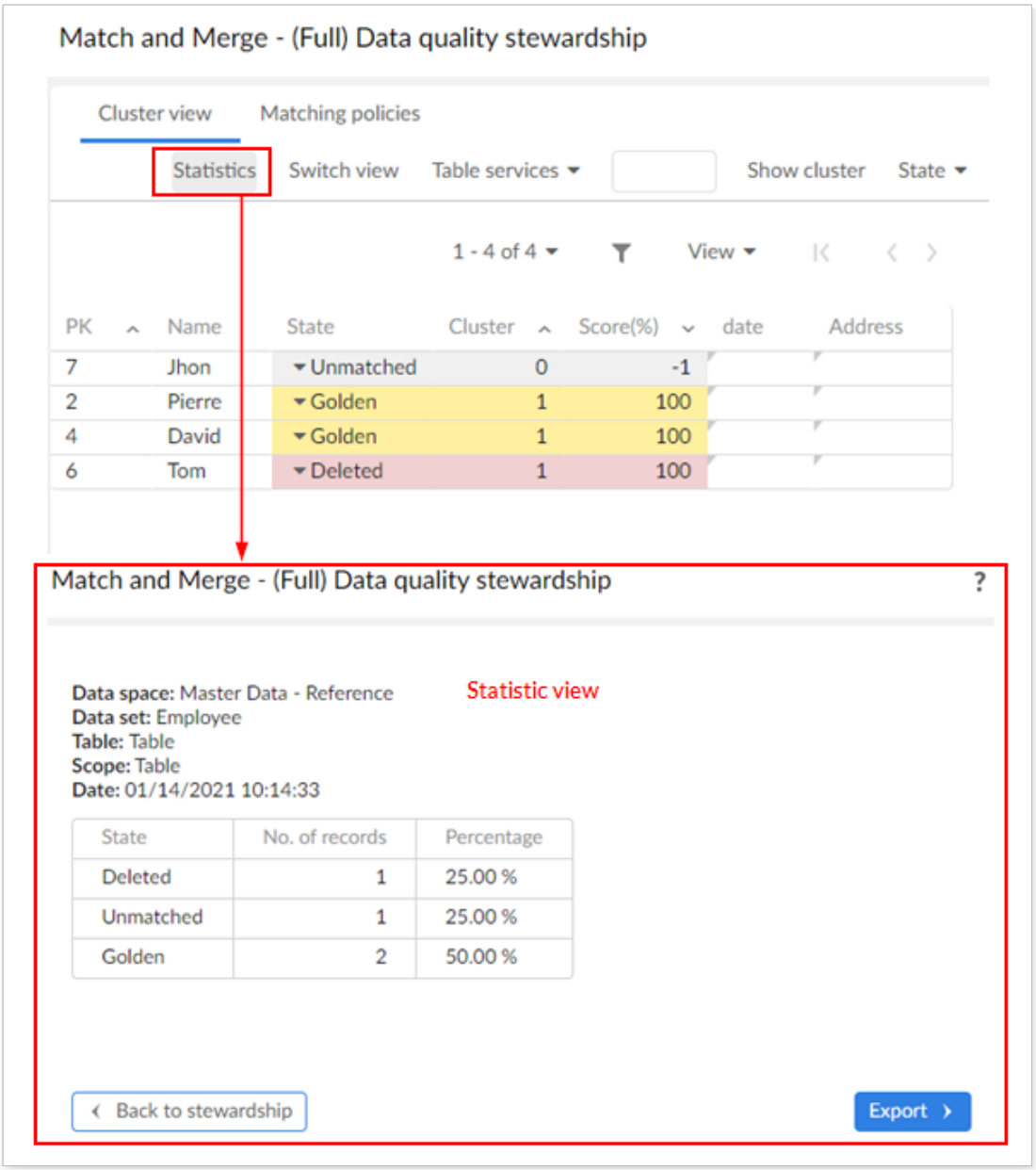
policy execution by matching fields, relying on matching algorithms and matching fields of the current configuration.

Matching view

Even when you hide **DaqaMetaData** in **Default view and tool**, the two columns **Score** and **Cluster** still can be seen in Matching view.

Statistics view

The **Statistics** button in matching views allows you to view matching statistics and export them to an excel file.



About the Merge view

The **Merge view** allows you to combine values from records to create a golden record. The number of steps required to complete the process depends on your data structure. The merge process may impact data not directly involved in the merge due to foreign key relationships to other tables. You will be given the opportunity to address each one of these dependencies.

Available actions and view concepts are discussed further in the following headings:

- Merge view actions [p 200] presents an overview of the view's interface.
- Steps to merge records [p 204] walks you through a basic merge scenario to create a golden record.
- Relationship dependencies [p 206] describes how the merge view handles certain foreign key relationships.

Merge view actions

Interaction with the **Merge view** falls into two basic categories—navigation and value selection. The add-on automatically guides you forward in the merge process, but at times you may need to return to a previous step. Selecting values to merge is pretty straightforward, but there are a couple of available options to help the process along.

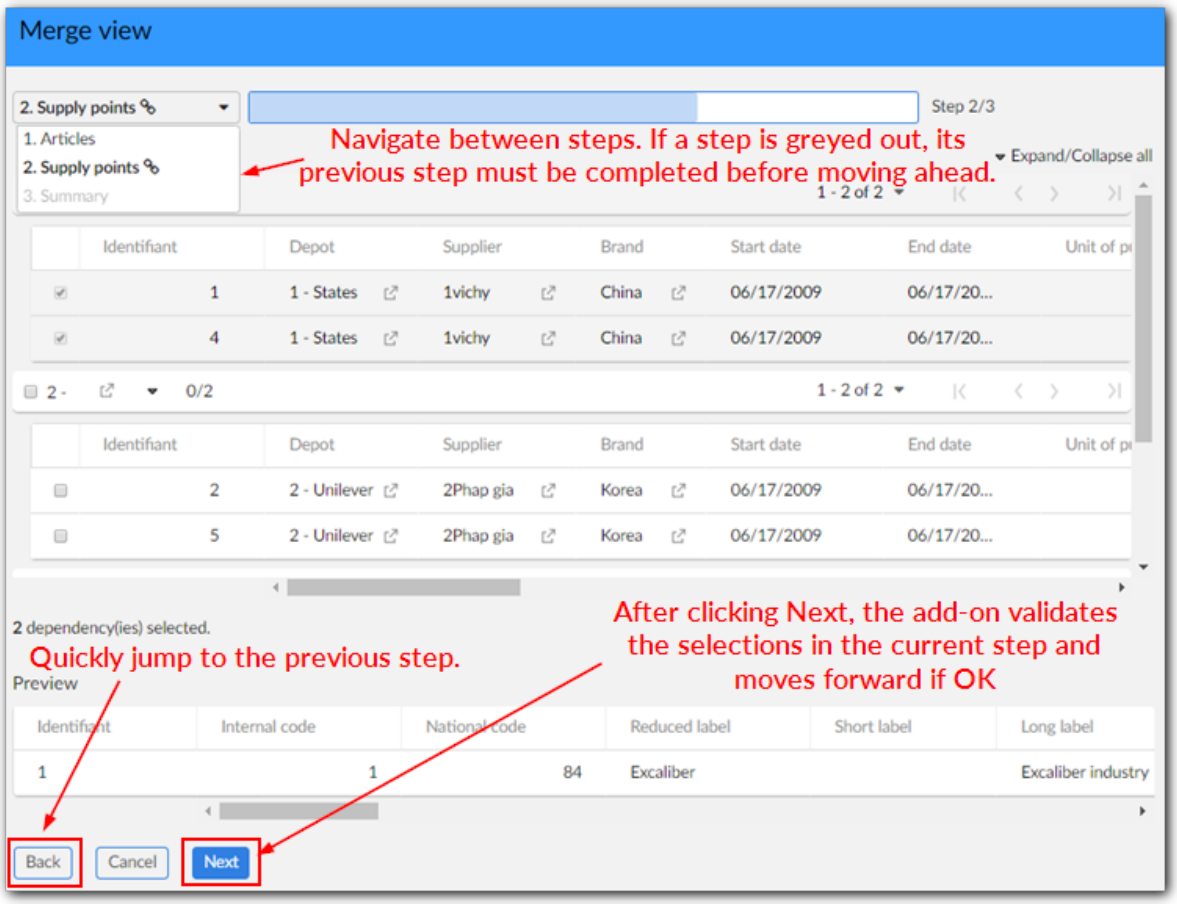
See the following headings for more details on these subjects:

- Merge view navigation [p 200]
- Value selection [p 201]

Merge view navigation

Navigation in the **Merge view** is fairly intuitive. The upper-left section of the view contains a dropdown list of all steps required to complete the current merge operation. The bottom of the page includes **Next** and **Back** buttons to navigate one step at a time. Depending on your data structure, a

merge may involve many steps. This is where the dropdown list provides an advantage by allowing you to jump back to any previously completed step.



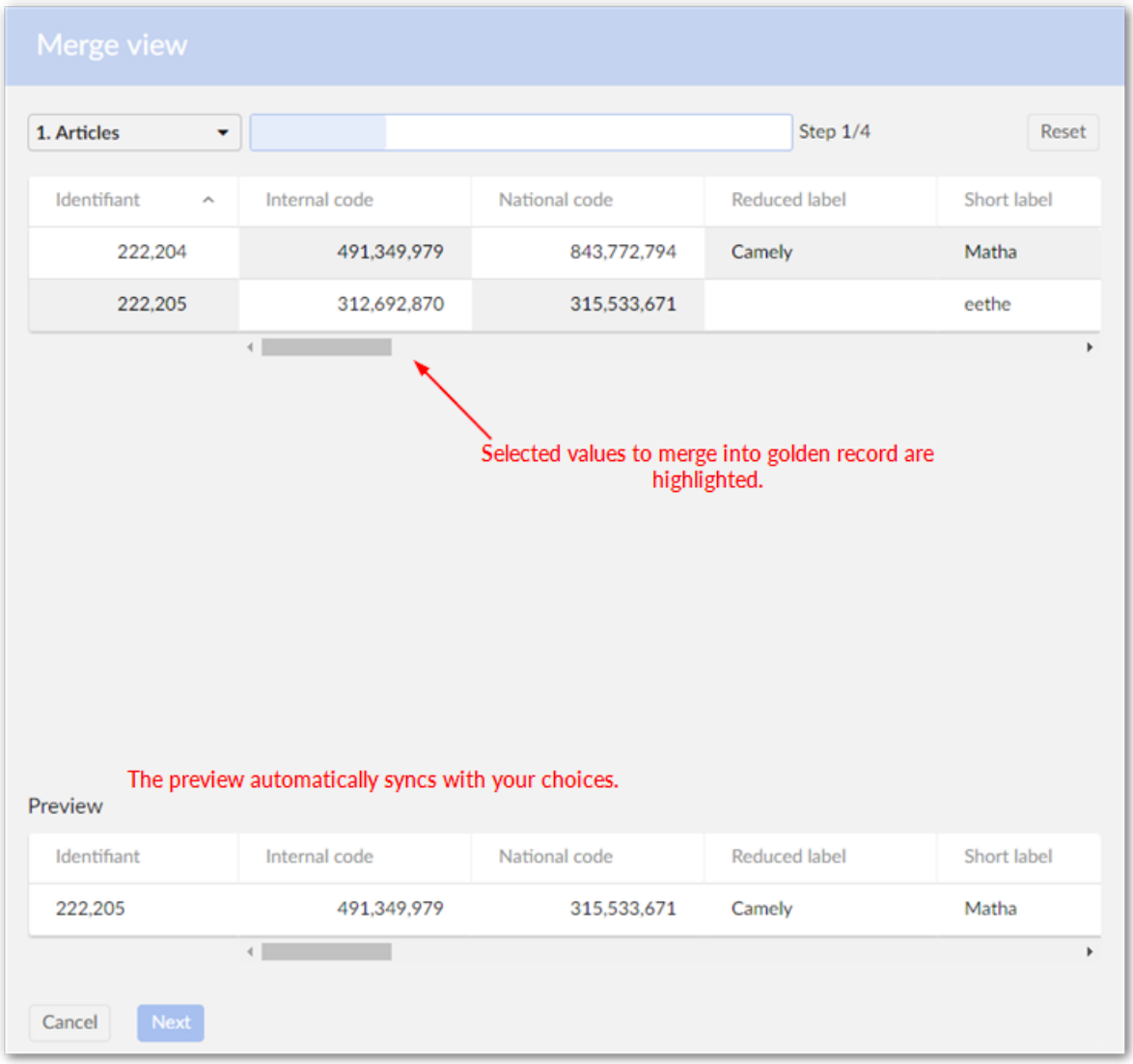
You have the ability to preview a record by hovering over its primary key field and clicking the icon. You can lock columns which allows you to scroll through others while still viewing the locked column. Click the icon when hovering over a column's title. Locked columns stack left to right after the primary key column. Additionally, horizontal scrolling is synchronized so that you can see the same information in both tables at the same time.

Value selection

Selecting values to merge happens in the following ways:

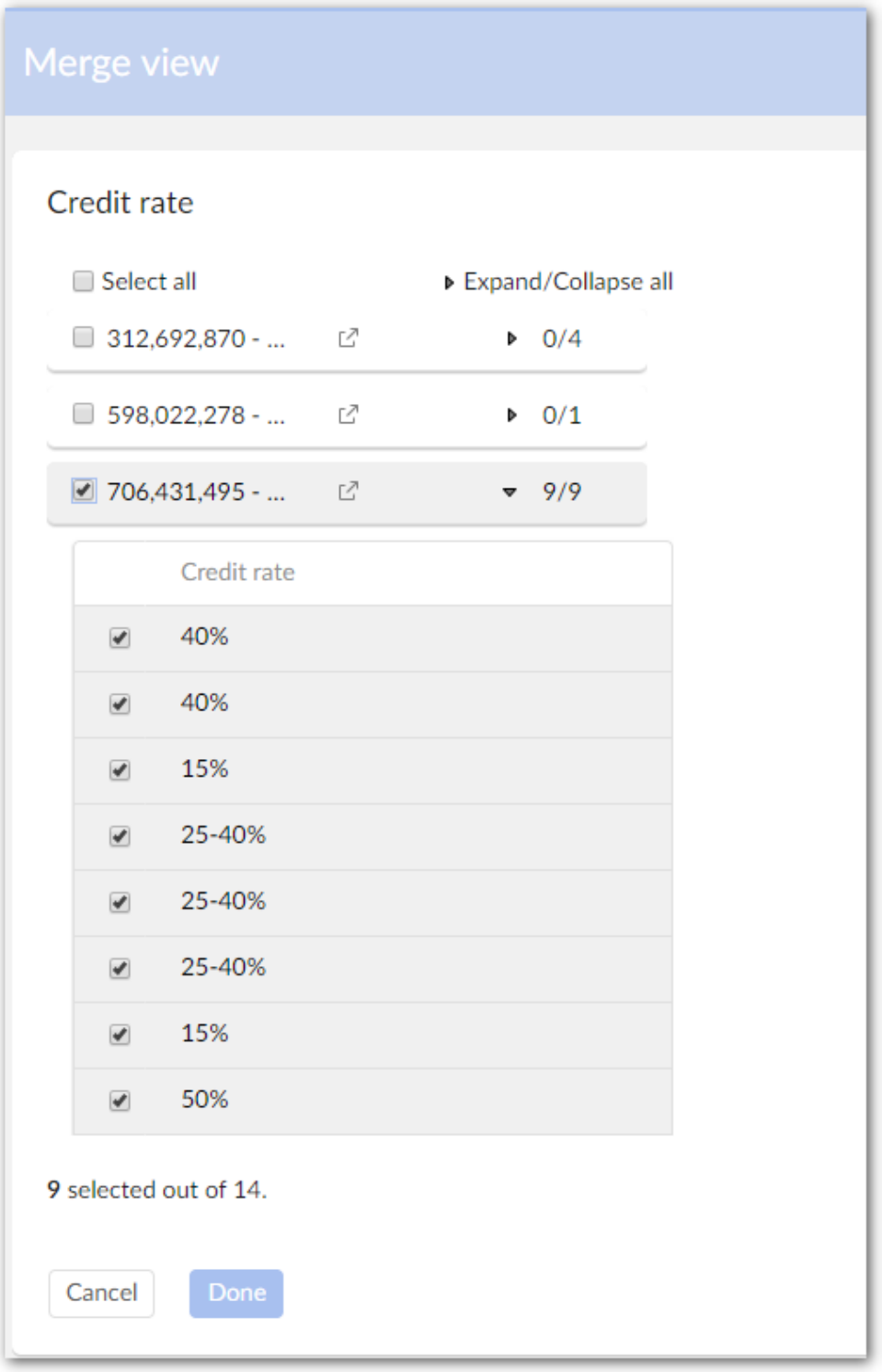
- **Selecting field values:** Click a field to select its value for inclusion in the golden record. If you click the first field of a record, the entire record gets selected. You may want to perform this step

first to provide a reference record that will be displayed in the preview section. All subsequent selections are reflected in the preview record at the bottom of the page.



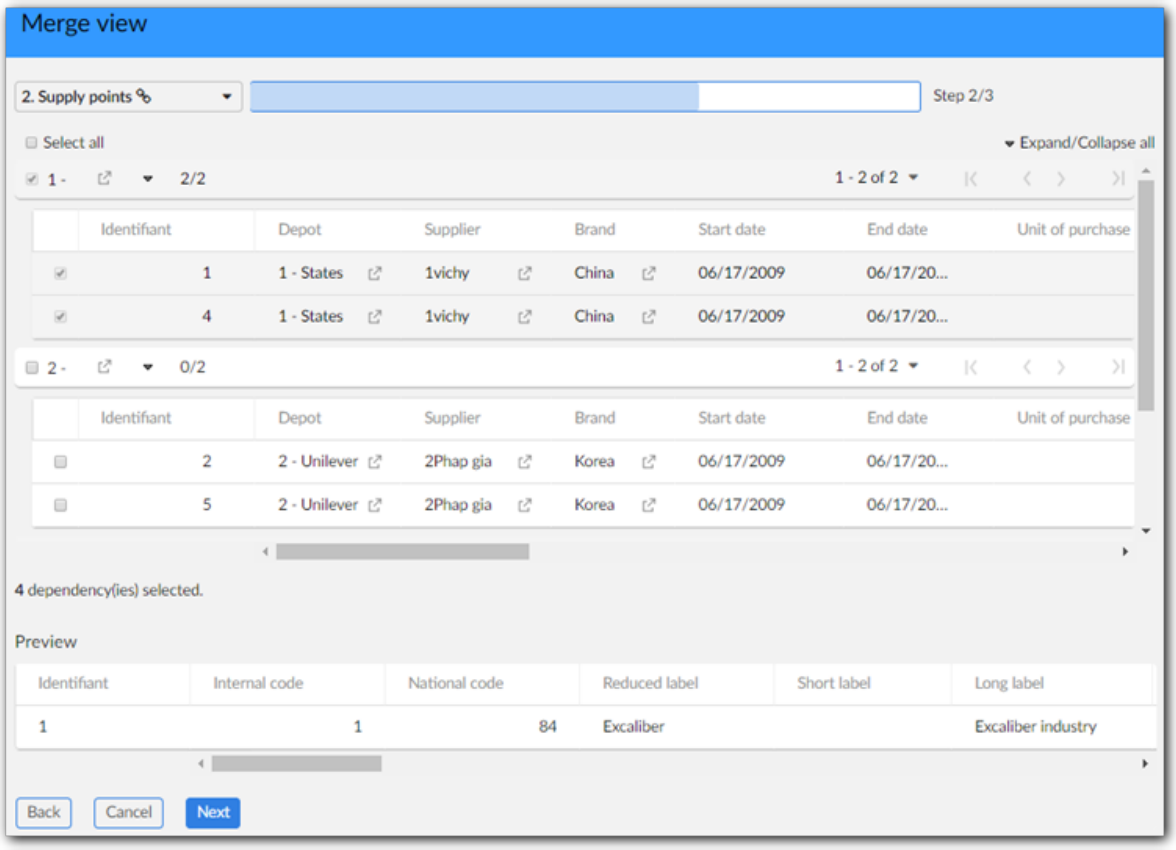
- **Selecting list values:** Some fields, such as enumerations and groups, can contain a list of values. You have control over which of these values are included in the merge. To choose the values,

click the field's icon and select the desired values. The counter shows how many of the possible values are included.



- **Selecting dependencies:** After completing the first page in the **Merge view**, you will be guided through choosing which dependencies (from related tables) are included in the merge. The add-on

presents a different page for each dependency. See Relationship dependencies [p 206] for more information on dependencies.



> **Note**
>
> The ability to select field values may depend on your level of permissions. For example, if you are not permitted to modify a specific field and an administrator has enabled the **Apply EBX® permission on merge view** property, the add-on will automatically set the fields value based on the Pivot record.

Steps to merge records

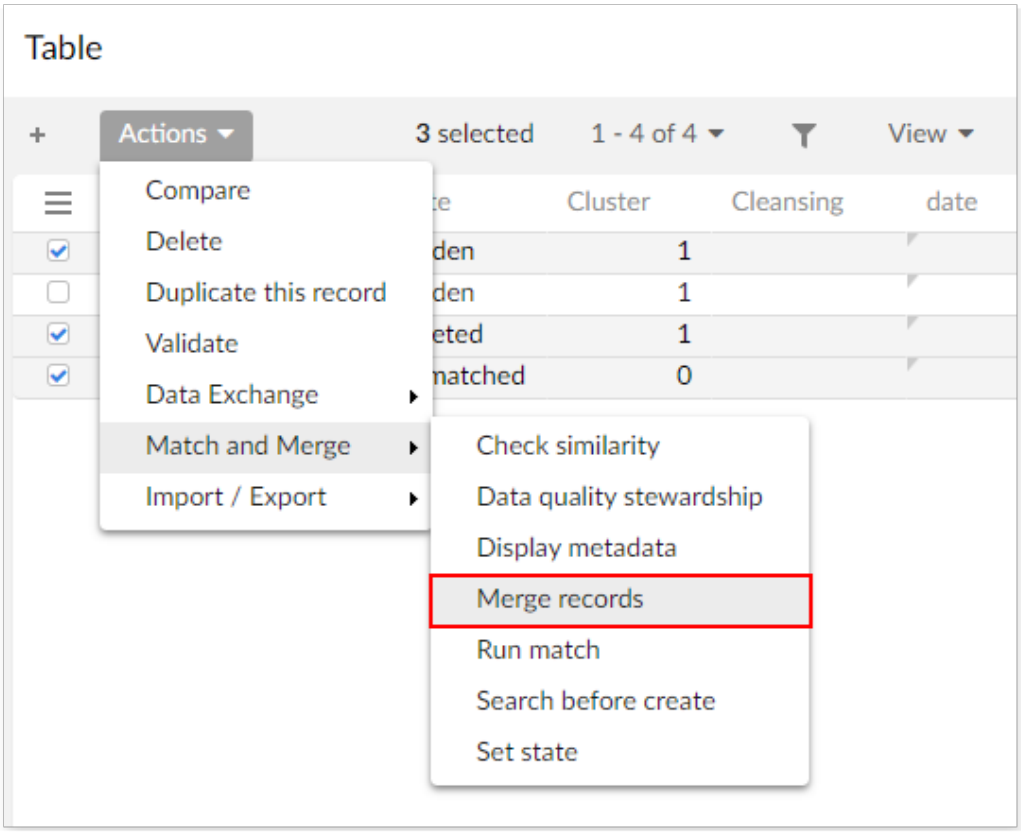
When merging records, you select the field values you want to include in the new golden record. The first page in the view allows you to select values. Each additional page will allow you to manage how the merge will impact related tables. The number of pages required to complete the merge process can vary depending on your data structure and configuration settings.

> **Note**
>
> The merge process can be interrupted depending on whether the **Check null input** is active. This option is set at the data model level and determines whether a constraint is placed on empty or null field values. When this option is not enabled, the add-on displays a warning message and allows you to complete the merge operation. However, if this option is enabled the add-on prompts you to input a value for required fields.

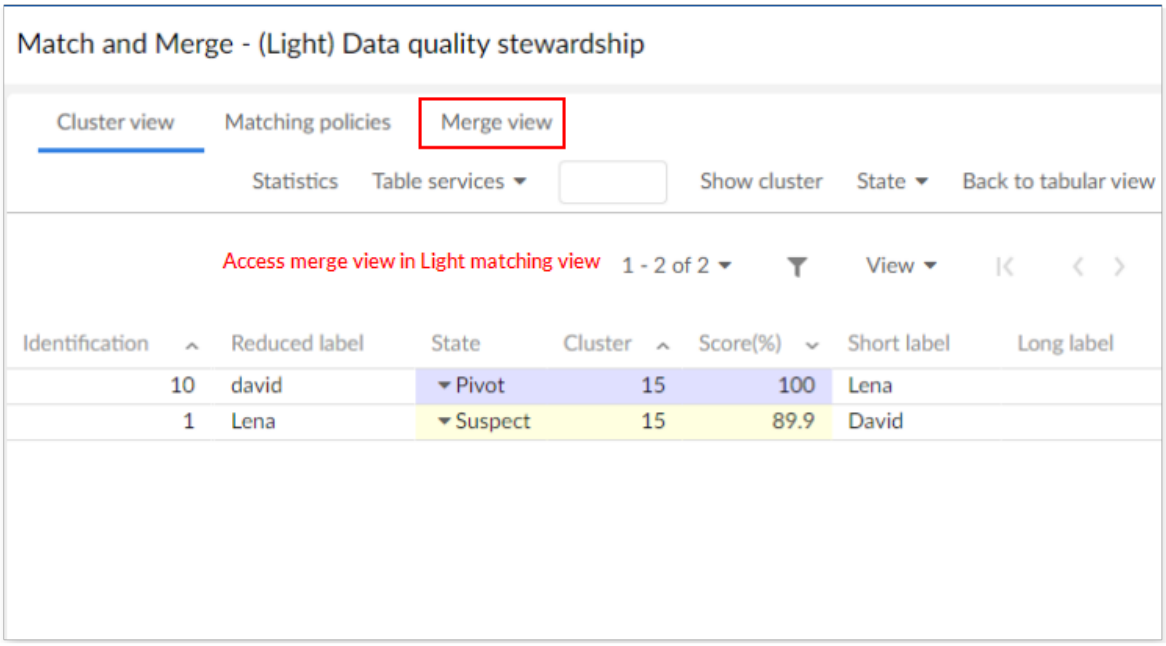
To begin the merge process:

1. Access the merge view in one of the following ways:

- When viewing a table: Select the records to include in the merge and from the actions menu choose *Match and Merge* > *Merge records*.



- After stewardship:



- After inline matching.

2. Click to select the fields you want to merge into the golden record. Clicking a record's primary key selects the entire record. You may want to select the most accurate record first to act as a baseline. Then you can fine tune values by selecting other fields. The choices you make are reflected by the preview at the bottom of the page. Once satisfied, click **Next**.

   > **Note**
   >
   > The ability to change the pivot record by selecting its primary key depends on the process policy settings configured by an administrator.

3. This step depends on your data structure and configuration settings:
   - If this merge will not impact other tables, or your configuration is set to ignore relationship dependencies, the add-on presents you with the **Summary** page. See the next step for more information.
   - If other tables can be impacted by this merge, you are presented with a page to select the relationships to update. The values from the pivot record are selected automatically when entering the **Merge view** after stewardship. If you do not have sufficient permissions to display the dependency, the add-on automatically makes the selection based on the pivot record.
4. If your process policy has the **Make definitive golden** and **Make golden** options enabled, this step displays and allows you to set the merged record to golden, or definitive golden. A golden record can be used in future matching operations, whereas a definitive golden record will not be used.
5. When you reach the **Summary** step, the add-on provides you an overview of what the merged record will look like. Click **Merge** to complete the process, or use the navigation dropdown to access previous steps to make changes.

Relationship dependencies

A merge operation can impact related table records. Depending on data structure and add-on configuration, new records may be created, or existing fields updated. When many dependencies exist, the add-on allows you to specify the number dependencies that display on the current page. If you navigate between pages, the add-on remembers any existing selections.

Administrators can configure the following merge behaviors that affect related records:

- Allow users to manually update.
- Enable the add-on to automatically update.
- Ignore relationships altogether.

Relationship tab

The **Table** configuration's **Relationship** tab displays information about tables related to a configured table. Administrators can set whether the relationship is used in merge operations.

The following table describes the **Relationship** tab's properties and options:

Property	Description
Relationship code	Any naming convention without white spaces.
Relationship name	The relationship name is automatically derived from the label of the related table.
Relationship management	**Manual**: Each relationship needs to be selected manually during the merge process. **Automatic**: Relationships will be automatically selected based on the pivot. **None**: No relationships are selected.

Limitations

The add-on can only lookup relationships that exist within the current data model. Additionally, the add-on cannot update dependent records in the current data model when the related record's primary key includes the foreign key of a merged record. Other relationships are updated automatically based on any selected pivot record.

Simple matching view

The simple matching view allows you to manage suspicious records and provides specific operations to accomplish this:

- **Make definitive golden**,
- **Make golden**,
- **Delete**,
- and stewardship execution when the merge process is needed (merge view)—namely, **Run stewardship on suspicious** and **Run stewardship on pivot**.

The suspicious records are created when add-on configuration is based on one of these use cases:

- **Simple matching view by using a workflow**. At submit time, if the record is suspicious, then a workflow is launched. In this case the **Simple matching** user task holds the parameters that can be used to configure the simple matching view and the merge view.
- **Simple matching view at the submit time**. At submit time, if the record is suspicious, then the simple matching view is automatically opened. In this case option configuration for the simple matching view and the merge view is done in the add-on configuration itself (EBX® administration area).
- **Simple matching view when the 'Match suspicious' operation is executed**. This operation is available in the light and full matching view. In this case option configuration for the simple matching view and the merge view is done in the data matching configuration itself (EBX® administration area).

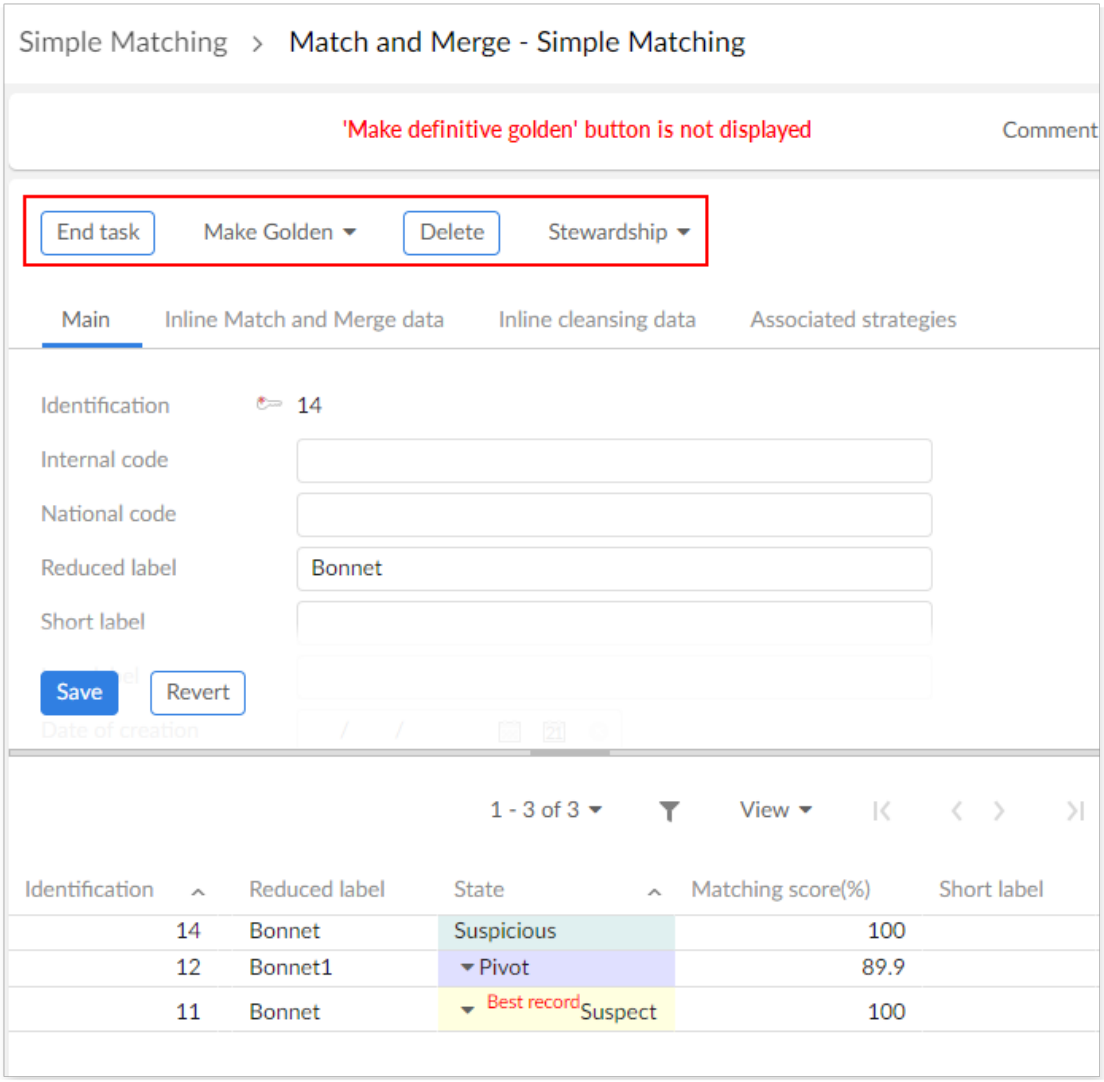
Stewardship actions available on simple matching:

Actions	Definition
Run stewardship on suspicious	Launches the 'Merge view'. When no auto-created record exists, the Suspicious record becomes the Pivot and survivorship rules are applied. If an auto-created record exists, it is prioritized to become the pivot and survivorship rules are applied.
Run stewardship on defined pivot	Launches the merge view. A record marked as Pivot becomes the Pivot. Suspicious records are matched against the Pivot and the add-on computes new scores of Suspicious records. The 'Merge view' only displays Pivot and Suspicious records. Survivorship rules are not applied. Note that this service does not display when the Suspicious record is auto-created.
Run stewardship on best record	Launches the 'Merge view'. A record marked as the best record becomes the Pivot. All potential duplicate records are matched against the Pivot. The add-on computes new scores and survivorship rules are not applied. Note that the add-on determines the best record using the configured survivorship function, but auto-created records are given the highest priority.
Reset pivot selection	Removes records' 'Pivot' designation.

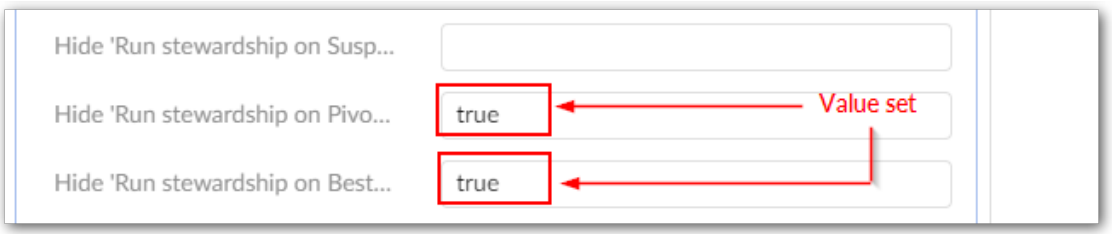
In the screen below, the simple matching view displays the suspicious record and offers three actions:

- **Make golden** (suspicious becomes golden directly without any merge),
- **Delete** (logical deletion)
- and run **Stewardship** actions.

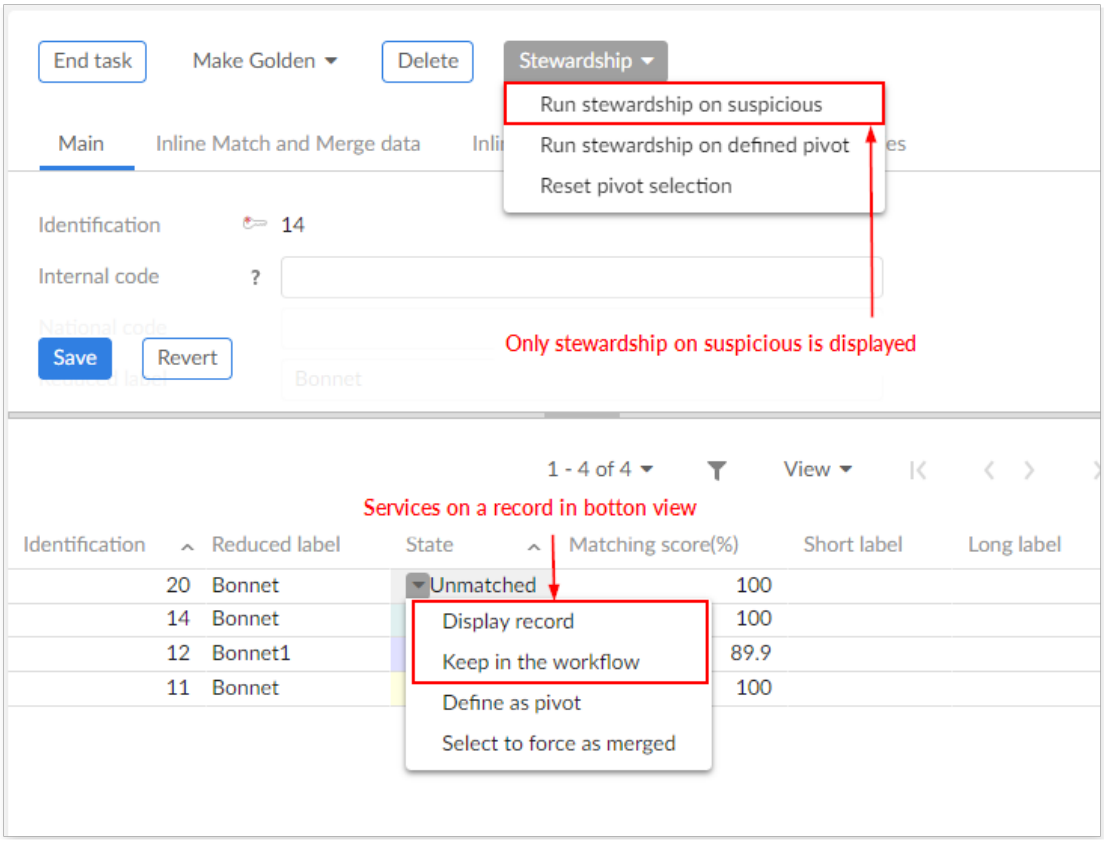
This example does not show all options such as **Make definitive golden**.



Depending on the configuration, it is possible to offer different modes for stewardship execution either on the suspicious record, on Best record directly or on the selected pivot record.



Here is an example with the **Run stewardship on Suspicous** option configured.



Operations to a single record:

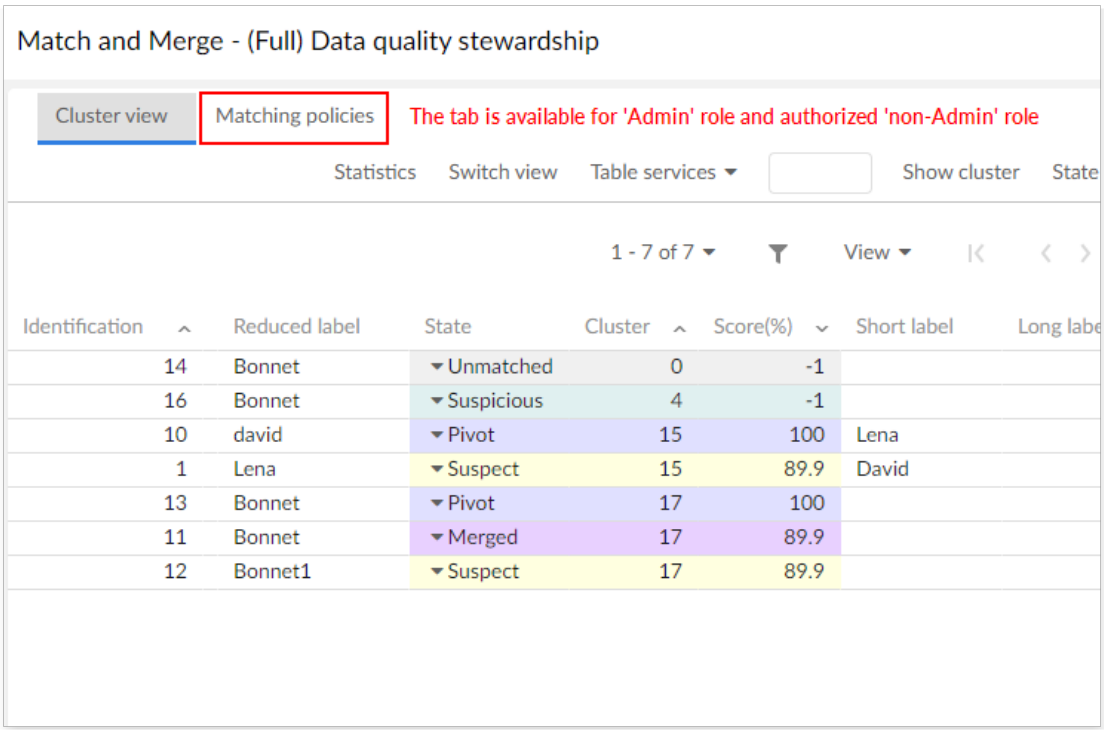
- Keep in the workflow - allows you to select one to many records to keep in the workflow context. Then, the next user tasks in this workflow can use these selected records depending on the functional needs.
- Define as pivot - allows you to select a record to use as the pivot record.
- Display record - displays record in a pop-up screen.
- Select to force as merged - marks the current record as Merged

Special notation:	
✔	The 'Match table' and 'Match cluster' operations still generate suspect, golden, pivot and merged records even when the simple matching mode is activated.

Policies view

By default, only administrators can access the **Policies view** tab. However, they can set permissions to allow other users to access this view. This view displays the current process policy and matching policy used by the add-on. Even though the matching configuration is achieved through the EBX®

Admin area, this view is helpful to get direct information about the process and matching policies that are being executed.



Special notation:	
✕	Policies view in consultation mode only. Policies view in restricted modification mode.

Scores view

Three scores are displayed as follows:

- #1(%) is the score computed by the first matching algorithm.
- #2(%) is the score computed by the second matching algorithm when it is configured.
- Score(%) is the score readjusted by the add-on to deal with false negative records.

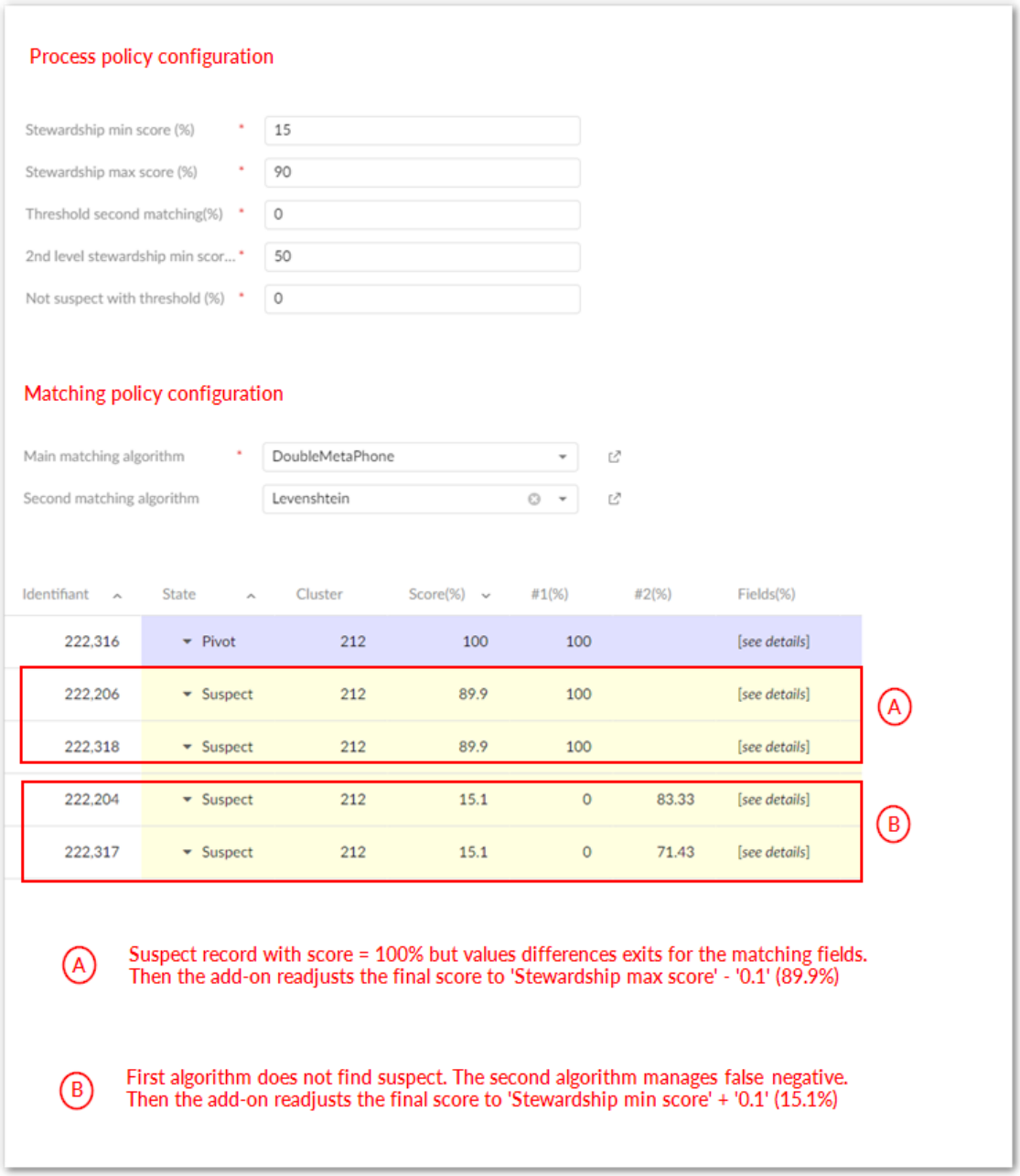
The rules applied by the add-on to readjust the scores are as follows:

- If (# 1 = 100 %) but matching fields values are not fully identical then Score = **Stewardship max score - 0.1**.
- If (#1 < stewardship min score) and no second algorithm then no readjusting of the Score occurs.

When the second algorithm is applied:

- If (#2 >= 2nd level stewardship min score) then Score #2 is persisted and Score = **stewardship min score + 0.1**.

- If (#2 < '2nd level stewardship min score') then no readjusting of the Score occurs.



CHAPTER 13

Record level matching

This chapter contains the following topics:

13.1 State association

Each record is associated with a well-defined state that indicates its match level: Suspicious, Suspect, Pivot, Golden, Merged, Unmatched, To be matched and Deleted.

Operational systems (transaction, business intelligence, reporting, etc.) should only be supplied with Golden records as input. Only these records are considered to have no risk of duplication with regards to other information contained in the dataset.

The set of available states is not extensible beyond the given eight states. It is a finite list governed by the add-on to carry out matching functions.

The following sub-states have been added: 'Was golden, 'Definitive golden', 'Ignore' (merged without target), 'Not suspect with', 'Under workflow', 'From match at once', 'From group at once' and 'Merge'.

13.2 State overview

The add-on can be configured either to execute direct or simple matching. Interaction with each State depends on the type of matching used.

When direct matching is configured, the new or updated record's state is computed directly by the add-on. For example, the record can be considered as a golden record directly, or a pivot with related suspect records.

When simple matching is configured, then every new or updated record is matched by the add-on but the decision regarding its state requires human action. The add-on moves the record into an intermediate 'Suspicious' state. However, if the creation or modification of a record results in a matching score that is compliant with the level of a golden record, then the add-on puts the record into a golden state and bypasses the human decision.

The automatic merge is not applied when the simple matching configuration is used. Indeed, a human decision is mandatory to manage the suspicious record.

13.3 **State definitions**

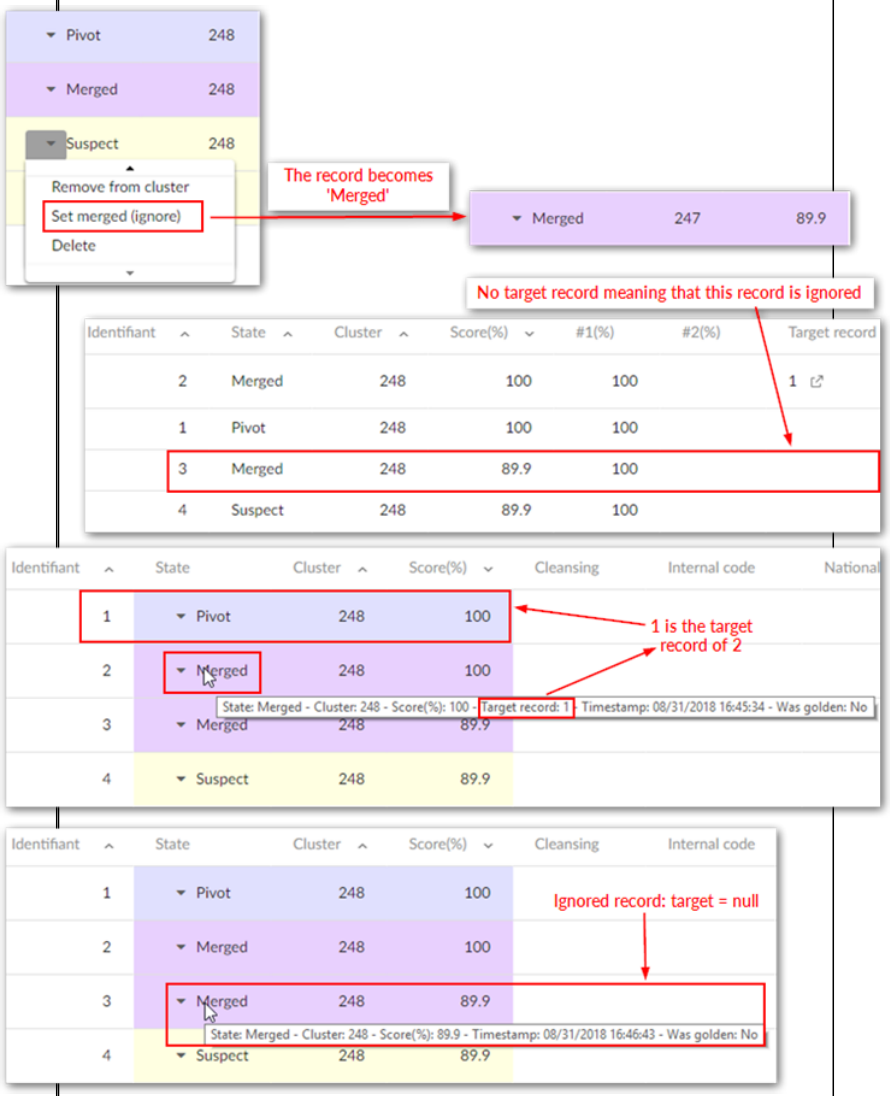
Record level matching (State)	Definition	Example
Suspicious	A record that likely has duplicate records. The add-on can be configured to suggest that the record is a suspicious record and needs to be verified. A suspicious record has not been classified as a golden, pivot or suspect record. A user task is required to decide the suspicious record's next state: golden, pivot (to be merged with suspect records) or deleted (see the definition of the 'Pivot' and 'Suspect' states). This only applies when the add-on is configured for 'Simple matching'. 'Suspicious' records are systematically attached to the '004' cluster. They are not attached to a specific cluster identifier with a set of suspect records, since suspicious records are not yet considered as pivot records.	A user creates the employee record 'Mooree' that has two employees as potential suspect records: Moore and Zooree. The 'Mooree' record is the suspicious record and a user decision is needed to decide if it becomes a golden record directly (the suspicious record being considered as a legitimate unique record), or if the record creation is canceled (the user determines that the suspicious record already exists, please refer to definition of 'Deleted' state). In these cases there is no change applied on the Moore and Zooree records. But the user can also decide that Mooree is a duplicate and to set it as a pivot against Moore and Zooree for subsequent merge (Stewardship task). In this situation these records are changed to suspect records automatically (refer to definition of state Pivot and Suspect), based on the user's decision.
Suspect	A record that is potentially a duplicate against a record with either a 'Pivot' or 'Golden' matching state. The add-on computes a similarity percentage score against the pivot or golden record. Then, a set of suspected related records is grouped into a cluster with a unique identifier.	The employee 'Mooree' is a suspect towards 'Moor' with a similarity percentage level of 80%. Moor is the record considered as the pivot or golden record.
Pivot	A record that is likely to be the best record to use amongst a set of suspect records. The similarity percentage of a pivot record is systematically set to 100%.	Group of records • Moore, pivot, 100% • Mooree, suspect, 80% • Zooree, suspect, 60% Another group of records • Bonnet, pivot, 100% • Ponney, suspect, 76% • Bonpey, suspect, 20%
Golden	A record that is either a unique record to be used directly, or the best record to use amongst a set of former suspect records. A former suspect record is one that has since been merged into the golden record. This merge can include specific fields only, use the entire suspect record or reject the suspect record (that is, ignore it).	Helen, golden, 100% Bonnet, golden, 100% Group of records • Moore, golden, 100% • Mooree, merged, 80% • Zooree, merged, 60%

Record level matching (State)	Definition	Example
	Operational systems (transaction, business intelligence, etc.) should only be given golden records as input. The similarity percentage of a golden record is systematically set to 100%.	
Merged	A record that has undergone a merging procedure into a pivot or golden record. A merged record keeps track of its targeted record. If a merged record is moved to a new cluster, its target record is updated to the new cluster's golden/ pivot record.	Group of records • Harry, pivot, 100% • Harrryy, merged, 80% • Barry, suspect, 60%
Unmatched	A record that is not yet matched. Its similarity percentage is not relevant and is thus set to '-1'. An unmatched record is attached to the predefined '000' cluster or located in a normal cluster through the 'Group at once unmatched' operation. This state is used when a table under add-on control is temporally deactivated (see the 'On matching Process', 'On creation' and 'On modification' properties in the Process policy configuration).	Jonnhy, unmatched, -1 Jhonny, unmatched, -1 Petrian, unmatched, -1
To be matched	A record that is not yet matched but is waiting for a match. This state is used when importing data. During the import all records default to this state. Then, a service allows you to match all 'To be matched' records at once. Its similarity percentage is not yet relevant and thus is set to '-1' by default. 'To be matched' records are attached to the '002' predefined cluster or located in a normal cluster through the 'Group at once to be matched' operation.	Dupand, to be matched, -1 Johnway, to be matched, -1
Deleted	A record that has been logically deleted. This is a logical deletion.	Dobbon, deleted

Table 50: Level of matching - state record definition

13.4 **Sub-State definition**

Record level matching (Sub-state)	Definition	Example
Was golden	Boolean value A record is 'Was golden'='Yes' when it was a golden record before its current state.	Time 01, creation of a direct golden record • Bonnet, Golden, Was golden='No' Time 02, after a match table • Bonnet, Pivot, Was golden='Yes'
Definitive golden	A definitive golden is no longer used when a match against the table is performed. Definitive golden records are located in the '003' predefined cluster.	Time 01, direct golden creation and set up to definitive golden by a user: • Bonnet, golden, ClusterID='003 Time 02 and after: • Every match table no longer uses or modifies records known as definitive golden.
Ignore	A record that is set as not relevant for merging. The ignored record moves to the merged state with no target value (no actual merge).	Harry, pivot, 100% Harrryy, merged, 80%, target=Harry Barry, merged, 60%, target=null In this example the record Barry has been ignored (state merged with target=null).

Record level matching (Sub-state)	Definition	Example
		
Not suspect with	List of records against which this record is not considered a suspect record during subsequent matching operations.	Bonnet, 'Not suspect with'='Monnet, Monney' Means that the Bonnet record will no longer be considered as a suspect against the Monnet and Monney records.
Under workflow	A record that is under workflow control.	Bannet, Suspicious, workflow=management of suspicious records
From match at once	A record whose latest modification has been done by a 'match at once' operation (operation applied to a set of records).	First stage: Pannet, Dannet, Ganet are all unmatched records After executing 'match at once unmatched records' the add-on keeps in memory that these records have been modified by the match at once operation: Pannet, pivot

Record level matching (Sub-state)	Definition	Example
		Dannet, suspect, 50% Ganet, suspect, 30% As soon as an operation is performed on any record, then it is no longer considered as a 'From match at once' record.
From group at once	A record that has been grouped in a cluster with the 'Group at once (unmatched)' service or 'Group at once (to be matched)' service.	Cluster 233, Jonnhy, unmatched, -1 Cluster 233, Jhonny, unmatched, -1 Cluster 233, Petrian, unmatched, -1
Merge	When a record has been merged, this sub-state indicates whether the merging has been done manually by a user, or automatically by the system (survivorship).	Cluster 111, Bonnet, golden Cluster 111, Mozzet, merged, User Cluster 111, Bonnnet, Auto

Table 51: Level of matching - sub-state record definition

CHAPTER 14

Cluster management

This chapter contains the following topics:

14.1 Definition

When the system searches for duplicate records, one or multiple records can be found. All these records are grouped under a unique identifier to form a cluster of potentially related records. The add-on automatically adds this identifier to the record as metadata. The following are some example clusters:

Cluster	Record
Cluster 020	Bonnet, pivot, 100% Ponnet, suspect, 80% Monney, suspect, 70%
Cluster 023	Helen, golden, 100% Halen, merged, 85% Bellen, merged, 80% Heelen, merged, 91%
Cluster 024	Durand, pivot, 100% Dupand, suspect, 60%
Cluster 025	Lapetina, golden, 100% Labetina, merged, 90%

Each table has its own cluster management. In other words, a cluster is limited to the scope of a table and remains fully autonomous from relationships between tables. A record is attached to one cluster only and can move from one to another depending on matching procedures executed on the table.

> **Note**
>
> Each cluster can only have one auto-created record.

When using matching for bulk data imports, the add-on creates as many clusters as necessary to put all duplicate records into groups. This allows you to manage each cluster independently when human decisions are required. Each cluster's identifier is automatically computed by the add-on. The first eleven clusters are reserved by the add-on (000 to 010).

Cluster ID	Definition
000	Groups all unmatched records not integrated in a group (Groups are created when the service Group at once (unmatched) is used). These records have not yet undergone matching so do not yet belong to a cluster of suspects.
001	Groups all golden records that are not attached to clusters. These records appear in two situations: when a golden record is created without any suspect records, or after all merged records in a cluster have been purged, which moves the golden record to this cluster.
002	Groups all records that are 'to be matched'. These records are waiting to be matched as a batch. This record state is used during bulk data import.
003	Groups all records that are 'Golden' and stated as definitive golden. These records are no longer used when matching is executed.
004	Groups all records that are 'Suspicious'.

Table 52: Predefined cluster ID

Special notation:	
✖	Cluster 005 for record in failure

14.2 Cluster life cycle

When matching a record against the table, the cluster identifier computation can fall into one of two scenarios:

- All suspect records are located in the same cluster. By default, this cluster is then reused and all existing records are kept even though there are no longer any suspect records for new matching. The score of these records is set to '-1'. The add-on applies a retention strategy on the existing cluster. This strategy can be changed by setting the 'On suspect record retention' property to false

in the process policy. With this configuration, the suspect records that no longer be matched with the pivot record move to the 'unmatched' state in the '000' cluster.

- Suspect records are located in different clusters. In this case, a new cluster is created to group all suspect records. In such a cluster, there are no records with a score set to '-1'.

Step	Record, state, cluster, score → Action	Description
1	R3, pivot, 298 R1, suspect, 298 R2 golden, 1 → Action: create R4	A number of records exist in the table when the record R4 is created.
2	R1, suspect, 298, -1 R4, pivot, 299 R3, suspect, 299 R2, suspect, 299 → Action: make record R1 pivot, match it against table	Record R4 matches with records located in two different existing clusters. Thus, a new cluster ('299') is created to group all suspect records.
3	R1, pivot, 299 R3, suspect, 299 R2, suspect, 299, -1 R4, suspect, 299, -1	Record R1 then matches with records that are all located in the same cluster. The existing cluster ('299') is reused. R2 and R4 are former suspect records. Since they no longer match against the new pivot R1, their scores are set to '-1'

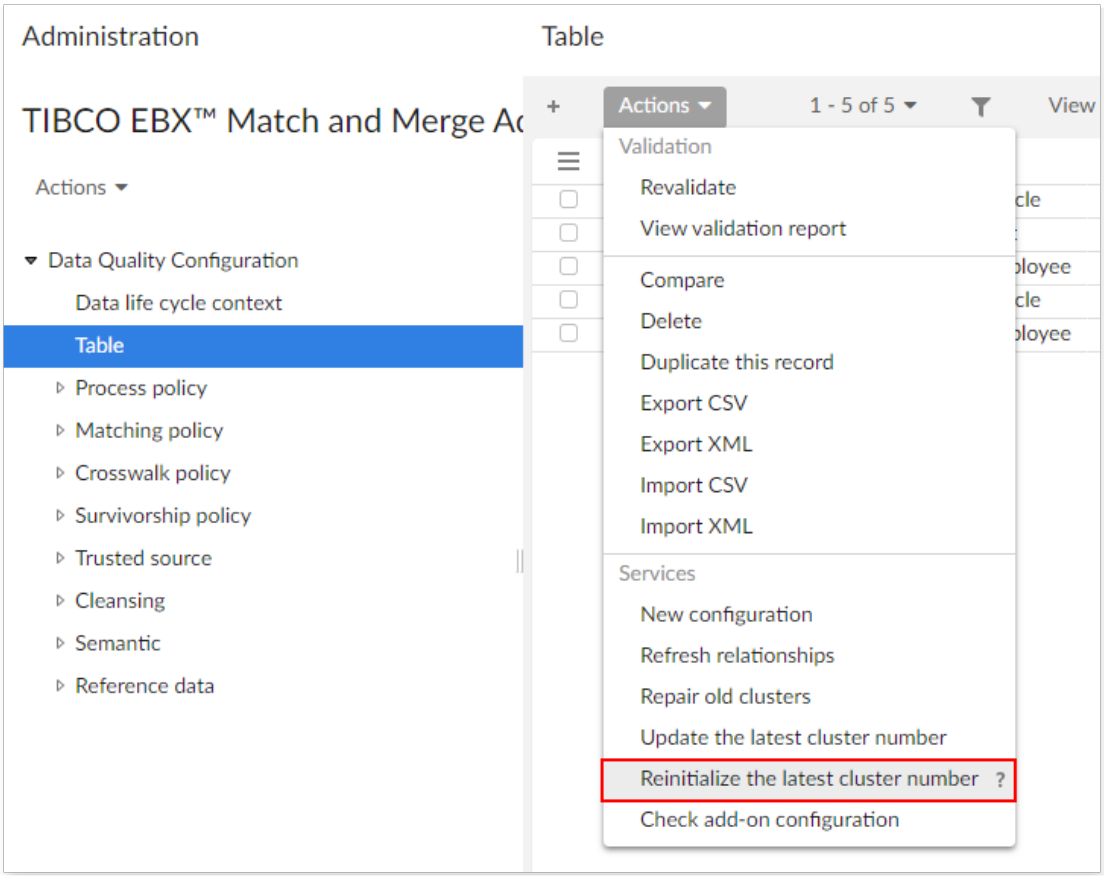
Table 53: Cluster life cycle

14.3 Reinitialization of the latest cluster number

Every table under the add-on control holds a 'Latest cluster number' property. This value is shared by all instances of a table located in different dataspaces and datasets.

This value is incremented by one when the add-on needs to use a new cluster. The clusters empty out when certain matching procedures are executed, such as a purge of records, a 'set definitive golden' or in a cluster with one record. In these types of cases it can be useful to reinitialize the latest cluster number value to get a fresh start. For instance, if the last value is '1,200' and after a purge only a few clusters are used, then it could be useful to reinitialize the latest cluster number to get, for instance, the value '20'. It can be easier for a user to handle clusters from '20' rather than from '1,200'.

This reinitialization is not mandatory, but it is helpful and makes using the cluster numbering easier. If you have administrative privileges, you can run the service from the location shown in the image below:



14.4 Updating the latest cluster number

It can be important to keep cluster numbers up-to-date to avoid any numbering conflicts. When updating the latest cluster number, the add-on loops through all datasets that contain the selected table and locates the largest cluster number. The table's 'Latest cluster number' property is then updated to this cluster number. This happens automatically when importing a matching configuration.

To manually update the cluster number:

- Navigate to *Administration* > *Data quality & analytics* > *TIBCO EBX® Match and Merge Add-on* > *Data quality configuration* > *Table*.
- From the **Actions** menu, select **Update latest cluster number**.

CHAPTER 15

Use case

This chapter contains the following topics:

15.1 Table with the 'On matching process' parameter active and an active policy

'On matching process' is a parameter in the table configuration. In this use case, 'On matching process' is set to 'Yes' to ensure matching execution.

Creation

When a record is created, matching automatically enforces a match against the table. This matching process uses records with unmatched, suspect, pivot and golden states only. 'To be matched' records are not used because they are processed by the 'match at once' operation. Any 'Suspicious' records are not used because they require a human decision. Any 'definitive golden' records are no longer matched. Records with 'merged' and 'deleted' states no longer hold matching value.

The result of this matching is one of the following:

- Golden record in the '001' cluster.
- Golden record in a non-predefined cluster. This implies a golden record that underwent an automatic merge.
- Suspect record in a non-predefined cluster.
- Pivot record in a non-predefined cluster.
- Merged record in a non-predefined cluster.

When 'simple matching' is configured then the possible results are:

- Suspicious record in the '004' cluster.
- Golden record in the '001' cluster.

Modification

When a record is modified:

- The modification of a 'to be matched' record does not result in a matching operation.
- The modification of a golden, suspect, pivot, unmatched or suspicious record results in a matching operation against the table (same as for the creation).
- The modification of merged and deleted records is not permitted.

Record deletion

When a record is deleted its state is set to 'deleted' and it stays in its current cluster. This is a logical deletion only.

15.2 Table with the 'On matching process' parameter deactivated

'On matching process' is a parameter in the table configuration. In this use case 'On matching process' is set to 'No' to disable matching execution.

Record creation

When a record is created, the matching sets its state to 'unmatched' and places the record in the '000' cluster. Matching is not executed.

Record modification

When any field of a record is modified, the matching sets its state to 'unmatched' and places the record in the '000' cluster. Matching is not executed.

Record deletion

The add-on executes the same operation as the one used when the 'matching process' parameter is active. This means that the record's state is set to 'deleted' and it stays in its current cluster.

15.3 No table configuration or policy trouble

When a table under add-on control is not configured or faces trouble with its process or matching policy (impossible to match any field due to error in the matching configuration), then the matching runs as if the 'On matching process' parameter were set to 'No'.

CHAPTER 16

Golden record life-cycle

The purpose of the EBX® Match and Merge Add-on is to find duplicate records which leads to creation or definition of a Golden record—a record that is either a unique record to be used directly, or the best record to use among a set of former suspect records—its life-cycle is described in detail in the following sections.

This chapter contains the following topics:

16.1 'Was golden' indicator

A golden record is important for any system because it is considered to be the best value that a table can provide from a set of suspect values. Each suspect that has been merged into a pivot or a golden record is easily traceable by means of a link automatically supplied by the add-on from the merged record to the pivot or the golden record. This makes it simple to retrieve a list of all records that have been merged into a pivot or a golden record.

The golden record can subsequently undergo other matches. For example, when creating a new record in the table, if this record matches with a golden record, the golden record is moved to the suspect state (or pivot if the 'Golden is preserved for selection record' property is set to 'Yes'; refer to matching policy configuration). When this situation occurs, it is no longer possible to get all past golden records. This can be of concern when systems must use golden records for update processes. In such situations, a suspect record that used to be a golden record will not participate in such processes.

To fix this issue, the add-on provides a field named 'was golden' that automatically follows this life cycle:

- When a golden record is created, 'was golden' is set to 'No'.
- When a golden record changes to another state (suspect, deleted, etc.), the 'was golden' is set to 'Yes'.
- When a record goes into the golden state then 'was golden' is set to 'No'.

To get a full list of all current and past golden records, this data can be queried with EBX® directly.

By default, the 'was golden' value cannot be changed manually by an end-user. Only data repository administrators can change its value. Nevertheless, there are no restrictions preventing a custom

procedure from modifying the information under certain conditions. For example, once legacy systems have been aligned with past golden records, a realignment of the 'was golden' values can be performed.

16.2 Set back golden

When a record's 'was golden' property is 'Yes' and it is in the suspect or pivot state, the user can set its state back to golden, bypassing any matching or merging operations.

During enforcement of survivorship policies, a suspect with a 'was golden' state of 'Yes' can be merged automatically. The record then goes into the merged state with 'was golden' still equal to 'Yes'. In this situation, the add-on does not allow the 'set back golden' operation as a merge has occurred.

16.3 State as 'definitive golden'

A 'Set definitive golden' service can be applied to any suspect, pivot, unmatched or golden record. The record moves to the reserved '003' cluster with its state set as golden. When a matching is executed at any level (table, cluster, real-time, batch) all records located in the '003' cluster are ignored.

Services that are available for golden records in the '003' cluster are the same as for other golden records in other clusters.

CHAPTER 17

Permission

All services encompassed by the add-on are under permissions enforced by EBX®. However, for the merge user interface, it must be taken into consideration that-at the default level-permission applied to the data is not yet enforced. This means that only a user with appropriate access must be authorized to merge records because all data are directly available. It is possible to force the application of EBX® permission on the merge view from the process policy configuration ('Applied EBX® permission on merge view').

CHAPTER 18

Matching metadata

The add-on enriches each table's data structure with a generic data type that provides the metadata required for matching and cleansing. These data values cannot be modified by the user. Their life cycle is fully managed by the add-on.

Properties	Definition
State	Record matching state.
Cluster ID	Identifier of the cluster. Clusters '000' to '010' are reserved by the add-on
Score (%)	Match score of the record against its pivot or golden record. When the score has not been computed it is set to '-1'. When a suspect is considered even though its score is lower than the minimum stewardship threshold then its score is set to '-1'. This situation occurs when the add-on must keep a record in an existing cluster not matter what new computations of this record's score are (retention strategy).
Score first matching (%)	Internal score of the record computed by the first algorithm.
Score second matching (%)	Internal score of the record computed by the second algorithm.
Field score (one to many)	Score for each field participating in the matching: Score (%), Score first matching (%), Score second matching (%).
Simple matching score (%)	When simple matching is used this score is against the suspicious record. This is a transient score since the record is not yet a suspect (its state does not change). The record is just a potential suspect against the suspicious record.
Target	When the state is 'merged', this value provides the link to the pivot or golden record into which this record was merged. Except for a 'merged' record that is ignored. In this case the target still remains undefined.
Merge by	Indicates how the merge has been executed either by a user or automatically by the add-on (survivorship procedure).

Properties	Definition
Timestamp	Date and time of the last modification of any kind applied to any field of the record other than the matching metadata (except state value). When the state is updated then the timestamp is updated as well.
Was golden	Indicates that this record was previously a golden record.
List of not suspect with	List of records against which this one is not a suspect record. Takes effect for all matches over time (Multi-valued complex data type (foreign key to the record, score when the not suspect was saved). The score of each record is saved to apply the 'not suspect with threshold (%)' property of the process policy.
Auto-created	Indicates if a record is auto-created by the add-on. For example, a Golden record can be auto-created when we activate the 'Automatically create new golden' option in Matching policy.
Used surrogate fields matching	Indicates that the matching score is calculated by using surrogate field.
Group at once	Grouped: • Indicates that the record is in a group computed by the 'Group at once' operation. Cluster size: • Number of records in the group when the 'Group at once' operation executes.
Ongoing workflow	Identifier: • Technical identifier of the workflow that has been launched for the record. • Undefined if there is no ongoing workflow. Name: • Name of the workflow corresponding to the workflow identifier. • Undefined if there is no ongoing workflow. Timestamp: • Date and time of the workflow creation. • Undefined if there is no ongoing workflow. User • User who originated the workflow. • Undefined if there is no ongoing workflow.
Last process policy code	Code of the last process policy executed on the record.
Last matching policy code	Code of the last matching policy executed on the record.
Last survivorship policy code	Code of the last survivorship policy that has been executed on the record

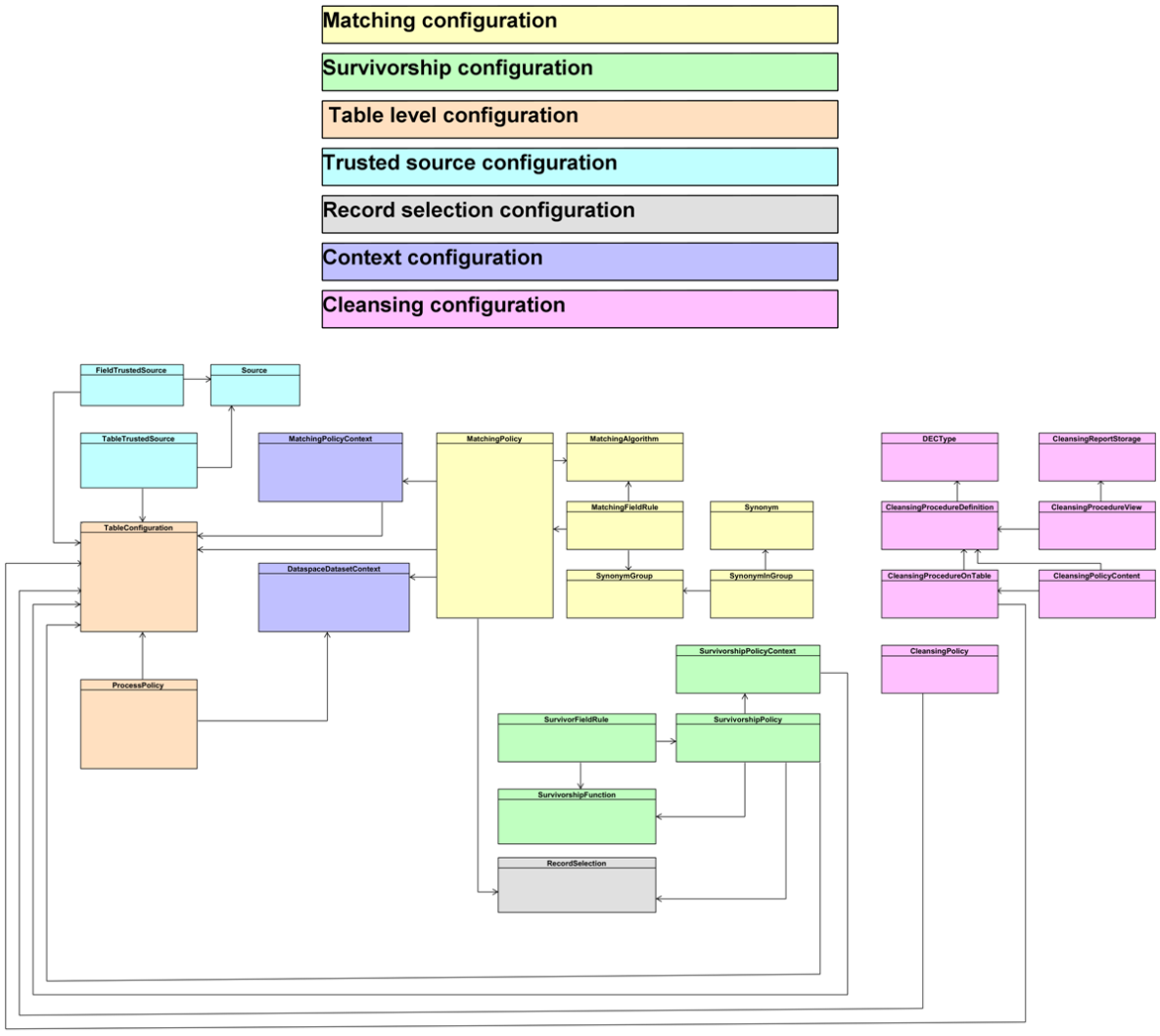
Properties	Definition
Last operation code	In certain situations, the add-on saves an information related to the last operation applied to the record. For example, in case of an automatic merge a special code is used
Date of last operation code	Gives the last date of any operations of match or merge applied to the record
Batch operation code	Use by service 'match at once' to manage life cycle of matching applied on set of records
Cleansing meta-data	
Cleansing state	This value indicates the record's cleansing quality level. The cleansing state can be one of four values: Undefined (null), Clean, To be fixed or Fixed. When a table is enriched with Cleansing metadata and does not run any Cleansing procedures/ operations, all records have an 'Undefined' state. After cleansing procedure execution a record is moved to the 'Clean' state if no defects are found. However, when a record is found to have defects it is moved to the 'To be fixed' state. After running a cleansing operation to fix the defect, the record is changed to a 'Fixed' state.
Cleansing procedure code	Code of the last cleansing procedure applied to the record.
Cleansing operation	Code of last cleansing operation applied to the record.
Execution date	Last execution date of the cleansing procedure or operation on the record.
User	User having executed the last cleaning procedure or operation.
Field	The field affected by the last execution of the cleansing procedure or operation.
Quality defect	During the execution of a cleansing procedure or operation if the record has a defect, then this field is set to 'Yes', otherwise it is set to 'No'.
Merged field logging	Logs data each time the field is merged into the target record.

Table 54: Metadata for matching and cleansing

CHAPTER 19

The EBX® Match and Merge Add-on data model

The following image highlights the add-on's logical data model configuration. This is not the full and updated version but it gives an overview of the data foundation used to configure the matching and cleansing policies.



CHAPTER 20

Matching strategies

This chapter contains the following topics:

20.1 Types of matching strategies

The add-on provides two strategies for finding duplicates: phonetic and distance. An additional configuration allows you to mix these two strategies, which improves the quality of matches.

It is possible to extend these strategies by implementing any other form of matching. The add-on accepts custom matching algorithm configuration.

20.2 Phonetic matching

The phonetic matching relies on how words are pronounced. When pronunciation is similar, the match score is higher.

For example, phonetic matching will provide a high score for the following:

- Billie (pivot)
- Beellie
- Billy

However, for these values, phonetic matching fails to find a match:

- Billie
- Bellie
- Millie

Moreover, the language used influences the matching sensitivity. For instance, depending on whether the language used is English or French, the matching results are different for the following:

- Billie (pivot)
- Beellie - not considered a suspect in French, as it is in English

The language used by the matching process is a property in the 'Table configuration'.

When terms to match are not names but phone numbers, email addresses, codes, etc., phonetic matching is likely not the best strategy. In such cases, using a distance matching strategy is generally preferable.

20.3 Distance matching

Distance matching computes the *distance*, or number of differences, between two terms. When the terms to compare are long (that is, more than 30 characters) or have varying sizes, distance matching may not be suitable.

Distance matching is a highly efficient method of comparing terms such as phone numbers, email addresses and business codes. Moreover, since distance matching is not language-specific, it can be more efficient for multilingual terms.

For example, distance matching provides correct outcomes for the following case:

- marie.haady@gmail.com (pivot),
- parie.haady@gmail.com (one distance)

For the same example, phonetic matching fails to find a match.

20.4 Using a double matching strategy

Deciding which matching strategy to apply depends on the data being compared. It may be helpful to test different strategies before launching the matching process over the scope of the entire database. Even with the most suitable matching strategy, 'false negative' records may occur.

A 'false negative' record is a record that should have been identified as a suspect record by the matching procedure, but was not. This situation is problematic because it marks records as golden even though potential suspect records still exist in the database. To fix this issue, the EBX® Match and Merge Add-on can be configured to apply two levels of matching using different strategies.

- The first level applies the best matching strategy to each field to be matched. For instance, a phonetic matching can be used for a name, or distance matching for a postal code.
- To catch 'false negative' records, a second level matching strategy can be configured using a different matching strategy. This means that all 'negative' scores for the name are recomputed automatically using distance matching. When a record is marked as a suspect by the second level strategy, its score is set as the 'minimum score stewardship + '0.1' by the add-on ('StewardshipMinScore' property in the 'Process Policy' table.).

The second level matching strategy is optional.

20.5 Matching algorithms by strategy

The table below highlights the most popular matching algorithms used by the phonetic and distance matching strategies.

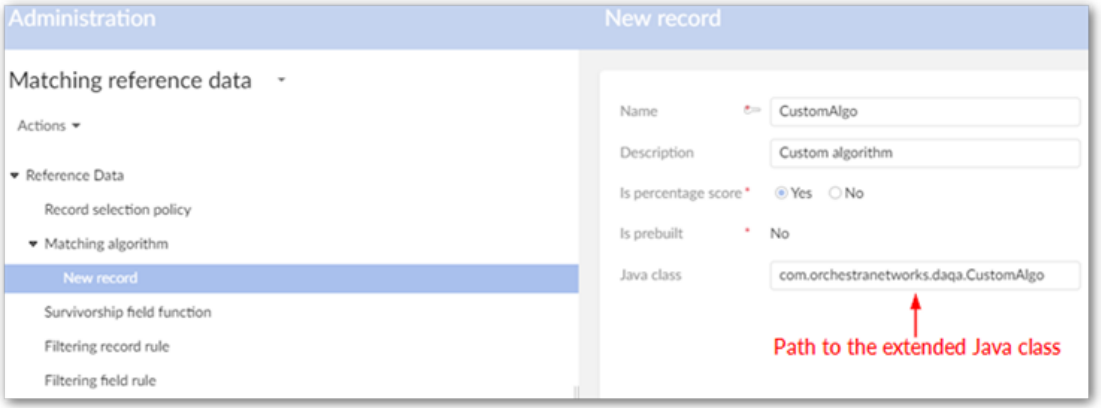
Matching algorithm	Phonetic	Distance	Use context
NY SIIS	X		Better for European and Hispanic name
Double metaphone	X		More generic than Soundex and NY SIIS
Levershtein, Jaro Winker, Fuzzy Full text		X	For short string, not reliant on language, best applied to password, email, business code, phone number, postal code, etc.

Table 55: Matching algorithms by matching strategy

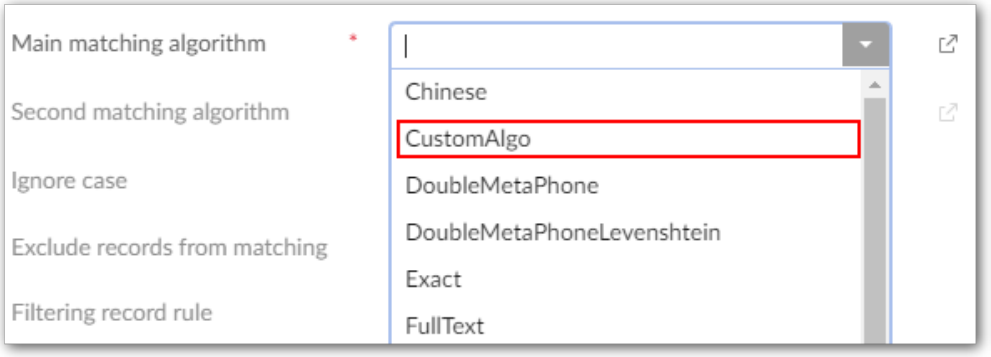
20.6 Implementing a custom matching algorithm

Besides the predefined matching algorithms, you also can create a matching algorithm as your desire. The following section describes in detail step by step on how to implement a custom matching algorithm.

- Extend SearchDistance API
- Export this class to a jar file ,then put this jar file to the same location as 'ebx.jar'.
- In Administration → Matching reference data → Matching algorithm' table, create a new record with path to the extended Java class.

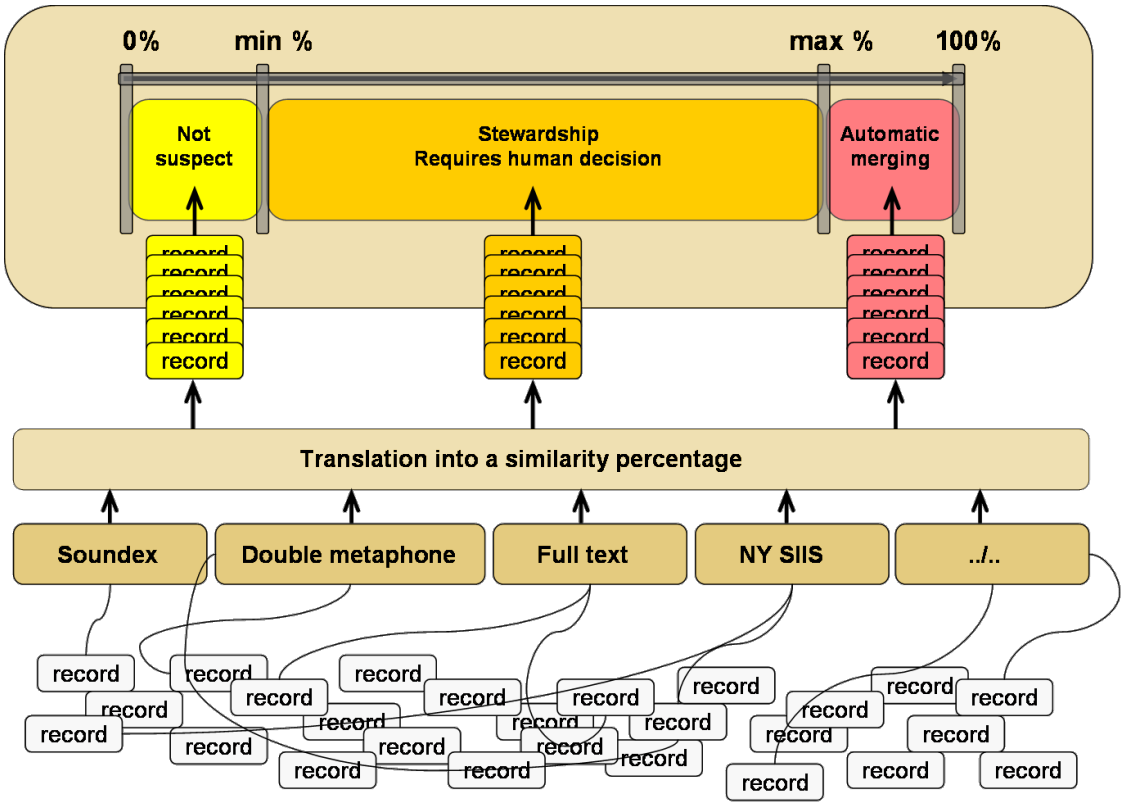


Once you finish, this custom algorithm will be displayed in the list of available algorithms.



20.7 Using matching algorithms with the add-on

The EBX® Match and Merge Add-on manages matching scores formatted as similarity percentages. The highest similarity is 100% (equality).



Depending on a matching process policy's configuration, the similarity percentage is used by the add-on to decide whether the record is a suspect or not. The figure above highlights this decision process.

The minimum and maximum percentages are the 'stewardship min score' and 'stewardship max score' properties, respectively in the 'Process policy' table.

The matching algorithms integrated into the add-on translate their scores into a similarity percentage. When configuring new matching algorithms, the score must always be translated into a similarity score.

CHAPTER 21

Workflow integration

This chapter contains the following topics:

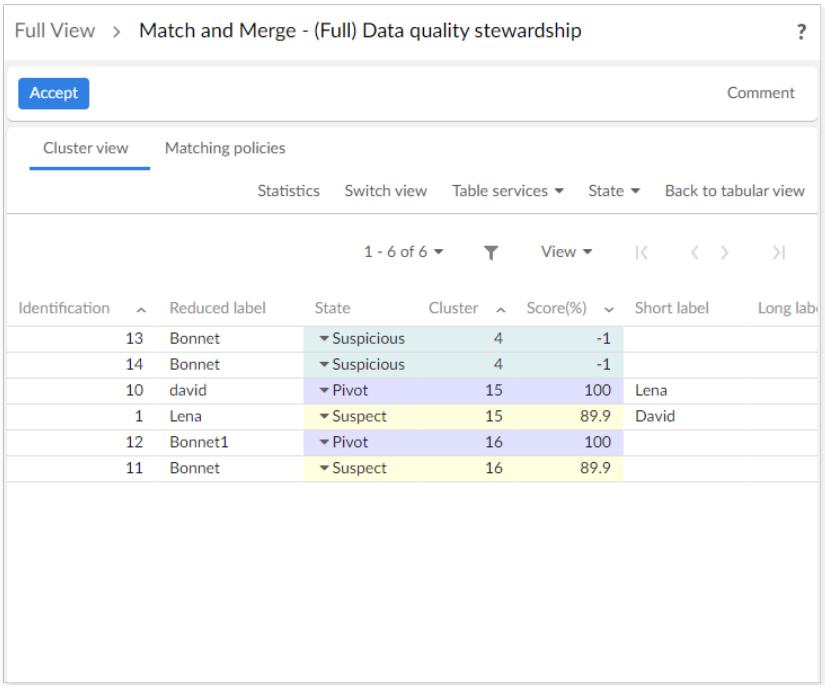
21.1 Workflow users tasks

Beyond configuring the add-on to make it responsible for creating workflows (see process policy), it is possible to design and integrate any bespoke workflow. These workflows are then based on user tasks and scripts provided with the add-on.

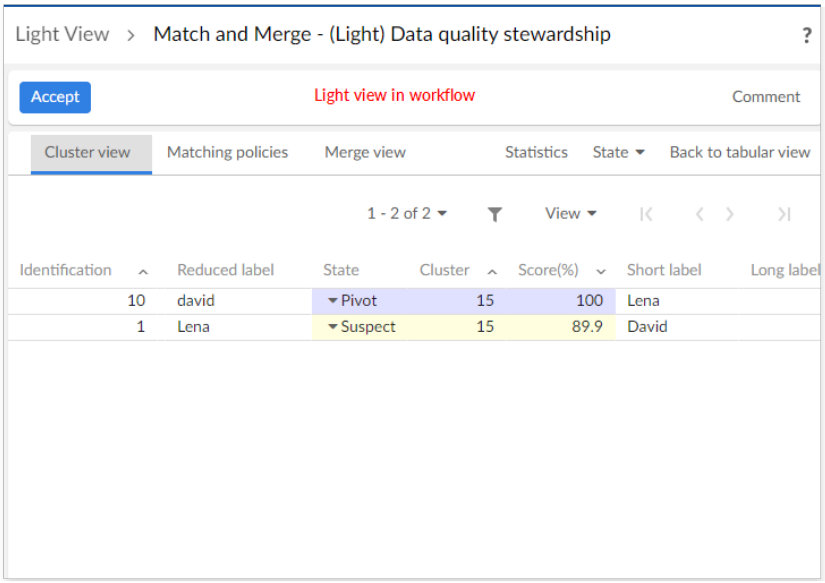
To get samples of workflow please contact our professional service team.

Special notation:	
✔	Please refer to the online documentation in the EBX® workflow modeling domain to get further information about the parameters available to configure the workflow users tasks and scripts.

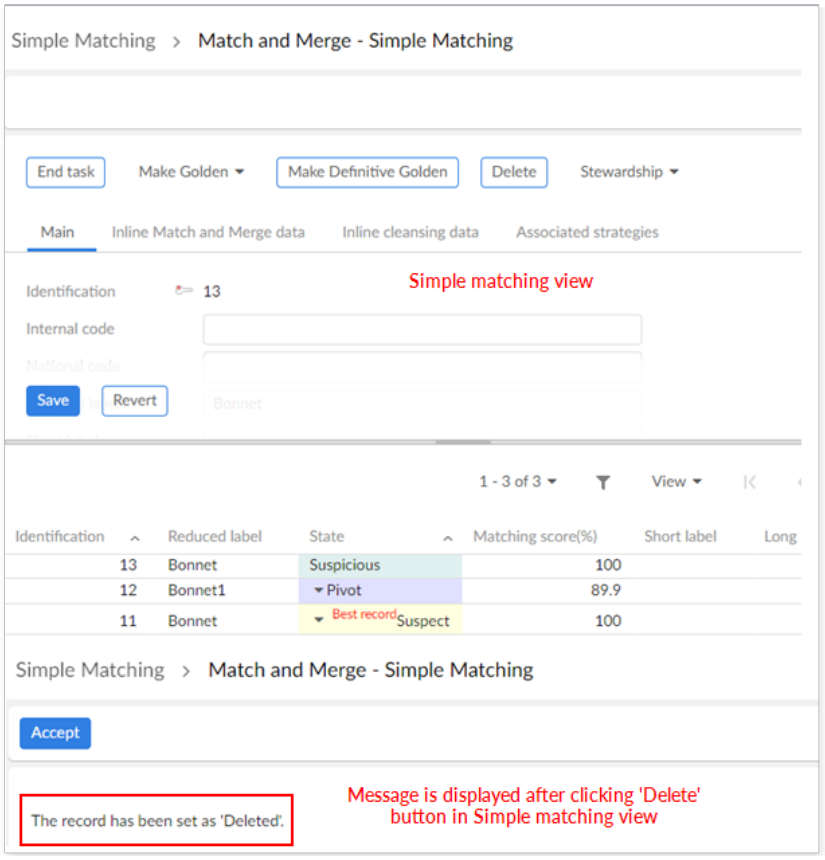
(Full) Data quality stewardship

User task	Overview
(Full) Data quality stewardship task	
	Data context
	Branch (dataspace), instance (dataset), xpath (table), selectClusterId, selectRecord
	Use
	For experimented. Provide all matching operations.

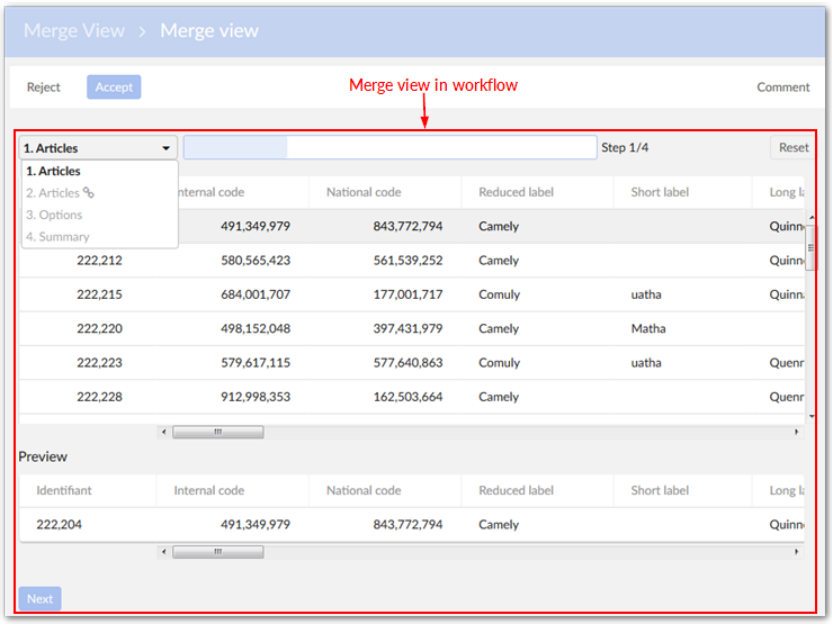
(Light) Data quality stewardship

User task	Overview
(Light) Data quality stewardship task Note that when creating and configuring this type of task, an OR operation applies to the **Cluster to select** and **XPath predicate of record to select** options. If both options are configured, priority is given to the cluster. Also, when you configure the **XPath predicate of record to select** option, all records in the cluster display—not just the selected record.	
	Data context
	Branch (dataspace), instance (dataset), xpath (table), selectClusterId, selectRecord
	Use
	To manage a list of suspect records in a cluster.

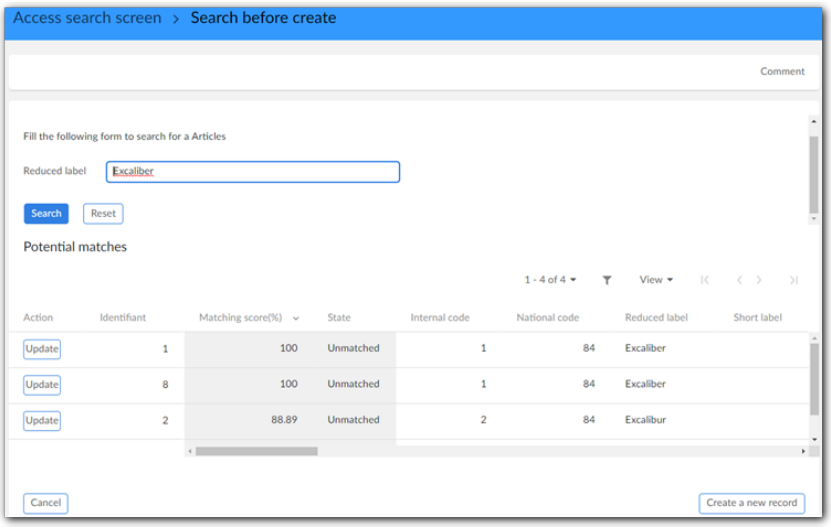
Simple matching

User task	Overview
Simple matching task	
	Data context
	Branch (dataspace), instance (dataset), xpath (table and record ID)
	Use
	To manage a suspicious record against a list of potential suspect records.

Merge view

User task	Overview
Merge task	
	Data context
	Branch (dataspace), instance (dataset), xpath (table and record ID)
	Use
	For users managing a set a suspect records.

Search before create

User task	Overview
Search before create	
	Data context
	Branch (dataspace), instance (dataset), xpath (table)
	Use
	This task allows users to search for a record before potentially duplicating a record.

21.2 **Workflow scripts and conditions**

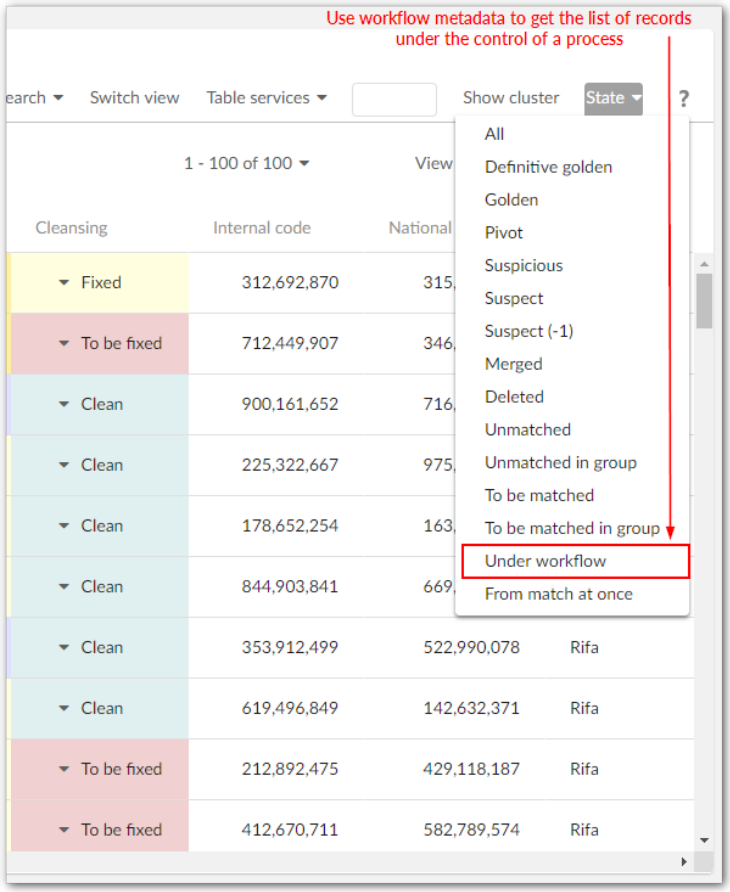
This section describes workflow scripts and conditions bundled with the add-on.

Feed data context with metadata from a record

This script is run to feed a data context with a record's matching metadata: state, score, cluster, and a custom label.

Remove all workflow metadata for the current workflow

By default, the add-on avoids creating duplicate workflows on the same record. In other words, a record cannot have more than one active workflow at a time. To ensure this rule, a matching metadata is used to state if a workflow has been created on a record.



This is the responsibility of the workflow to reset this metadata, and then to allow a new workflow creation. This reinitialization is executed by the 'Remove all workflow metadata for the current workflow' script.

This metadata is also used by the 'Under workflow' service to get the list of all records under the control of a process.

Feed workflow metadata

A workflow not initiated by the add-on can feed the workflow metadata of a record using this script. Therefore the record will be known as 'under workflow' by the add-on. This metadata has to be removed at the end of the workflow using the script described above 'Remove all workflow metadata for the current workflow'.

Change state of a record to suspicious

The record state moves to the 'Suspicious' state.

Simulate the match table operation

The result is a list of all records that match without changing their states. The list is empty if no match.

Check if a record is in a given state

Condition. The result is a boolean.

Align foreign keys

All foreign keys related to the table managed by the add-on are checked. The check includes foreign keys in the table on which the service is executed (in case of self-referencing relationships). The scope of the services includes all related dataspaces and datasets. When a foreign key is linked to a merged record, the add-on automatically updates it with the correct target record in the appropriate dataspace and dataset. For more information on service behavior, see Align foreign keys of all merged records [p 139].

CHAPTER 22

Importing bulk data

When importing data in bulk, it is better to configure a policy that does not execute matching in a transaction scope at the record level. This scope may be too fine-grained, thus leading to performance issues without otherwise benefiting the import process.

To handle this issue, the EBX® Match and Merge Add-on allows you to configure a process policy with 'Is import mode' set to 'Yes' (refer to process policy configuration).

Once a table is associated with this type of policy, matching is not executed when creating or modifying records; only the record's state is modified to 'to be matched' (see detailed rules in the table below).

After the import, the 'Match at once (to be matched)' service manages all records with a 'to be matched' state. At any time, a user can manually execute the 'Match table' service on a record with the 'to be matched' state. At a software level, it is also possible to use the 'Match table' API on a record (refer to java doc).

Context	'Is import mode' = No	'Is import mode' = Yes
New record	New records are matched except 'to be matched', merged and deleted.	Match is not executed. New records are placed in the '002' cluster with the state 'to be matched'.
Update on suspicious, suspect, pivot, golden, unmatched, to be matched	Updates on suspicious, suspect, pivot, unmatched and golden records entail a match. Updates on 'to be matched' do not entail a match.	If 'is import mode insert only' = 'Yes' => the records with suspicious, suspect, pivot and golden states will entail a match. Records with unmatched and to be matched state do not entail a match. If 'is import mode insert only' = 'No' => update records move to the 'to be matched' state in the '002' cluster.
Update on deleted or merged records	Not permitted The import process is stopped at the first attempt to update or delete a merged or deleted record.	
Import merged and deleted record	Permitted Match is not executed.	

Table 56: Importing bulk data rules

Administration Guide

CHAPTER 23

Installation and first configuration

This chapter contains the following topics:

23.1 Environment

- EBX® 5.5.1 or higher.
- The add-on war package must be set up.
- Copy 'ebx-addons.jar' into 'catalina.base/shared/lib' folder (same location with 'ebx.jar').
- When an official EBX® license is used you must get a license key for the add-on.

23.2 Add the add-on metadata type to a table

To be known by the TIBCO EBX® Match and Merge Add-on, a table must be enriched with a matching metadata type that is provided with the add-on.

In the DMA:

- In Configuration => Included Data Models => you must declare the 'ebx-addon-daqa-types.xsd' schema from the 'ebx-addon-daqa' module.
- In ComplexData Types => On your table => add a group with the name 'DaqaMetaData' from the type 'Inline Match and Merge data' stemming from the included schema. The name of this data type 'DaqaMetaData' cannot be modified.
- Move this group just after the primary key of the table.
- Publish and create a dataset if needed.

Special notation:	
✔	See *Uninstallation* section to remove the matching metadata type from a table.

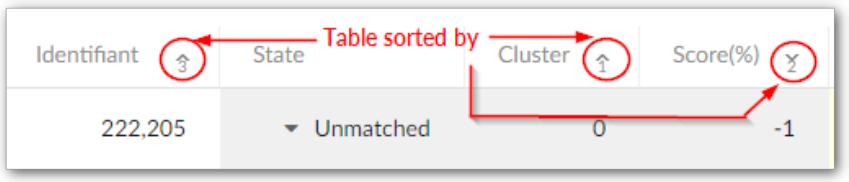
23.3 **Creation of indexes on the table**

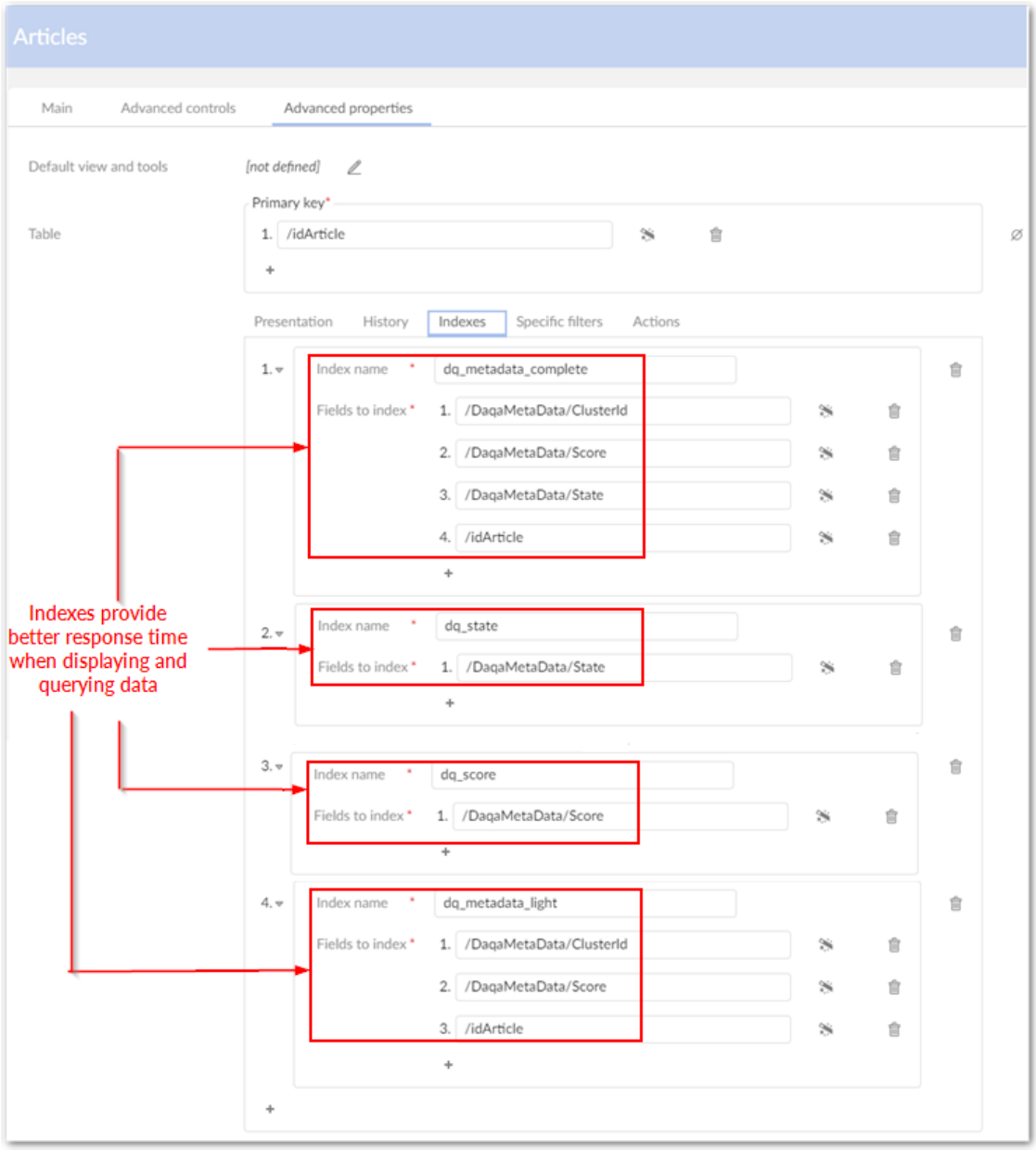
Indexing will optimize the response time when displaying or querying data. The following sections indicate how to create indexes on a table.

Special notation	
✔	You should define indexes on Source field and all Filter by fields.

Basic indexing

Every table under add-on control is sorted by the cluster, the score and then the primary key.





Special notation:	
✔	In order to get better response time when displaying and querying the data, create the following indexes: [ClusterId, Score, State, PK], [ClusterId, Score, PK], [State], [Score].

Indexing for the operation 'Exact match at once'

Regarding the particular case of the 'Exact Match at once' operation, a dedicated index must be created for fields of the matching policy in the exact same order. This step must be done for each matching policy that can be executed in an exact match at once operation.

CHAPTER 24

Using the matching configuration

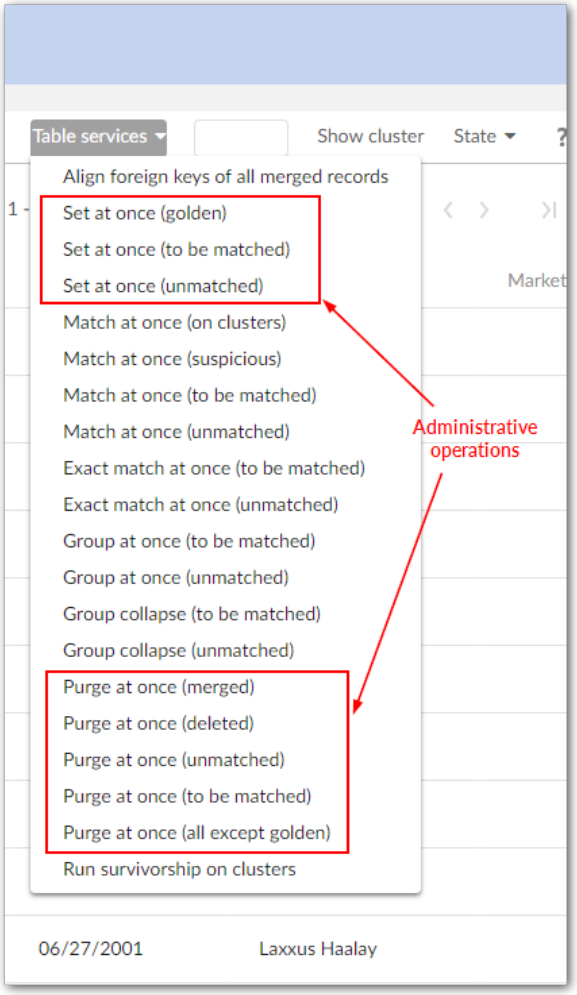
The matching configuration is accessed under 'Matching Configuration' in the Administration area of EBX®. See also 'Concepts' section to understand the 'Process policy' and 'Matching policy' concepts.

CHAPTER 25

Administrative operations

This chapter contains the following topics:

25.1 **Types of administrative operations**



The add-on supplies two types of administrative operations to initialize records in a table, and to purge them.

25.2 Initialization of a table

These operations are not available if the process policy is incorrectly configured. They still remain available when 'On matching Process' is set to 'No'.

Operation	Description
Set at once (unmatched)	Applied on a table directly and for all its content. State values to reset can be selected. 'Unset' value takes undefined record state only. Matching metadata are initialized without any matching execution: • State= unmatched • ClusterID= 000 • Score=-1 • Target=null • Timestamps= current timestamps • Was golden= No • Not suspect with= null
Set at once (to be matched)	Applied on a table directly and for all its content. State values to reset can be selected. 'Unset' value takes undefined record state only. Matching metadata are initialized without any matching execution: • State= to be matched • ClusterID= 002 • Score=-1 • Target=null • Timestamps= current timestamps • Was golden= No • Not suspect with= null
Set at once (golden)	Applied on a table directly and for all its content. State values to reset can be selected. 'Unset' value takes undefined record state only. Matching metadata are initialized without any matching execution: • State= golden • ClusterID= 001 • Score=100 • Target=null • Timestamps= current timestamps • Was golden=No • Not suspect with= null

Table 57: Services applied to initialize a table under the matching

25.3 Purge of a table

These operations are not available if the process policy is incorrectly configured. They still remain available when 'On matching Process' is set to 'No'. When applied to the relational persistence mode, the purge stops if there are referential integrity constraints against records to delete.

Service	Description
Purge at once (merged)	Applied on a table directly and for all its content. Every 'merged' record is physically deleted by EBX®. Every golden record alone in its cluster moves to the '001' cluster.
Purge at once (unmatched)	Applied on a table directly and for all its content. Every 'unmatched' record is physically deleted by EBX®.
Purge at once (to be matched)	Applied on a table directly and for all its content. Every 'to be matched' record is physically deleted by EBX®.
Purge at once (deleted)	Applied on a table directly and for all its content. Every 'deleted' record is physically deleted by EBX®.
Purge at once (all except golden)	Applied on a table directly and for all its content. Every record is physically deleted by EBX® except golden records. Definitive golden records stay in the '003' cluster. All other golden records move to the '001' cluster.

Table 58: Services applied to purge a table under the matching

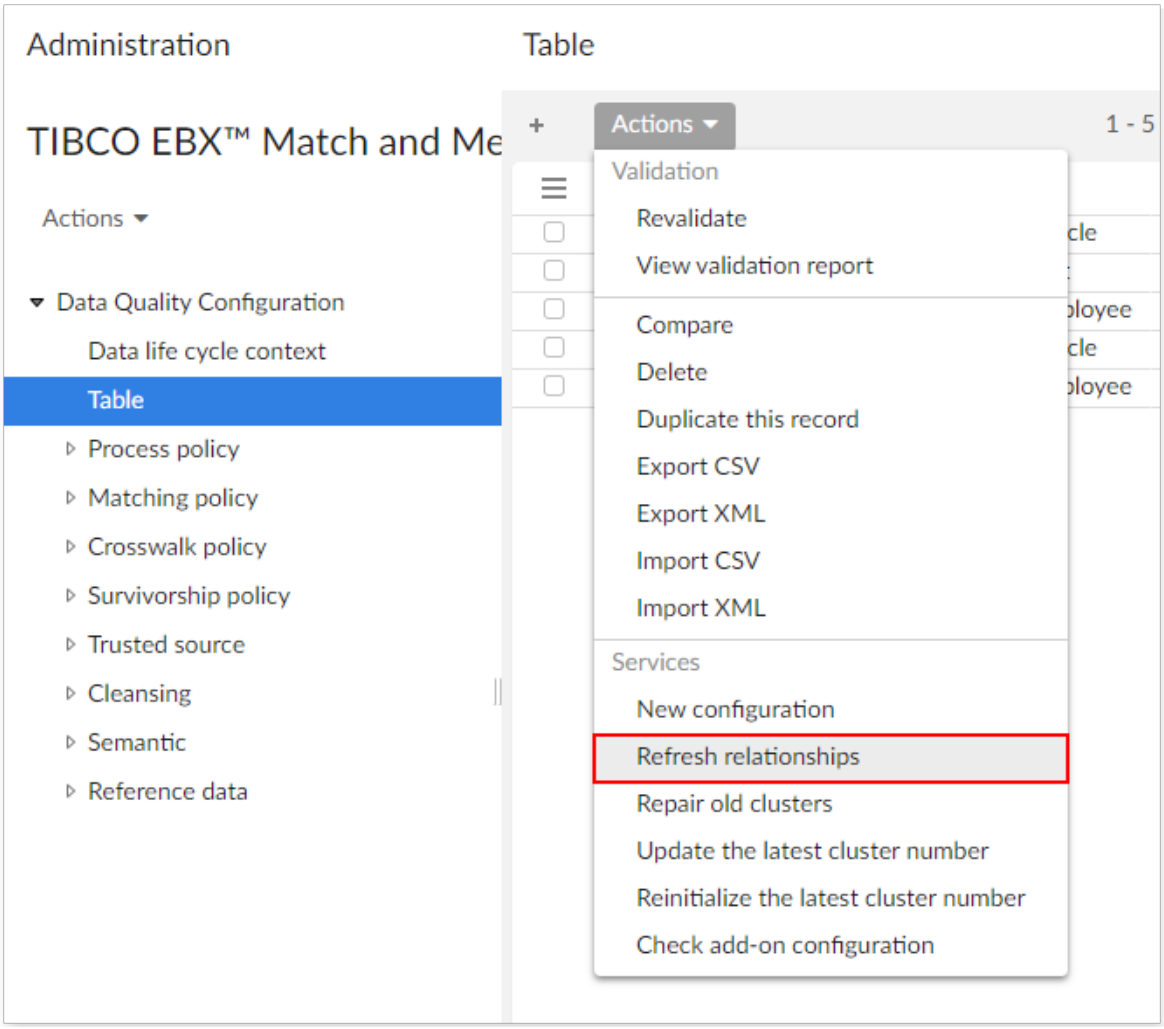
25.4 Refreshing relationships

When you save a **Table** configuration record, the add-on automatically builds and saves relationship data in a **Relationship** table. You can access this information by selecting a **Table** record's **Relationships** tab.

For some actions, the add-on dynamically updates the **Relationship** table's data. For example, when you delete a record, the add-on automatically deletes affected relationships. However, if you make modifications to a data model that impact stored relationship data, you should run the **Refresh relationships** service. Changes that could require a refresh include updates to the model's structure, or import of a previous add-on configuration. Depending on the changes made to the data model, this service may insert or delete records in the **Relationship** table.

> **Note**
>
> Only relationships included in the same data set as the configured table display in the **Relationship** table.

The **Refresh relationships** service can be run from the location shown in the following image:

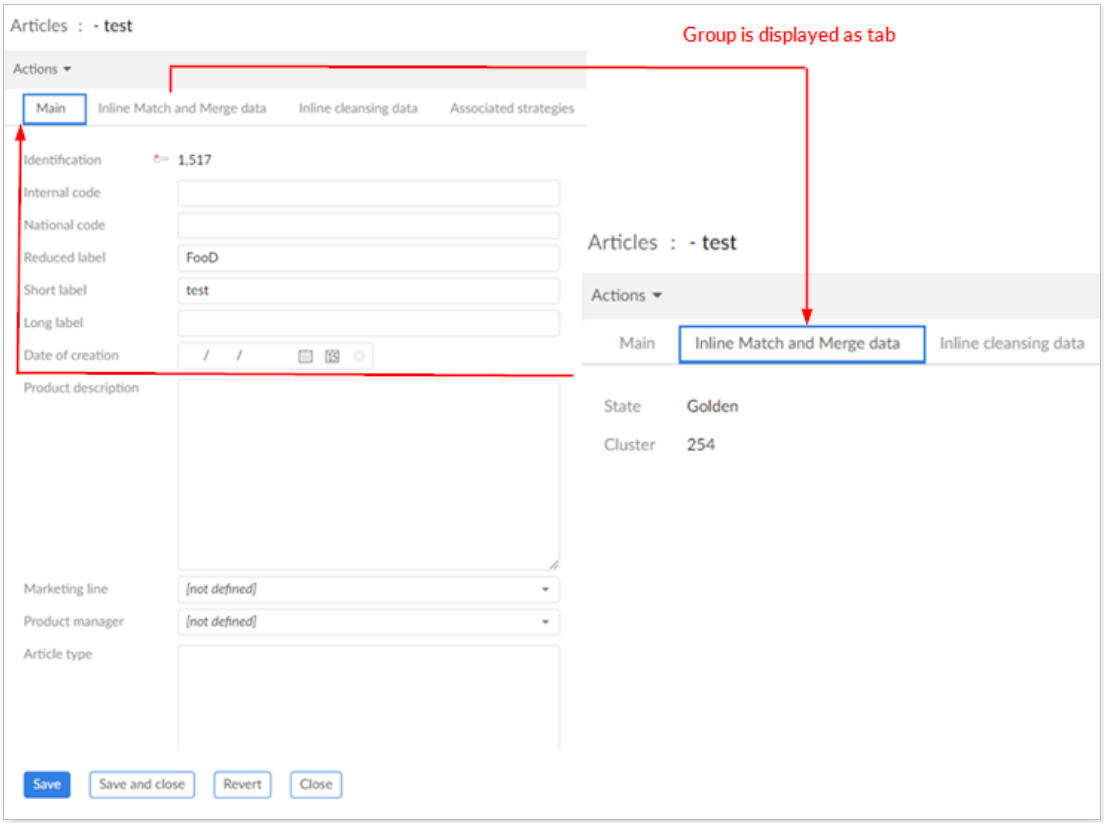


CHAPTER 26

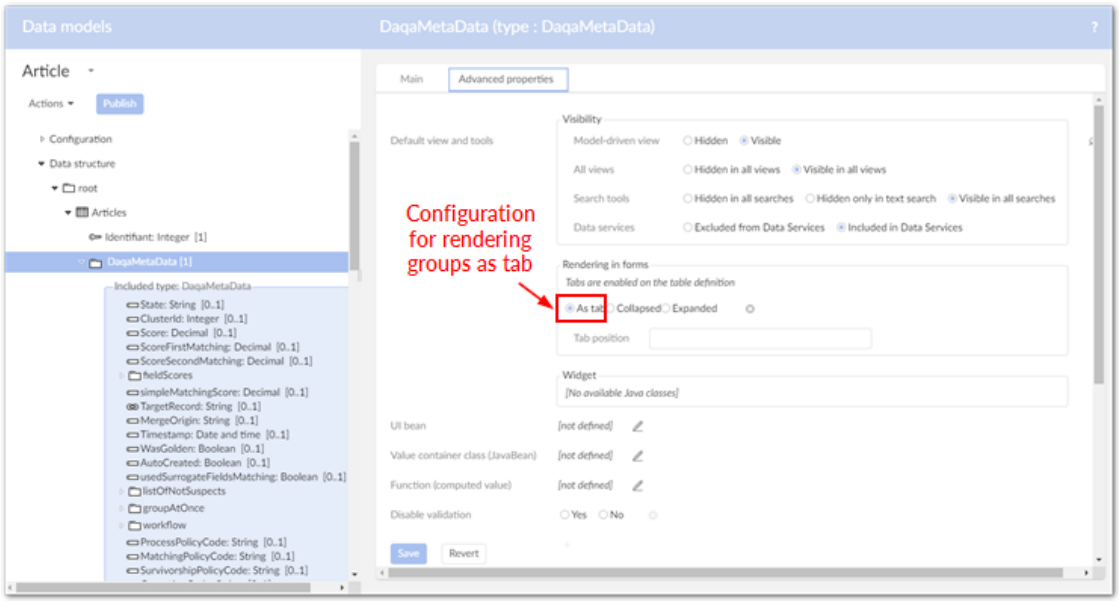
Displaying metadata group as tabs

When EBX® displays the tabular view of a record in a table that includes matching metadata, it may be convenient to display the matching metadata values in a separate tab, as shown in this example:

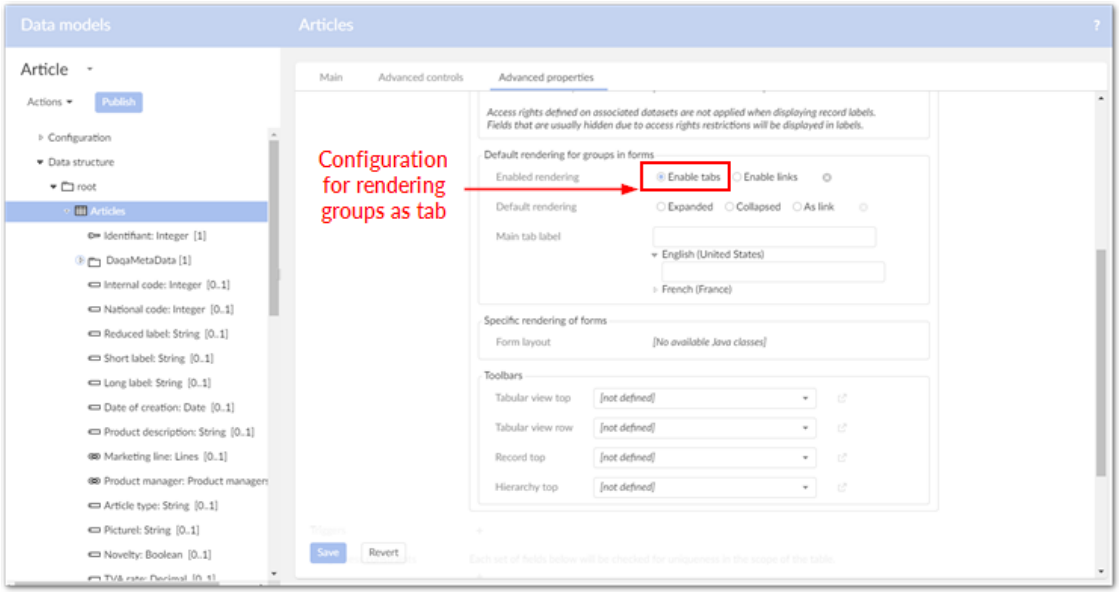
To display these fields in a separate tab, the user interface must be configured as follows:



- Under the 'Advanced properties' of the 'DaqaMetaData' group, select the option 'Tab' under Default view > Rendering.



- In the table's 'Advanced properties', select 'Enable tabs' under Table > Presentation > Default rendering for groups in form > Enabled rendering.



CHAPTER 27

Data initialization

The EBX® Match and Merge Add-on offers several mechanisms to initialize a table under its control. A process policy is then configured to manage the best way to face the initialization.

This chapter contains the following topics:

27.1 Process policy properties

The properties of a process policy that can be used to drive the initialization are as follows (refer to the explanation in this user guide to get further information): 'On matching process', 'Forced golden creation', 'Is import mode' and 'Is import mode insert only'.

27.2 Operations

Two operations are available to initialize the state of records if needed:

- Set at once (unmatched) with option='Only undefined record state'.
- Set at once (to be matched) with option='Only undefined record state'.

27.3 **Examples of some data initialization uses cases**

	Process policy properties						Operation	
Context of use	**On DQ Process**	**On Creation**	**On modification**	**Forced golden creation**	**Is import mode**	**Is import mode insert only**	**Set at once unmatched**	**Set at once to be matched**
Initialization from an external source (CSV). All records are set to unmatched	No	any	any	any	any	any	not used	not used
Existing records in the table before installing the add-on. All records are set to unmatched	any	any	any	any	any	any	used	not used
Initialization from an external source (CSV). All records are set to golden	Yes	yes	yes	yes	any	any	not used	not used
Initialization from an external source (CSV). All records are set 'to be matched'	Yes	yes	yes	no	yes	yes	not used	not used
Existing records in the table before installing the add-on. All records are set to 'to be matched'	any	any	any	any	any	any	not used	used

CHAPTER 28

Uninstallation

By removing the matching metadata type from a table, all matching information is physically removed. The records of the table are still saved. For example, this use case is meaningful:

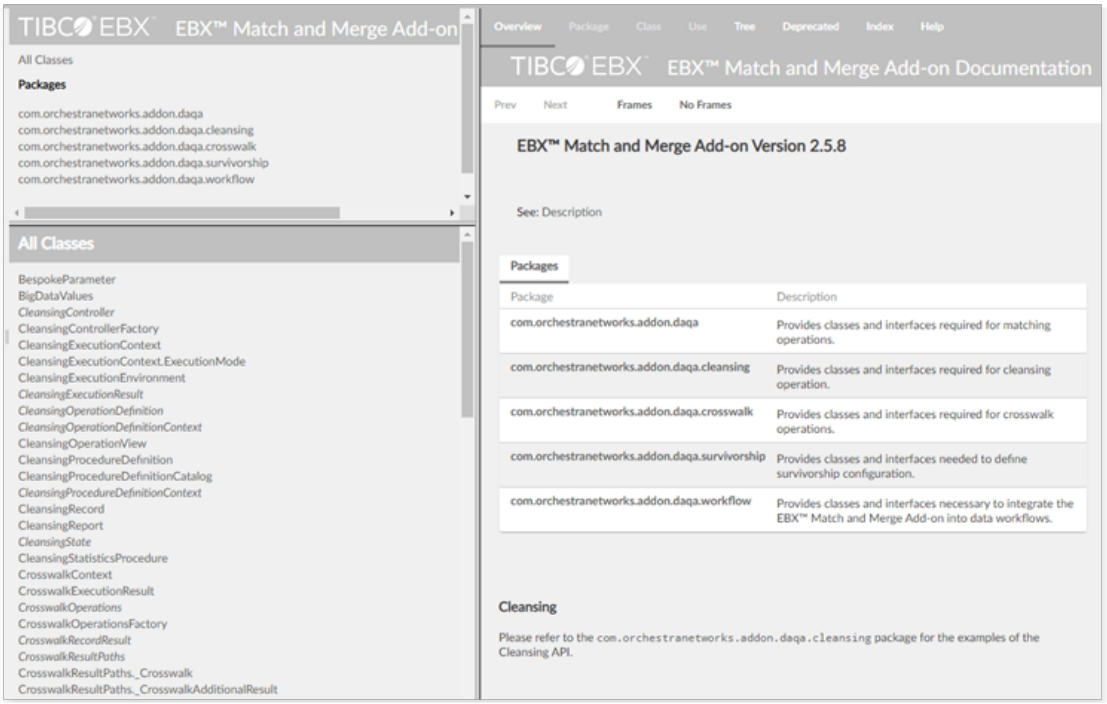
- Add the matching metadata type.
- Cleansing of this table with the EBX® Match and Merge Add-on.
- Purge all records except golden (see purge operation).
- Remove the matching metadata type, then the table is no longer known by the add-on.

Developer Guide

CHAPTER 29

API Introduction

A set of java APIs allows you to integrate the EBX® Match and Merge Add-on to any external systems. Please refer to the add-on java doc via the online help 'Java API' section.



Release Notes

CHAPTER 30

Version 2.5.14

Released: December 2021

This chapter contains the following topics:

30.1 New features

This release contains no new features.

30.2 Changes in Functionality

This release contains no functionality changes.

30.3 Changes to third-party libraries

This release contains the following third-party library updates:

- The Spring Framework was updated to version 5.2.15.
- The jQuery UI library was updated to version 1.13.0.

30.4 Closed issues

[DAQA-3890] Validate data in the **Table/Field trusted source** configuration.

30.5 Known issues

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

- The EBX® session lifetime is limited for security reason, meaning the user is logged out automatically after a certain amount of time passes without activity. In some situations, this can open the login EBX® screen inside a frame of the EBX® Match and Merge Add-on views. To

exit from this type of situation, select any action outside the frame, and then EBX® full login screen will appear. This limitation has no consequence on data integrity or security issues. This is an ergonomics known limitation.

- When the search panel is active on the EBX® tabular view, it will be inherited in the matching views. Then, it is no longer possible to remote it at the level of the matching views. It is mandatory to remove it from the EBX® tabular view.
- All EBX® Match and Merge Add-on services and triggers are deactivated on children data-sets.
- If you want to use old configuration archive files from a previous version, after importing them, you must restart the system.
- Search option on matching views is disabled for searching on foreign key which is configured as matching field with multi hop links.
- Multi occurrence field and foreign key field with multi hop are not supported when executing 'Exact match at once' operation.
- You should avoid matching with inherited fields or value functions because they are not indexed in EBX® and can lead to performance issue.
- Matching is not enabled on D3 slave delivery dataspaces.
- The enumeration fields are not taken into account when matching.
- Only single-occurrence fields at the first level of a multi-occurrence group are considered when matching.
- Match at once parallel does not support 'Data life cycle context' in Matching policy.
- Exact match at once in memory does not support 'Matching policy context' in Matching policy.
- In a surrogate matching, complex fields, foreign key fields, multiple-value fields and enumeration fields cannot be taken as alternative matching fields.
- When changes are made to the configuration while the 'Match at once' service is running, the add-on reloads the index and executes matching with the new configuration.
- When data referenced by foreign keys, or contained in linked tables, is modified during a matching operation, the system automatically clears and reloads the cache.
- If a related table's foreign key is in a list, you cannot run the **Align foreign key of merged record** service on the golden, or pivot record.
- The **Restore from history** service is not available on tables containing the DaqaMetaData group.

This release contains the following known issue related to cleansing functionality: The cleaning procedure 'Foreign key fixing' does not support multi-valued FK. It is not possible to fix a FK that is a part of a primary key.

This release contains the following known issue related to the crosswalk feature: The crosswalk function does not support matching on foreign keys.

CHAPTER 31

All release notes

This chapter contains the following topics:

31.1 Version 2.5.14

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

This release contains the following third-party library updates:

- The Spring Framework was updated to version 5.2.15.
- The jQuery UI library was updated to version 1.13.0.

Closed issues

[DAQA-3890] Validate data in the **Table/Field trusted source** configuration.

Known issues

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

- The EBX® session lifetime is limited for security reason, meaning the user is logged out automatically after a certain amount of time passes without activity. In some situations, this can open the login EBX® screen inside a frame of the EBX® Match and Merge Add-on views. To

exit from this type of situation, select any action outside the frame, and then EBX® full login screen will appear. This limitation has no consequence on data integrity or security issues. This is an ergonomics known limitation.

- When the search panel is active on the EBX® tabular view, it will be inherited in the matching views. Then, it is no longer possible to remote it at the level of the matching views. It is mandatory to remove it from the EBX® tabular view.
- All EBX® Match and Merge Add-on services and triggers are deactivated on children data-sets.
- If you want to use old configuration archive files from a previous version, after importing them, you must restart the system.
- Search option on matching views is disabled for searching on foreign key which is configured as matching field with multi hop links.
- Multi occurrence field and foreign key field with multi hop are not supported when executing 'Exact match at once' operation.
- You should avoid matching with inherited fields or value functions because they are not indexed in EBX® and can lead to performance issue.
- Matching is not enabled on D3 slave delivery dataspaces.
- The enumeration fields are not taken into account when matching.
- Only single-occurrence fields at the first level of a multi-occurrence group are considered when matching.
- Match at once parallel does not support 'Data life cycle context' in Matching policy.
- Exact match at once in memory does not support 'Matching policy context' in Matching policy.
- In a surrogate matching, complex fields, foreign key fields, multiple-value fields and enumeration fields cannot be taken as alternative matching fields.
- When changes are made to the configuration while the 'Match at once' service is running, the add-on reloads the index and executes matching with the new configuration.
- When data referenced by foreign keys, or contained in linked tables, is modified during a matching operation, the system automatically clears and reloads the cache.
- If a related table's foreign key is in a list, you cannot run the **Align foreign key of merged record** service on the golden, or pivot record.
- The **Restore from history** service is not available on tables containing the DaqaMetaData group.

This release contains the following known issue related to cleansing functionality: The cleaning procedure 'Foreign key fixing' does not support multi-valued FK. It is not possible to fix a FK that is a part of a primary key.

This release contains the following known issue related to the crosswalk feature: The crosswalk function does not support matching on foreign keys.

31.2 Version 2.5.13

Released: October 2021

New features

This release contains the following new features:

- The **Align foreign key of all merged records** service allows you to align tables configured in the **Relationship** table.
- You can now configure a permission for the **Modify record** service.
- A workflow is automatically launched when a Suspicious record is modified.
- The latest timestamp is now applied for most trusted source records.
- You can now specify a **Matching policy** when using the `MatchingOperations.matchSelection()` API.
- The **Search before create** service now redirects to the initial screen, where it starts.
- The API's new `MatchingOperations.disableMatchingPolicy()` and `MatchingOperations.enableMatchingPolicy()` methods allow you to disable/enable a list of Matching policies.
- You can now configure to display the **Toolbars** button on the **Data quality stewardship** view.

Changes in Functionality

This release contains no functionality changes.

Changes to third-party libraries

This release contains no updates to third-party libraries.

Closed issues

This release contains the following bug fixes:

- **[DAQA-3863]** An impacted cluster is not fixed when a record becomes Suspicious.
- **[DAQA-3873]** There are two Pivot records in one cluster after running the **Match table** service.
- **[DAQA-3879]** A `Performance` issue occurs when executing a **matchBestCluster()** operation.
- **[DAQA-3880]** A `Deadlock` occurs when executing a **simulateMatchTable** operation while a cache update is in progress.

Known issues

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

- The EBX® session lifetime is limited for security reason, meaning the user is logged out automatically after a certain amount of time passes without activity. In some situations, this can open the login EBX® screen inside a frame of the EBX® Match and Merge Add-on views. To exit from this type of situation, select any action outside the frame, and then EBX® full login screen will appear. This limitation has no consequence on data integrity or security issues. This is an ergonomics known limitation.
- When the search panel is active on the EBX® tabular view, it will be inherited in the matching views. Then, it is no longer possible to remote it at the level of the matching views. It is mandatory to remove it from the EBX® tabular view.
- All EBX® Match and Merge Add-on services and triggers are deactivated on children data-sets.
- If you want to use old configuration archive files from a previous version, after importing them, you must restart the system.

- Search option on matching views is disabled for searching on foreign key which is configured as matching field with multi hop links.
- Multi occurrence field and foreign key field with multi hop are not supported when executing 'Exact match at once' operation.
- You should avoid matching with inherited fields or value functions because they are not indexed in EBX® and can lead to performance issue.
- Matching is not enabled on D3 slave delivery dataspaces.
- The enumeration fields are not taken into account when matching.
- Only single-occurrence fields at the first level of a multi-occurrence group are considered when matching.
- Match at once parallel does not support 'Data life cycle context' in Matching policy.
- Exact match at once in memory does not support 'Matching policy context' in Matching policy.
- In a surrogate matching, complex fields, foreign key fields, multiple-value fields and enumeration fields cannot be taken as alternative matching fields.
- When changes are made to the configuration while the 'Match at once' service is running, the add-on reloads the index and executes matching with the new configuration.
- When data referenced by foreign keys, or contained in linked tables, is modified during a matching operation, the system automatically clears and reloads the cache.
- If a related table's foreign key is in a list, you cannot run the **Align foreign key of merged record** service on the golden, or pivot record.
- The **Restore from history** service is not available on tables containing the DaqaMetaData group.

This release contains the following known issue related to cleansing functionality: The cleaning procedure 'Foreign key fixing' does not support multi-valued FK. It is not possible to fix a FK that is a part of a primary key.

This release contains the following known issue related to the crosswalk feature: The crosswalk function does not support matching on foreign keys.

31.3 Version 2.5.12

Released: August 2021

New features

This release contains the following new features:

- The API's new `MatchingOperations.checkSimilarity()` method allows you to check similarity between two records.
- A new **Match best cluster** service allows you to put a record in the best cluster.
- You can no longer configure the **Is prebuilt** property when creating a new custom algorithm.
- The **Matching** operation is now improved to protect existing clusters.
- The **Exact match** without the **In memory** mode operation is now improved to protect existing clusters.

Changes in Functionality

This release contains no functionality changes.

Changes to third-party libraries

This release contains the following third-party updates:

- Apache Commons Compress to version 1.21.

Closed issues

This release contains the following bug fixes:

- **[DAQA-3838]** An `ArrayIndexOutOfBoundsException` occurs when executing a **Matching** operation when the **Matching policy** has a matching policy context of another table.
- **[DAQA-3840]** A performance issue occurs when executing the **Align foreign key** service.
- **[DAQA-3842]** A performance issue occurs when executing the **Matching** service on a multi-valued foreign matching field.

31.4 **Version 2.5.11**

Released: June 2021

New features

This release contains the following new features:

- The **Auto create new golden** property is now applied when fixing an invalid cluster.
- The API's new `MatchingOperations.matchBestCluster()` method allows you to put a record in the best cluster.

Library updates

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

Bug fixes

This release contains the following bug fixes:

- **[DAQA-3518]** An incorrect survivorship result is returned when the **Field trusted sources** property does not contain the source value of the survivor record.
- **[DAQA-3817]** The error message *Cannot access the service* is displayed in the **Merge view** when executing merge manually with the **Automatically create new golden** option activated.
- **[DAQA-3819]** An incorrect *Merged field logging* message is returned when executing the **Merge record** service.
- **[DAQA-3831]** An incorrect result is returned when executing a **Simulate match table** operation.

31.5 **Version 2.5.10**

Released: May 2021

New features

This release contains the following new features:

- You can now configure the **Number of processed records to update status** property to decide when the **Matching** process will update the status of monitoring.
- To ensure cluster retention, a record is moved to an existing cluster that has a Golden record when the record and the Golden record match.

Bug fixes

This release contains the following bug fixes:

- **[DAQA-3786]** Multi-occurrence data is not merged into the Golden when executing merge data manually from the **Merge view**.
- **[DAQA-3787]** An incorrect result is returned when modifying a record when the **No match records when same source** property is activated.
- **[DAQA-3789]** A Pivot does not become Golden when the cluster has only Merged and Deleted records.
- **[DAQA-3790]** An exception is improperly handled when executing a **Survivorship** operation.
- **[DAQA-3796]** An incorrect result is returned when executing a **Match table** operation when the **On not suspect with** property is activated.
- **[DAQA-3799]** An incorrect result is returned when executing a **Match table** operation on a broken foreign key field.

31.6 **Version 2.5.9**

Released: March 2021

Updates

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

31.7 **Version 2.5.8**

Released: February 2021

New features

This release contains the following new features:

- The add-on was renamed to TIBCO EBX® Match and Merge Add-on.
- Clusters are now fixed after executing a **Run match** operation or the `MatchingOperations.matchSelection()` method in the API.
- Merged records will follow their target record if the record is moved out of a cluster.

31.8 **Version 2.5.7**

Released: January 2021

New features and upgrades

This release contains the following new features and upgrades:

- The progress bar was enriched to include additional information when executing the **Set state** and the **Set at once** operations.
- The **Back to tabular view** button is always displayed when accessing into the **(Light) Data quality stewardship** from a tabular view.
- The **Matching** operation was optimized to reduce processing time when using the **Levenshtein** algorithm with long string data.
- The following libraries were updated:
  - 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 contains the following bug fixes:

- **[DAQA-3730]** An incorrect result is returned when executing a **Matching** operation when the **Filtering field rule** and the **Handle null value matching** properties are configured.
- **[DAQA-3732]** A Null pointer exception occurs when executing the **Check add-on configuration** service.
- **[DAQA-3758]** An incorrect matching score is returned when executing the **Simulate matching** operation on a composite foreign matching field and the **Exact** algorithm.
- **[DAQA-3761]** The **MatchingOperations.matchSelection()** is not applicable for the **Golden** record.
- **[DAQA-3763]** A hidden record is still displayed when accessing to the **Merge** view from the **Matching** view.

31.9 **Release Note 2.5.6**

Release Date: October 20, 2020

New features

This release contains the following new features:

- It is now possible to execute the **Align foreign key of all merged records** operation in a specific dataspace or dataset.
- The API's new `MatchingOperations.alignForeignKeysInDataspace()` method allows you to execute the **Align foreign key of all merged records** service in a specific dataspace.
- The **Table services** button is always displayed when access the **(Full) Data quality stewardship** through a workflow.

Bug fixes

This release contains the following bug fixes:

- **[DAQA-3710]** An incorrect result is returned when executing the **Matching** operation when using matching through relation and the **Handle null value matching** is configured.
- **[DAQA-3725]** The status screen is flickering continuously at the end of the process when executing the **Set state** service.

31.10 Release Note 2.5.5

Release Date: September 18, 2020

New features and enhancements

This release contains the following new features and enhancements:

- Table services have been grouped into a collapsed list.
- The **On not suspect with** property will be checked when removing a record from the **Cluster**. This will be the case despite the **Suspect record retention** property setting.
- The **Trusted source list** configuration has been enhanced with the ability to move up, down, left and right.
- It is now possible to execute the **Unmerge** operation using history from a parent dataspace.
- It is now possible to change the Pivot record when executing the **Check similarity** service.
- The **Matching** operation has been optimized to use less memory when using Exact and Fuzzyfulltext algorithms.
- The add-on has been updated to support the OpenJDK8 and OpenJDK11 libraries.
- Libraries were updated to fix some potential issues.
- The **MatchingOperations.matchSelection()** API has been enhanced to inject a list of **Matching** states.
- The **MatchingOperations.alignForeignKeys()** API has been enhanced to align records in same dataset.

Bug fixes

This release contains the following bug fixes:

- **[DAQA-3649]** An incorrect result is returned when executing the **Match table** operation on a record having the Pivot in the *Not suspect with* list and the **On not suspect** property is activated.
- **[DAQA-3650]** An incorrect number of records in group is displayed when executing the **Match at once** by groups service.
- **[DAQA-3651]** An incorrect result is returned when creating a new record and the **On not suspect with** property is activated.
- **[DAQA-3652]** An IndexOutOfBounds Exception occurs when executing the **Match table** operation when the **Matching field** and the *Filter by* properties are configured the same field.
- **[DAQA-3653]** An incorrect result is returned when creating a new record and the **On not suspect with** property is activated.
- **[DAQA-3655]** The **Final score** is displayed incorrectly in the **Check similarity** screen with one matching field.

- **[DAQA-3656]** An incorrect result is returned when executing the **Records linking analysis** service.
- **[DAQA-3660]** An incorrect result is returned when executing the **Match table** operation with FuzzyFullText algorithm.
- **[DAQA-3661]** Merged records were not included in the **Survivorship** progress when executing the **Exact match at once** service.
- **[DAQA-3706]** An incorrect condition can be configured for the **Condition for field value survivorship** property.

31.11 Release Note 2.5.4

Release Date: June 23, 2020

New features and enhancements

This release contains the following new features and enhancements:

- The **Check similarity** service allows you to compare the similarity of two records in the same table. To test, you choose one of the matching policies configured for the table. The policy specifies which fields to compare and the algorithms to use. You can look at the resulting scores to get a preview of how the add-on would handle these records during a matching operation with the selected policy. For additional documentation, see Testing a matching policy [p 78].
- The **Search before create** service no longer re-displays the search screen when accessing from a completed workflow.
- You can now display the **Table services** button when accessing the **(Full) Data quality stewardship** through a workflow by configuring the **Display 'Table services' button** property on the workflow.
- The API's new `matchSelection()` method allows you to execute matching on a specific selection against the target states.
- You can now configure the **Russian** and **FuzzyRussian** algorithms to execute matching on a Russian character set.
- When a record is put in the deleted state, merged records that target the deleted record are automatically updated with new target records.
- The **Repair old clusters** service aligns merged records, those that target a deleted record, with a new pivot or golden. This allows you to repair defective clusters from older versions of the add-on.

Bug fixes

This release contains the following bug fixes:

- **[DAQA-3591]** An incorrect result is returned when executing the **Match table** operation on a record having a null matching field and the **Funneling matching** property is activated.
- **[DAQA-3594]** An incorrect result is returned when executing the **Match table** operation when the **Funneling matching** property is activated and the **Filter by** option is configured.
- **[DAQA-3597]** An incorrect result is returned when executing the **Exact match at once** service with the **Exclude records from matching** property configured.

- **[DAQA-3598]** An incorrect result is returned when executing the **Match at once** full mode service with the **Exclude records from matching** property configured.
- **[DAQA-3606]** All groups are not listed when executing the **Match at once** by groups service.
- **[DAQA-3609]** An incorrect result is returned when executing the **Match at once** full mode service while having a large of number records in excluded records.
- **[DAQA-3615]** A Null pointer exception occurs when attempting to merge two unmatched records in the **Merge view** and the **Automatically create new golden** property is activated.
- **[DAQA-3630]** A Null pointer exception occurs when attempting to merge two auto-created golden records in the **Merge view** and the **Customize source value for new golden** property is activated and the **Source field** is not defined.
- **[DAQA-3634]** The **batchOperationCode** of all records are cleansed after executing the **MatchingOperations.executeSurvivorshipOnClusters()** API.
- **[DAQA-3635]** All records are displayed in the **Simple matching view** when a view configured as the default view is applied.

31.12 Release Note 2.5.3

Release Date: April 20, 2020

New features and enhancements

This release contains the following new features and enhancements:

- The behavior of the **Align foreign key of merged records** service has been improved in terms of precision.
- The new **Base record** attribute has been added into the **DaqaMetaData** group to store the auto-created golden when it is created.
- The display of error messages in the **Merge view** has been arranged to be more consistent.
- The **Merged by** attribute has been updated to be more user-friendly.
- When executing the **Search before create** service, the add-on now checks for differences between the parent and child dataspaces. If it does not find any differences, it only indexes the parent dataspace.
- The **Survivorship** policy is now applied when executing the **AutoCreateNewGolden** operation on a single golden.
- It is now possible to cancel the **Run match** operation when it is in progress.

Bug fixes

This release contains the following bug fixes:

- **[DAQA-3522]** A Null pointer exception occurs when executing manually a merge in the **Merge view**.
- **[DAQA-3523]** It takes too long to manually merge two records in the **Merge view**.
- **[DAQA-3525]** The **Last survivorship policy code** is not updated when executing the **Automatic merge** operation.

- **[DAQA-3526]** The **Last survivorship policy code** is updated incorrectly when executing the **Merge manually** operation.
- **[DAQA-3527]** The **Merged field logging** is updated incorrectly when executing the **Merge manually** operation.
- **[DAQA-3533]** An incorrect result is returned when executing the **Match at once** operation with full mode and the **Auto create new golden for single golden** option is activated.
- **[DAQA-3562]** The **Handle null value matching** option is not taken into account when executing the **Matching** option with the **FuzzyFullText** algorithm configured.
- **[DAQA-3563]** The **Survivorship** operation merges data from the record even when its score is lower than the **Stewardship max score**.
- **[DAQA-3574]** A golden record is not created when executing the **Move to a new cluster** operation with the **Auto create new golden for single golden** activated.
- **[DAQA-3575]** An incorrect result is returned when executing the **Match at once** operation when the **Exclude records from matching** property is configured.

31.13 Release Note 2.5.2

Release Date: January 15, 2020

New features and enhancements

This release contains the following new features and enhancements:

- The add-on now provides an alternative option to import archived configuration settings. When importing an archive to update an existing, or migrate to a new environment, this option does not overwrite latest cluster ID values. You can access this functionality via the new **Import configuration** service. For information on migrating between environments and using the new service, see the *Migrating configuration settings* section in the *User Guide*.
- The **Output** variable for a record in the EBX® Match and Merge Add-on script and user tasks is now the returned record's absolute location path.
- The EBX® Match and Merge Add-on now does not check a null matching field before executing a **Matching** operation.
- The **Matching** operation has been optimized to reduce processing time.
- You can now configure the **Handle no relationships matching** property to ignore *Relationship matching* when executing matching on a record without a relationship.

Bug fixes

This release contains the following bug fixes:

- **[DAQA-3448]** No results are returned when executing **Search before create** on a default label foreign key.
- **[DAQA-3451]** The wrong button is displayed on the record form when accessing the **Search before create** service from a workflow.
- **[DAQA-3464]** An error message is displayed when configuring the **Crosswalk fields** property.
- **[DAQA-3465]** The **Display relation records** and **Display merge record** services are not displayed when accessing the **Matching** view.

- **[DAQA-3466]** A blank page is displayed when running the **Search before create** wizard with read-only permission.
- **[DAQA-3467]** An incorrect result is returned when executing the **Match table** operation with **Field to exclude records from match** configured as a Date or time field.
- **[DAQA-3502]** The **On workflow by state** property is not taken into account when creating a new golden record.

31.14 Release Note 2.5.1

Release Date: December 10, 2019

Bug fixes

[DAQA-3460] A `NullPointerException` occurs when starting the repository.

31.15 Release Note 2.5.0

Release Date: November 8, 2019

New features and enhancements

This release contains the following new features:

- Add-on configuration can now be completed using the new configuration wizard. Administrators can access the wizard using the **New configuration** service.
- A **User interface** tab has been added under *Data Quality Configuration* > *Table*. From this tab, it is now possible to customize the data quality stewardship, simple matching, and merge views. From this tab, the states filter can also be hidden by configuring the **State(s) to filter from selection** property.
- You can now configure a **Korean** algorithm to execute matching on a Korean character set.
- It is now possible to set the matching state for selected records using the **Set state** service located in a table's **Actions** menu.
- The **Record XPath** property in EBX® Match and Merge Add-on script and user tasks can be configured as either a relative or an absolute location path of an existing record.
- The **EBX® record form** view is now applied when creating a new record with the **Search before create** service.
- It is now possible to use some add-on features with a relational data model.
- You can now configure a constant value for the **Survivorship field** by using the **Constant survivorship fuction**.

The following list describes updated behavior in this release:

- The **Force orphans golden** option will set all records with no identified suspect into single golden.
- It is now possible to match through a relationship table across dataspaces and datasets.
- The **Fix relation records** is now applied to fix related records in a relationship table across dataspaces and datasets.

- If the link is broken, the raw value is used to match when executing the **matching** operation on a foreign key field.
- The **Matching policy** with a context is now prioritized when executing the **Match at once** operation.
- The bottom part of the **Simple matching** view will be reloaded when updating a suspicious record.
- The **DoubleMetaPhone** and the **Soundex** algorithms have been enhanced to improve performance.
- The **Most trusted source** configuration has been simplified.
- When a merged record is modified, the add-on executes the **Match table** operation on the merged record after executing on its target.
- When executing a matching operation on a Golden or Pivot record, its merged records are not included in the operation regardless of the **Merged records is recycled** property setting.
- The target of a merged record is only updated if its target record is not in the same cluster.
- When a merged record is modified on a non-matching field, the add-on only executes the **Merge** operation.
- When accessing the **Simple matching** view after creating a new record, if executing the **Make Golden**, **Make definitive golden** and **Delete** operations the add-on will navigate to the **EBX® record form** view.

Bug fixes

This release contains the following bug fixes:

- **[DAQA-713]** The value of **Latest cluster number** is increased even though no new cluster is created.
- **[DAQA-3015]** A JavaScript error occurs when clicking on the **Selector** pop-up of a foreign key matching field in the **Search before create** screen.
- **[DAQA-3089]** The **Main** tab dissapears when accessing the **Inline cleansing data** under the metadata tab.
- **[DAQA-3117]** An unexpected error is displayed when executing the **Add into cluster** operation on a record in a read-only dataspace.
- **[DAQA-3161]** A `NullPointerException` occurs when transferring data through the **TIBCO EBX® Data Exchange Add-on**.
- **[DAQA-3211]** The auto-created golden state becomes deleted when moving all merged records outside.
- **[DAQA-3265]** The display layout breaks in the **Process policy** screen when viewing the screen with 4K resolution.
- **[DAQA-3269]** A **Snapshot** property does not display correctly when configuring the **Search before create** user task in a workflow.
- **[DAQA-3275]** The **Relationship** configuration is deleted when accessing the **Merge** view.
- **[DAQA-3276]** An incorrect score is returned when executing the **Match table** operation on a multi-valued field.

- **[DAQA-3281]** An orphan suspect occurs when executing the **Run stewardship on defined pivot** service.
- **[DAQA-3334]** Incorrect behavior occurs when adding more input parameters in a **Cleansing procedure** configuration.
- **[DAQA-3344]** An error occurs when updating a record in the **Search before create** workflow.
- **[DAQA-3380]** The **Matching** cache is reloaded after the *loadCache* API is called.
- **[DAQA-3408]** The **Survivorship** is not applied for an auto-created single Golden when executing the **Exact match at once** operation.

31.16 Release Note 2.4.0

Release Date: June 20, 2019

New features

This release contains the following new features:

- The new **Run match** service—available from a table's **Actions** menu—has been added to the matching services catalog. This service allows you to select one or more records to include in a matching operation. If you run the service without making any selections, matching runs on the entire table.
- Administrators can now enable the **Search before create** service. This service helps users avoid creating duplicate records by allowing them to search existing records prior to creating a new record.
- You can now configure matching algorithm input parameters to reach expected results.

Merge view

- It is now possible to merge a null value on a mandatory field if the **Check null input** property is inactive in the **Merge view**.
- The **Merge view** has been enhanced to allow paging in the dependency table.
- You can configure an output parameter to retrieve the Golden record when accessing the **Merge view** from a workflow.
- You can now select two or more auto-created records then access the **Merge view**.
- It is now possible to switch the Pivot in the **Merge view** by setting the **Apply EBX permission on merge view** property to true in the **Process policy** configuration.

Workflows

- It is now possible to select a dataspace, dataset from a combo box when configuring a workflow.
- By configuring the **Hide the Back to tabular view button** property, the **Back to tabular view** button can now be hidden when accessing the **Matching** view workflow.
- The **Align foreign keys** service is now available from a workflow script task.

Updated functionality

- The **(Light) Data quality sterwardship** and the **(Full) Data quality stewardship** menu items have been merged into the **Data quality stewardship** option.
- In the Matching configuration, when there is more than one attribute in a multi-valued group, you no longer have to select an attribute and save to refresh the other drop-down lists.
- The **Table trusted source** value is now applied for all fields by default if the **Field trusted source** is not configured.
- The **Search** feature has been removed from the **Matching** view.
- The **Align foreign key of all merged records** operation now fixes records across different dataspaces and datasets.
- A progress bar is added to display status when executing the **Align foreign key of all merged records** operation.
- You can now configure a condition for the **Default survivorship function**.
- An auto-created golden record cannot stand alone in a cluster or being a single golden.
- If a record no longer exists when executing the **Display metadata** service, the raw value is displayed instead of the link to the record.

API updates

This release contains the following API updates:

- The API's new `executeSurvivorshipOnClusters()` method allows you to execute survivorship on a list of clusters.
- The API's new `unmerge()` method allows you to execute unmerge on a record.
- The **MatchingOperations.matchSelection()** API has been enhanced to get better results.
- It is possible to disable or enable the trigger when executing the **MatchingOperations.exactMatchAtOnce(TableContext,ExactMatchAtOnceContext)** API.
- You can now execute the **simulateMatch** operation through the **Matching REST** service.
- You can now execute the **Align foreign keys of all merged records** or **Align foreign key of merged records** services through the **Align foreign keys** script task.
- It is now possible to configure the **Source record(s)** property as a table record XPath expression in the **Records linking analysis** script task.

Bug fixes

This release contains the following bug fixes:

- **[DAQA-1004]** An incorrect result is returned when running the **Fix relation records** service and the relation table has no matching configuration.
- **[DAQA-3073]** A blank page is displayed when running a **Cleansing** operation.
- **[DAQA-3136]** A NullPointerException occurs when executing the **Records linking analysis** script task.

31.17 Release Note 2.3.2

Release Date: March 25, 2019

New features

This release contains the following new features:

- The **Match table** operation is executed when modifying an **Unmatched** record with the **On Modification** option activated.
- Permissions are now by-passed when executing an **Automatic merge** operation.
- The **Match table** operation is executed when modifying a **Merged** record with the **On suspect record retention** option activated and the **Merged** record is not matched with its target.

Bug fixes

This release contains the following bug fixes:

- **[DAQA-2957] Survivorship** is not applied when accessing the **Merge** view.
- **[DAQA-2959]** An incorrect result is returned when using the **simulateMatchTable(SearchContext)** API on a large set of records.
- **[DAQA-2992]** After executing the **Match table** service, all records are displayed when accessing the **Light view** via a workflow.

31.18 Release Note 2.3.1

Release Date: December 13, 2018

New features

This release contains the following new features:

- The EBX® Match and Merge Add-on has undergone extensive updates to ensure compatibility with the EBX® 5.9.0 Fix A release.
- All add-on functionality now works as expected on the Firefox web browser.
- When fixing an impacted cluster, the **Auto create new golden for single golden** option is applied to set the record as the Single Golden.
- The **Handle null value matching** property is now applied when executing matching with the **Funneling matching** mode activated.
- Matching operations have been enhanced to get better results.
- The **Unmerge** operation will unmerge all merged records that target the Golden or the Pivot.
- The number of active records in the cluster cannot be greater than the value specified by the **Max number of records in cluster** property plus 1.
- When a Pivot record is manually removed from a cluster, the add-on fixes the cluster to improve results.

- The add-on executes matching on the Pivot or the Golden when modifying a Merged record with the **Modify merged without match** option deactivated. The add-on follows this behavior regardless of whether the Pivot or Golden is auto-created.

Bug fixes

This release contains the following bug fixes:

- **[36849]** When a configuration uses multiple relationships on the same table to find possible matches, only the score found using the last relationship is returned.
- **[36850]** The API's `MatchingEventListener.onUnmerge()` is not taken into account when executing an `Unmerge` operation.
- **[36851]** An incorrect result is returned when executing matching using a multi-valued matching field and the **Funneling matching** property is activated.
- **[36853]** A Null pointer exception occurs when accessing the **Merge view** when the **Options for direct matching UI merge** property is not defined.
- **[36872]** A Deadlock happens when executing **Simulate match table** and modifying the matching configuration at the same time.
- **[36972]** An incorrect result is returned when executing matching on a large number of records and the **Max number of records in cluster** property is set to a small value.

31.19 **Release Note 2.3.0**

Release Date: October 26, 2018

Updates and enhancements

This release contains the following updates and enhancements:

- The EBX® Match and Merge Add-on has undergone extensive updates to ensure compatibility with the EBX® 5.9.0 GA release.
- The **Bottom view** is now hidden by default when accessing the **Full view**.

Bug fixes

This release contains the following bug fixes:

- **[33934]** An incorrect number of selected values in [List] is returned in the Merge view.
- **[34607]** Executing the **Exact match at once unmatched** service does not return the correct number of records that have undergone a matching operation.
- **[34618]** An incorrect message is displayed when executing the **Exact match at once** service with the **Execute survivorship** option activated.
- **[34637]** An error message is raised when running match table when a **Condition for field value survivorship** is configured.
- **[34649]** An incorrect result is returned when executing the **Exact match at once** service with the **Auto create new golden in match at once** option activated.
- **[34837]** An incorrect result is returned when matching **Kanji text** with Japanese as the second algorithm.

- **[34852]** An unexpected error occurs when executing the **Match at once (unmatched)** service.
- **[34995]** An incorrect result is returned when executing matching with a foreign key matching field and the **Score calculation** option as **Real Score**.
- **[35065]** All records are displayed when accessing the **Light view** via the perspective mode.
- **[35253]** If a group has at least two auto-created records, it is not fully collapsed after executing the **Match at once** service.

Known limitations

When using the Firefox browser, some add-on functionality does not work as expected.

31.20 **Release Note 2.2.5**

Release Date: October 26, 2018

New features

The API's new `matchSelection()` method allows you to execute matching on a specific selection.

Bug fixes

This release contains the following bug fixes:

- **[36121]** All records are displayed when accessing the **Light view** via the perspective mode.
- **[36199]** The **Cancel** and **Back to main view** buttons do not work when accessing the **Merge view** via the perspective mode.

31.21 **Release Note 2.2.4**

Release Date: October 12, 2018

Bug fixes

This release contains the following bug fixes:

- **[36047]** A 404 error is returned when opening the **Merge view** on a WebLogic server.

31.22 **Release Note 2.2.3**

Release Date: October 5, 2018

New features

This release contains the following new features and enhancements:

- The `SurvivorFunctionSpec` API now allows you to return the current `ProcedureContext`.
- The **Survivor record** is now selected from among the active records in the cluster regardless of their scores.

Bug fixes

This release contains the following bug fixes:

- **[35619]** The incorrect survivor record is selected to create a new golden when using **Auto create new golden in match at once** and **Merge all records in cluster**.
- **[35710]** The **Auto create new golden** property is not applied when no default survivorship policy is configured.
- **[35730]** The second algorithm score is empty when using **Real score** for **Score calculation** and the second algorithm is applied.

31.23 Release Note 2.2.2

Release Date: July 31, 2018

Bug fixes

This release contains the following bug fixes:

- **[33907]** When a Golden record in a cluster is modified and no survivorship policy is defined, its state is set to Pivot.
- **[34372]** The **Real score** option is not applied when executing matching on a foreign key field.

31.24 Release Note 2.2.1

Release Date: July 17, 2018

Enhancements

Memory usage has been enhanced when executing the **simulateMatchTable** API with SearchContext.

31.25 Release Note 2.2.0

Release Date: June 22, 2018

New features and enhancements

This release contains the following new features and enhancements:

- Merged records are now taken into account in all **Merge** operations.
- A merged record's target is now updated when the merged record is moved to a new cluster.
- The **Auto create new golden for single golden** option is now applied for the **Set golden** and **Remove and set golden** operations.
- All built-in **Survivorship function on field**(s) take into account the **Condition for field value survivorship** for every record.
- A default Survivorship policy is no longer required.
- When a Suspect record is manually removed from a cluster, the add-on checks if there are any remaining suspects left in the cluster, and turns the Pivot into a Golden.
- The **Survivorship function** is re-calculated when a Merged record is manually removed.
- A new **purgeAtOnce** API has been added to physically delete all records in the specified Matching state.

- EBX® permission settings are now applied to the **Merge** view.
- An error message is displayed in the **DEBUG** mode only when the Activity Monitoring Add-on is not deployed.
- After a cluster has undergone a matching operation, if it contains only Suspects, a Pivot will be selected and the add-on will run **Match cluster** based on the new Pivot.
- A new **Run survivorship on clusters** table service has been added to execute **Survivorship** on all existing clusters for Merged records.
- A new **Update the latest cluster number** service has been added to update the **Latest cluster number** field in the Table configuration.
- The **RemoveWordFilter** filter has been enhanced.
- Any redundant **auto-created** records will become **Deleted** in the cluster.
- The **survivorship** process can now be applied to clusters that contain records with surrogate matching scores.

Bug fixes

This release contains the following bug fixes:

- **[32521]** The timestamp of auto-created record is not updated after the state is changed as Pivot/ Golden or the data is updated.
- **[32798]** A matching operation returns less results than expected when using the **Filter by** option.
- **[32864]** Users cannot proceed from the merge summary screen when merging a read-only field without applying EBX® permission.
- **[32883]** Users cannot select data of a list field by clicking on the cell in the **Merge** view.
- **[33137]** A Null pointer exception occurs when executing **Match at once** in **Full mode**.
- **[33581]** An unexpected error occurs when modifying a record in the **unset** state.

31.26 **Release Note 2.1.0**

Release Date: April 20, 2018

New features

- The value of an auto-created record is ignored when using the **Most frequent** survivorship function.
- When the **Automatically create new golden** property is set to **Yes**, the Pivot record is now re-selected based on the **Survivor record selection mode** property setting. This process occurs before the new golden record is automatically created.
- A cluster cannot have more than one auto-created record.
- The **Real score** calculation strategy is now applied for all algorithms and scoring criteria.
- An exception is thrown to avoid an invalid **TableContext** being passed to **MatchingOperations.removeFromNotSuspectWith(TableContext, PrimaryKey)**.
- An auto-created record is always prioritized to be selected as Golden regardless of the **Auto create new golden** configuration.

Bug fixes

- **[31842]** The **Last survivorship policy code** field is not updated correctly after merging data in the **Merge view**.
- **[31971]** The **Preview** button of a single foreign key inside a list shows the wrong record.
- **[32040]** The **Triggered action output** variable is set to null after completing a **Merge view** workflow.
- **[32091]** When moving a golden record into a new cluster, it becomes Suspect (-1).
- **[32138]** In the **Merge view**, the **Preview** button does not work on a foreign key list, or the list included in the summary step.
- **[32193]** The **Merge view** displays incorrectly when the primary key field is hidden in the permission settings.
- **[32297]** The lower view in the **Full stewardship** screen is not refreshed after running table services.
- **[32309]** A new auto-created golden is not created when **On process driven**, **On simple matching** and **Embedded at submit** are activated.
- **[32379]** An auto-created record that is in the Deleted state is taken into account when running **Match at once by group**.

31.27 **Release Note 2.0.1**

Release Date: March 28, 2018

New features

- The data in the **Relationship** table is automatically updated upon start-up and data model change.

Bug fixes

- **[31827]** The **Accept** button does not appear after merging a record in a workflow.
- **[31829]** The **Set to Golden** and **Set to definitive golden** steps cannot be by-passed for Merge view in workflow.
- **[31873]** An infinite loading screen is shown when accessing Merge view workflow on a table without the Table configuration.
- **[31883]** A JavaScript error appears when including the **Merge view** inside an Iframe, or other domains.

31.28 **Version 2.0.0**

Release Date: March 16, 2018

New features

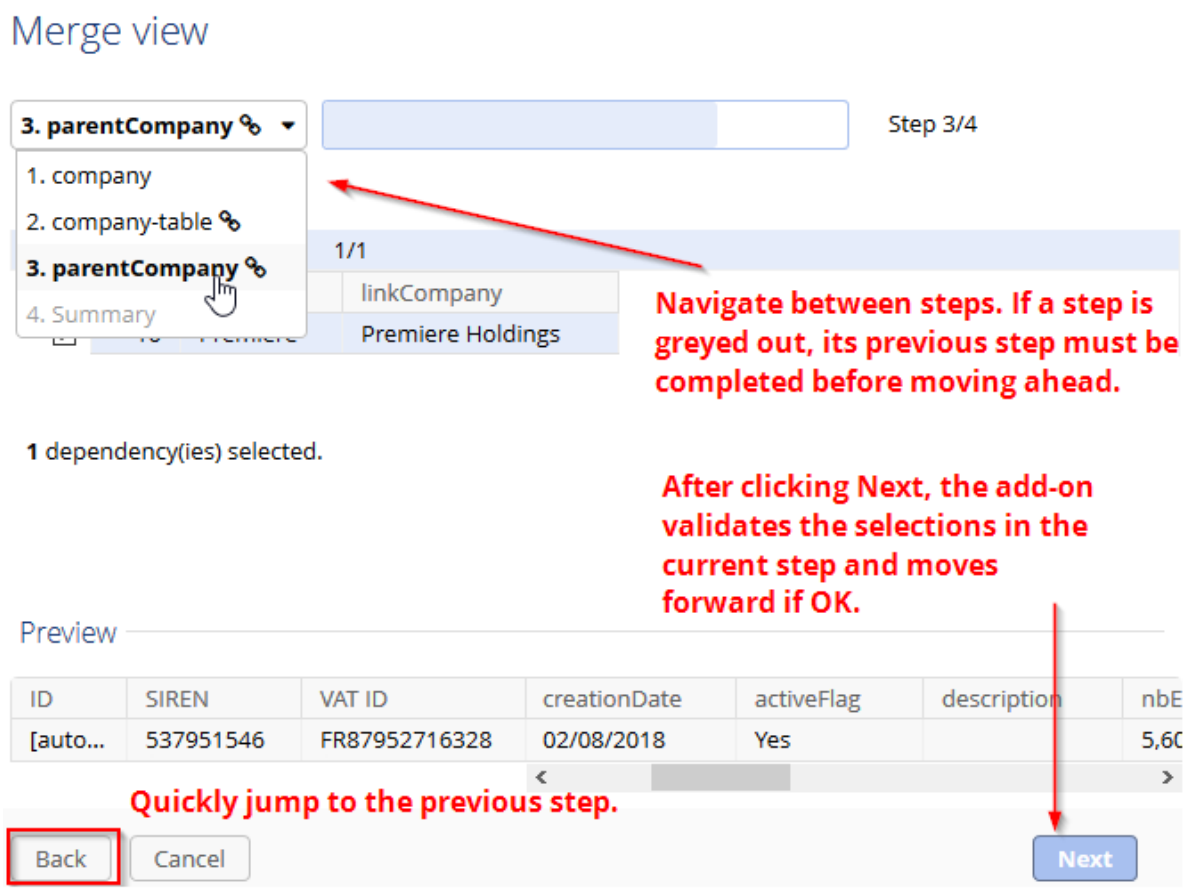
As discussed in the following sections, the **Merge view** has undergone significant interface updates and improvements.

- Navigation improvements [p 299]

Navigation improvements

The **Merge view** now walks you through steps to complete the merge process. The number of steps depends on your data structure and configuration settings. For example, the merge process may impact data not directly involved in the merge due to foreign key relationships with other tables. You will be given the opportunity to address each one of these dependencies.

The bottom of the page includes **Next** and **Back** buttons to move one step at a time. The upper-left part of the view contains a navigable dropdown list of all steps required to complete the current merge operation. The list allows you to return to any previously completed step.



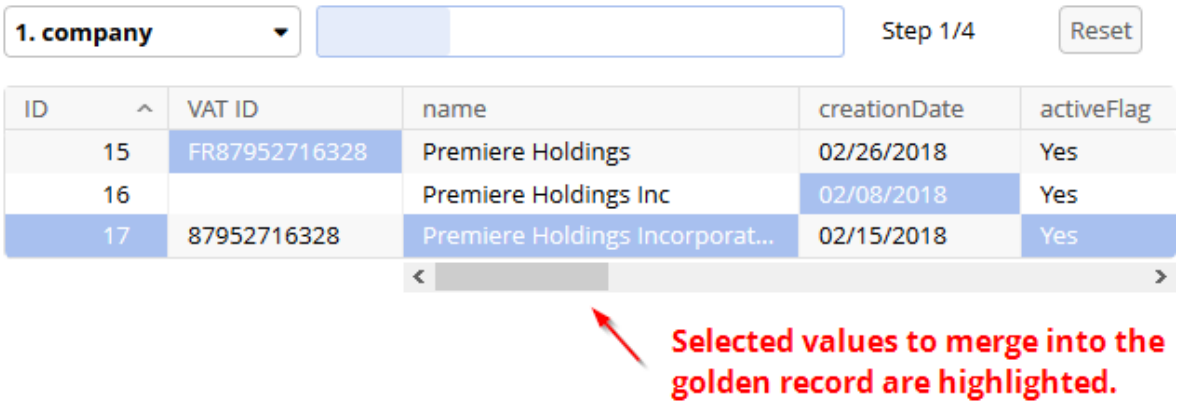
You can lock columns to keep them visible while scrolling through others. Click the icon when hovering over a column's title. Locked columns stack left to right after the primary key column. Hover over a column's primary key and click the icon to display a preview of the record. Additionally, horizontal scrolling is synchronized so that you see the same information in the main table and the preview record.

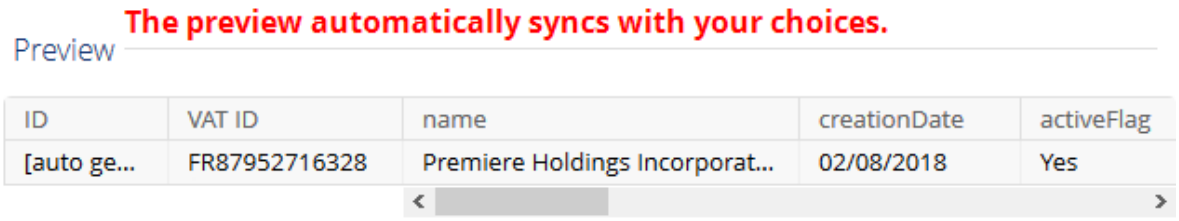
Enhanced value selection

A significant part of a merge operation involves selecting the values you want to be merged into a golden record. The updated interface allows you to select values in the following ways:

- **Selecting field values:** Click a field to select its value for inclusion in the golden record. If you click the first field of a record, the entire record gets selected. You may want to perform this step

first to provide a reference record that will be displayed in the preview section. All subsequent selections are reflected in the preview record at the bottom of the page.

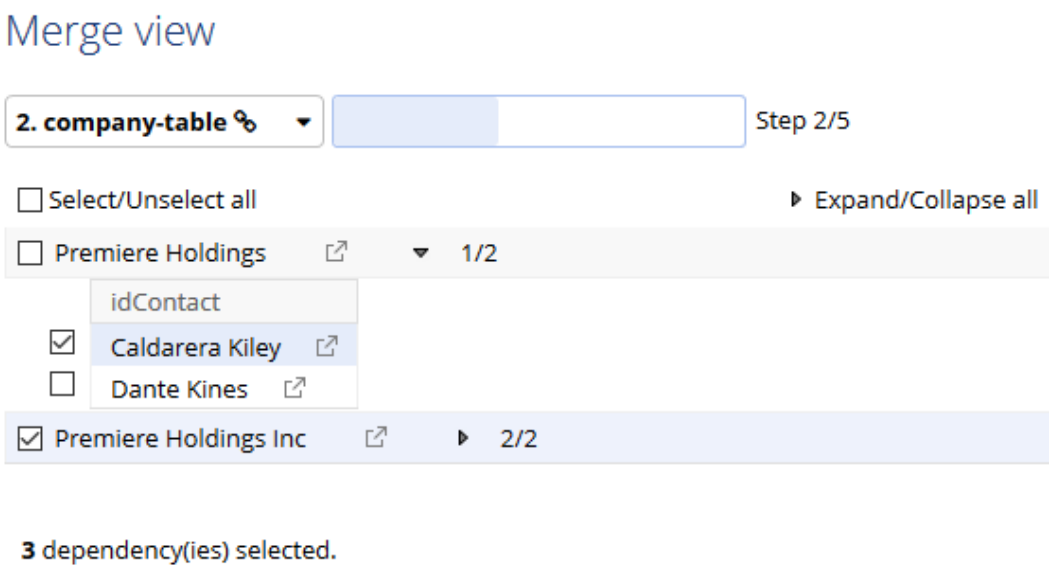




- **Selecting list values:** Some fields, such as enumerations and groups, can contain a list of values. You have control over which of these values are included in the merge. To choose the values,

click the field's icon and select the desired values. The counter shows how many of the possible values are included.

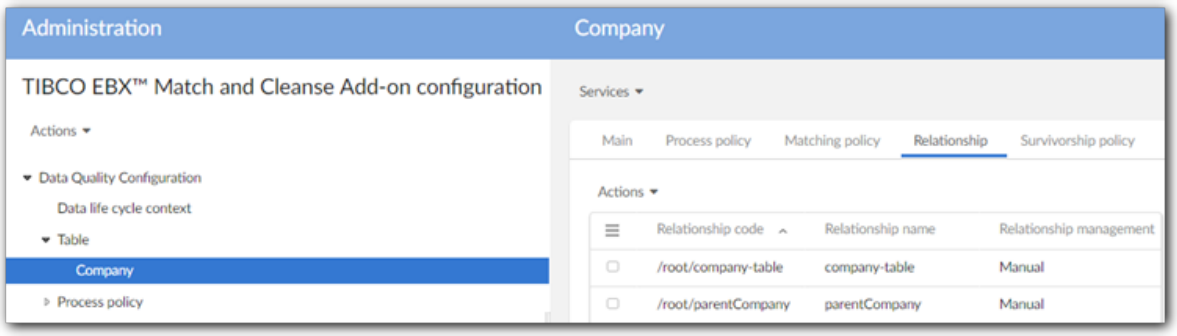
- **Selecting dependencies:** After completing the first page in the **Merge view**, you will be guided through choosing which dependencies (from related tables) are included in the merge. The add-on presents a different page for each dependency.



New Relationship tab

This release includes the addition of the **Relationship** tab. The tab displays information about relationships between a table configured for matching and related tables.

The add-on automatically performs a lookup and populates this table. Administrators can set the relationship management behavior for each relationship.



See also

About the Merge view *[p 200]*

Merge view actions *[p 200]*

Steps to merge records *[p 204]*

Relationship dependencies *[p 206]*

Additional updates and improvements

This release contains the following new features and enhancements:

- A matching algorithm to search Japanese text is now available.
- Descriptions have been added to each algorithm in the **Matching algorithm** table.
- Only **Unset** records become **Unmatched** when modifying a record with the **On modification** property set to **False**.
- It is now possible to get **Crosswalk results** directly without saving them into the crosswalk data set.
- A new **Narrow search** property has been added to the **Matching policy** table to reduce the memory usage for matching operations.

Bug fixes

- **[30122]** An error occurs when modifying a Suspect record with the **On cluster retention for matching suspect** property activated.
- **[30260]** An incorrect result is returned when matching with the Exact algorithm and the synonym group contains numeric and special characters.
- **[30285]** An incorrect result is returned when matching a multi-occurrence list with the Exact algorithm and the **Ignore case** property activated.
- **[30310]** It is not possible to match with a multi-occurrence matching field when using the Filtering field rule and the Exact algorithm.
- **[30327]** It is not possible to match a multi-hop foreign key with the Exact algorithm.
- **[30616]** An error occurs when running the **Match at once** and **Exact match at once** operation at the same time.

- **[30697]** The crosswalk matching field cannot be updated when there is an error in the crosswalk matching configuration.
- **[30701]** An incorrect result is returned when matching multi-occurrence foreign key hop with smart synonym.
- **[30718]** Results are inconsistent when executing the Crosswalk service twice.
- **[30753]** An incorrect score is returned when executing the Crosswalk service with at least two matching fields configured.
- **[31697]** The **Change the record's state to Suspicious** workflow script task only works with the **Golden** and **Suspect** states.
- **[31700]** The **Change the record's state to Suspicious** workflow script task does not work when the **Specific tracking info to use** property is defined.
- **[31710]** The **Check if record is in a given state** workflow condition does not work properly.

31.29 **Release Note 1.13.1**

Release Date: February 28, 2018

New features

This release contains the following new features and enhancements:

- It is now possible to configure auto create new golden when the primary key fields are not auto-incremented.

31.30 **Release Note 1.13.0**

Release Date: January 31, 2018

New features

This release contains the following new features and enhancements:

- You can now configure the **Latest cluster number** property to define the most recently used cluster number. Additionally, the **Reinitialize the latest cluster number** service has been improved to cover all datasets.
- When a matching operation entails moving a target record to a new cluster, the add-on now recursively moves the target's merged records to the new cluster.
- A new **Score calculation** property has been added to **Matching field** to provide a new strategy for score calculation and can be applied to **filtering field** and **smart synonym**.
- The clusters containing only the Pivot and Merged records are now fixed.
- The **Ignore case** property is now applied for Synonym matching.
- A new **Always apply survivorship** property is added to merge records even if their scores are less than the maximum score but greater than the minimum score.
- **RemoveValueFilter** is now optimized, the filtered values are sorted in descending order and you can put a space at the beginning or the end of the string.

Bug fixes

- **[30289]** In some cases the combination of **Filter By** option with context matching policies can lead to an incorrect result.
- **[30794]** A NullPointerException is thrown when using simulateMatchTable API on a table on which one of the records has a broken foreign key.
- **[30815]** An incorrect score is returned when using simulateMatchTable API on a table without the Daqa metadata.
- **[30988]** An error occurs when executing the **Exact match at once** operation with the **Auto create new golden** and **On survivorship** options activated.

31.31 **Release Note 1.12.2**

Release Date: December 22, 2017

New features

- Case sensitive search can now be disabled for all algorithms.
- A new **RemoveWordFilter** class is available for the field filtering feature. Only whole words will be filtered.
- A new API is available to listen to the Unmerge event.

Bug fixes

- **[30159]** A blank page is displayed when running the **Align foreign key of merge records** service on a record with a composite primary key.
- **[30193]** The navigation context is lost after using the **Display metadata** service.
- **[30199]** An incorrect matching result is returned when running **Match at once by group** with **Business ID** and Exact **Literal score** defined.
- **[30258]** An exception is thrown when using the API to run `simulateMatchTable(SearchContext)` on a table without Matching metadata.

Backward compatibility

- Due to an API update in this release, all custom code that uses the `MatchingEventListener` must be recompiled and redeployed.

31.32 **Release Note 1.12.1**

Release Date: December 15, 2017

Bug fixes

- **[29668]** ClassCastException is thrown when matching with a multi-valued Date or Numeric field.
- **[29893] Workflow context** is not taken into account when executing the **MatchAtOnce** operation via APIs.

- **[30003]** An incorrect matching result returns when executing matching on Multi-occurrence foreign key field with the **Handle null value matching** mode activated.
- **[30010]** An incorrect matching result returns when executing matching on an empty Multi-occurrence foreign key field.
- **[30011]** A NullPointerException is thrown when updating a Matching field record via the **Import XML** service.

31.33 Release Note 1.12.0

Release Date: November 10, 2017

New features

- It is now possible to apply the **Automatically create new golden** property when creating a new record.
- It is now possible to apply **Survivorship** to merge data even if the **Golden is preserved for selection** property is activated.
- You can set a value of up to two billion for the **Max nb. of matches per state** property.
- The **Auto-created** golden will be retained in the cluster.
- The **Auto-created** DaqaMetaData field will not be reset at any time.
- A record will be ignored when applying **Survivorship function** if its survivor field value is null.

Bug fixes

- **[29042]** An incorrect result is returned when running The **Align foreign key of merged records** service.
- **[29054]** An incorrect result is returned when Running **Match cluster** on Pivot record using matching through relation(s).
- **[29126]** Merged record becomes Suspect (-1) after running Match table on its target record with the **Modify merged without match** property activated.
- **[29129]** An exception occurs when matching with the **Auto create new golden** and **Customize source value for new golden** property activated.
- **[29138]** The records' scores are not re-computed against the pivot record after changing algorithm and adding a record in the cluster.
- **[29158]** Executing the **Match cluster** operation with synonym group configured return incorrect result.
- **[29198]** Executing the **Match cluster** operation with the **Exact** algorithm and numeric value on matching field returns incorrect result.
- **[29205]** The surrogate score returned is smaller than **Field stewardship min score (%)**.
- **[29246]** A `NullPointerException` is thrown when saving a matching field record with an invalid hop configuration.
- **[29292]** Suspect becomes Suspect(-1) when running **Match cluster** operation with **Handle null value for filter by**.
- **[29298]** Monitoring is activated when running parallel matching.

31.34 Release Note 1.11.0

Release Date: October 16, 2017

New features

- You can now execute the **Record linking analysis** service on a set of selected records.
- It is now possible to specify the target data spaces and datasets when executing the **Record linking analysis** service from the API, or workflow script task.
- A new Crosswalk API allows you to select the configured target table at execution time.
- You cannot execute **Match at once** in parallel mode when there is a configured replication in commit mode.
- A new cache mechanism has been used to optimize performance.
- The relational model is no longer supported in matching.
- The **SearchDate** and **Exact** algorithm for a date data type or **SearchNumber** and **Exact** algorithm properties can be configured for a numerical data type.
- The search result has been optimized to remove irrelevant results.
- The **Handle null value matching** property for multi-hop foreign key is no longer supported.
- A new API has been added to preload the cache.
- The **Filter by** option in relational tables is now applied for relational matching.

Bug fixes

- **[26174]** An incorrect relational matching configuration screen displays when defining multi-relational matching.
- **[26521]** A JavaScript error occurs, but no error messages display when running Crosswalk with an incorrect configuration.
- **[26565] Stewardship min score** can be set to equal to or greater than **Stewardship max score** in the **Process policy**and **Matching policy**.
- **[26595]** A blank page displays when running the **Statistics** service on a table that contains **Unset** record.
- **[26596]** An exception occurs when there is more than one process policy with defined context activated at **dataset is applied to all dataspace**.
- **[26598]** A blank page displays when users choose more than one record, then run the **Reinitialize the latest cluster number** service.
- **[26730]** All clusters are hidden after running the **Hide cluster** service on a record.
- **[26851]** A user should not have the permission to modify system data.
- **[26858]** The close button is hidden when opening a pop-up window.
- **[27053]** An **Unset** record is hidden in the **Apply the same value to the selected record(s)** section.
- **[27070]** An incorrect result is returned when users run matching operation with a date list field and use the **Exact** algorithm.
- **[27162]** An incorrect result is returned when the values of synonym group have the same prefix.

- **[27163]** The value of **Fields** in Matching meta data is not updated after running the **Set at once** service.
- **[27308]** The value of **Fields** in Matching metadata is removed after matching on suspect record.
- **[27385]** An error message displays when running the **Group at once** service with a void matching policy.
- **[27425]** It is possible to configure a surrogate field with a computed data type.
- **[27500]** A blank page displays when configuring a filter in Matching policy configuration.
- **[27515]** The pivot is displayed after running **Future merged records** in the Merge view.
- **[27529]** The value of **Fields(%)** field is not updated in the metadata after running the **Exact match at once** operation.

31.35 **Release Note 1.10.1**

Release Date: May 9, 2017

New features

- The **Auto create new golden in match at once** is now applied for **Exact match at once** operation.

Bug fixes

- **[25674]** The progress bar does not update when running **Exact match at once** with a void Matching policy.
- **[25685]** An unexpected error occurs when merging a record automatically without defining a Survivorship function for the Survivor field.
- **[25732]** The **Display record** service returns the wrong record in simple matching screen.
- **[25799]** Clicking on the **Merged record primary key** link under the **Merged field logging** group metadata displays an incorrect record.

31.36 **Release Note 1.10.0**

Release Date: April 18, 2017

New features

- A new **Keep not matched records untouched** property has been added at the matching policy level to retain the current states of records that do not produce a match. The property only applies to **Match at once** and **Exact match at once** operations.
- It is now possible to define a condition as a predicate expression that determines whether a field is survived.
- When a match results in a single golden, if **Auto create new golden for single golden** is activated, a copy of the single golden record is automatically created and these two records are put in a new cluster.
- The **Exact match at once** operation with a void matching policy completes even in memory mode.

31.37 Release Note 1.9.2

Release Date: March 31, 2017

New features

- The performance when using matching with synonyms has been optimized.

Bug fixes

- **[25093]** An exception occurs when a **Data life cycle context** referring to a deleted dataspace is called during matching.

31.38 Release Note 1.9.1

Release Date: March 20, 2017

Bug fixes

- **[24860]** A JavaScript error occurs when accessing the **Full matching**, **Light matching**, **Simple matching** and **Merge** views through a specific portal.

31.39 Release Note 1.9.0

Release Date: March 6, 2017

New features

- It is now possible to apply **Survivorship** when **Pivot selection mode** is different from **New updated**.
- The **Modify merged without match** property has been added to the **Matching Policy** table.

New APIs

- The new **MatchingStateFactory** API has been published.

31.40 Release Note 1.8.6

Release Date: February 23, 2017

Bug fixes

- **[24366]** It is not possible to modify the **To this date** input parameter while configuring a **Deprecated** cleansing procedure.
- **[24368]** When running **Deprecated records** cleansing, the **Current record** and **Next record** display incorrectly in the **Manage manually** screen.
- **[24373]** The **Apply the same value to the selected record(s)** field is blank when the input parameters include special characters.

- **[24514]** The `ExcludeMatchingFieldFilter` is applied regardless of whether the **Exclude records from matching** property is configured in the **Matching policy** table.

31.41 Release Note 1.8.5

Release Date: January 23, 2017

New features

- When a field's score is lower than the configured **Stewardship min score**, the add-on matches with a surrogate field instead of using the defined matching field.
- The **Match at once** operation will ignore the void matching policy to proceed, provided that it does not run in memory mode.
- It is now possible to run all records' services with the void Matching policy.
- When a match is executed on Suspect records with the **On suspect record retention** option deactivated, all Suspect (-1) records are moved to the Unmatched cluster.

New APIs

- It is now possible to get the Pivot record when using `MatchingRecordFilter`.
- A new API was added to execute **Exact match at once** on a record with the **Automatic merge** option activated.

Bug fixes

- **[23732]** The **Exact match at once** service still functions normally, although both **In memory** and **Is forced to void** options are activated.
- **[23734]** It is impossible to input integer values in the second input parameter of the **Deprecated records** cleansing procedure.
- **[23801]** Running Crosswalk with **SearchDate** algorithm activated returns an incorrect result.
- **[23808]** The Merged record is still modifiable after you manually replace a missing field value in the cleansing screen.
- **[23873]** It is impossible to export the statistics information into an Excel file in Full or Light view perspective screen.
- **[23874]** The **Hide state filter** and **Display cleansing buttons** options in Full or Light view perspective are not applied.
- **[23908]** Running **Match table** on a Suspect does not change the best record's state to Pivot with the **On survivorship** option deactivated.

31.42 Release Note 1.8.4

Release Date: December 16, 2016

New features

- The **Unset Golden recursively** operation is now available on Golden records.

- When using a table in more than one Crosswalk matching field, it must occupy the same location in each field. For example, a field from a Party table may be defined first in one Crosswalk matching field. If you use a different field from the Party table in another Crosswalk matching field it must also hold the first position.
- When modifying a merged record, the survivorship policy is applied to merge data to its Pivot or Golden target record.
- **Filtering record rule** is now applied when matching through related tables.

New APIs

- It is now possible to filter records when searching with `SearchContext`.
- An API method has been added to simulate a match table operation with `TransientRecordContext`.
- Two APIs have been added in `MatchingEventListener` to record whenever a new Golden or Pivot record is automatically created.

Bug fixes

- **[23271]** The **Handle null value** options are not taken into account when executing relational matching.
- **[23655]** When modifying a Merged record, the **Most recently acquired** survivorship function does not work properly.

31.43 **Release Note 1.8.3**

Release Date: November 18, 2016

New features

- It is now possible to configure **Matching policy context** using a foreign key raw value.
- The same number of crosswalk fields has to be used and they must belong the same target table.
- When running crosswalk, the target table selection has been optimized to be more user-friendly.

Bug fixes

- **[23090]** An incorrect score is returned when running crosswalk with different **Stewardship min score**.

31.44 **Release Note 1.8.2**

Release Date: October 28, 2016

New APIs

- It is now possible to launch Crosswalk using the API.
- The `simulateMatchTable(RecordContext)` API was enhanced with the ability to compare records.

31.45 Release Note 1.8.1

Release Date: October 12, 2016

New features

- A non-empty matching field context can now be defined.
- A **Match at once** operation can be performed simultaneously on groups of data.
- It is now possible to select the configured target table at execution time in crosswalk.
- A **Bidirectional not suspect with** property has been added to the **Process policy** table to save the Pivot record information into the metadata's **Not suspect with** list on the Suspect record.
- The **Switch pivot** and **Not suspect** services are now available on Suspect records in the **Merge view**.

Bug fixes

- **[22178]** An incorrect score is returned when running matching using two algorithms with foreign key or foreign key hops matching fields.
- **[22302]** An incorrect result is returned when running Match at once with the **Auto create new golden in match at once** option active.
- **[22456]** An exception is thrown when running **Match at once** using the **In memory** and **Matching policy context**.
- **[22560]** An incorrect result is returned when running Match table with the **Funneling matching** option activated and **Handling null value matching** checked.

31.46 Release Note 1.8.0

Release Date: September 9, 2016

New features

Matching

- **Clean up merged field log** is a newly added service available on all states (except Merged and Deleted) to delete saved **Merged field logging** in metadata.
- It is now possible to view relation records in the **Merge view** to determine whether the current suspect record is a duplicate.
- The **Merge view** can be launched by selecting records in the tabular view and executing the **Merge records** service.
- It is now possible to apply **Automatically create new golden** to Match at once operations.
- The **Foreign key** property is now available in **Matching field** configuration records. This property allows you to define a path to another foreign key field that you want to run matching on by creating hops to navigate the relationships.
- The **Activate monitoring** option is now available when configuring a table and allows you to determine whether the EBX® Activity Monitoring Add-on saves **Match at once** execution status.

- The **Match at once (on clusters)** service is available in the full matching view to execute the **Match cluster** operation on selected clusters.
- You can now view matching statistics and export them to an Excel file.
- It is now possible to select a matching policy context by using a Java class.
- A merged record will be moved to its target's cluster when executing the **Add into cluster** operation.
- The **Add into cluster** service is now available on merged records.
- The **Ignore case for Exact match** property has been added to allow you to ignore case when matching.
- Null values are now taken into account when executing matching with any algorithms.
- **SearchDate** and **SearchNumber** algorithms are now available.
- The **Simulation result** output parameter has been added to the **Simulate match table** script task to return search results formatted in JSON.
- Null values are now taken into account when applying **Filter by**.
- You can now clean up the list of **Not suspect with**.
- The **Item in group** table is now visible by default.
- It is possible to merge records with a score of -1 in all merge operations.
- It is possible to match fields under multi-value groups.
- The **Check EBX® Match and Merge Add-on configuration** service validates the **Source field** value when the **Most trusted source** mode is selected.
- The new **Smart synonym matching** option allows you to use matching with the advanced synonym mechanism.

Cleansing

- It is now possible to manually fix a set of records.
- The **Save and apply fix to next** button has been added to manual fix cleansing views.
- The field value can be replaced with the derived value in the preview record screen.

Crosswalk

- It is possible to match on multiple fields in crosswalk.
- The **Maximum number of results** property has been added to the **Crosswalk process policy** table to set a threshold on the number of records saved in the **Crosswalk additional result** table.

Bug fixes

- **[19054]** Records (except Golden and Pivot) have a **List of not suspects** value in metadata after running the **Force records to merged** service.
- **[19253]** There are some matching services that cannot be run by read-only users.
- **[19289]** Descriptions of cleansing procedure records with an [ON] prefix are not translated into French.

- **[19294]To be matched** and **Unmatched** states are not translated into French in **Set at once** screens.
- **[19623]** All predefined clusters display when running the **Add into cluster** service.
- **[19658]** An execution procedure's state is terminated before the transaction finishes.
- **[19692]** An exception is thrown when running **Match table** on suspect records with null score value.
- **[20336]** A record with a score below the max score threshold becomes a survivor when running matching with **was golden** selected for the **Survivor record selection mode** property.
- **[20358]** Merged records with a score of '-1' display after running **Exact match at once** with **Execute survivorship** and **Filter by** enabled.
- **[20397]** Session times out when running **Match suspicious** in the data workflow.
- **[20399]** Services on the **State** filter button do not work properly in Light view when accessed from a workflow.
- **[20401]** The **Select all** check box does not work properly in set at once screens.
- **[20564]** The suspicious record is selected as the best record in the simple matching view when **Best record selection mode** is set to **Was golden**.
- **[20755]** It is not possible to use the scroll bar to view the whole **Display record** screen in Simple matching view at Submit time.
- **[20842]** An exception occurs after creating a new record with the foreign key matching field containing render label.
- **[20916]** A redundant scroll bar displays when running the **Create new golden** service on Internet Explorer browser.
- **[21142]** The matching policy with defined workflow context is not applied when running services in Simple matching view under workflow.
- **[21182]** The golden record becomes golden in cluster 001 when there is no record matched.

31.47 **Release Note 1.7.11**

Release Date: July 8, 2016

Bug fixes

- **[21265]** Matching services are disabled in all the dataspaces of a slave or hub instance.
- **[21350]** The incorrect result is returned after running matching on a record when the **No match records when same source** property is set to **Yes**.

31.48 **Release Note 1.7.10**

Release Date: May 19, 2016

New features

- Additional Pivot record selection modes are now available.

- **Match at once** full mode and **Match table** on suspect have been optimized to get more consistent results.
- The cluster can be saved when matching suspect records using the new **On cluster retention for matching suspect** property.
- The states of records in clusters that contain only Suspect or Pivot can be changed to Golden when running **Match at once** in **Full mode**.
- Error messages are logged in **Debug** mode when the table is not configured and has only **Daqa metadata**.
- The log has been enriched to specify an error when migrating a data model if the dataspace or dataset is deleted.
- **Match at once** full mode now can be executed when there is no record in the corresponding state.
- The record will not be moved into a cluster when the **No match records when same source** property is set to **Yes** and at least an record (except Merged and Deleted) from the same source already exists in the cluster.
- Matching run on a pivot or golden record, when the **Merged record is recycled** option is on, causes the add-on to include previously merged records in the match operation.
- When running matching on a golden record, it remains in its current cluster if there is no potential duplicate record found.

Bug fixes

- **[19982]** An exception occurs when importing records with **Use matching** activated and **On suspect record retention** set to **No**.
- **[20073]** An exception occurs when using `RemoveValueFilter` and the value of the matching field is null.
- **[20373]** Services on suspect record are hidden in the Merge view when **On matching process** is set to **No**.
- **[20419]** The field values of golden or pivot records cannot be reverted to the previous values after running the **Unmerge** service.

31.49 **Release Note 1.7.9**

Release Date: March 18, 2016

New features

- When executing **Match at once full mode** and **Match table** operation on suspect record, the record will be not match with the records in the same its cluster.

Bug fixes

- **[19666]** Golden records that have scores equal to the min score are not matched after running the **Match table** service on suspect records.

31.50 Release Note 1.7.8

Release Date: February 4, 2016

Bug fixes

- **[19175]** Matching's **Purge at once** services have extended execution times.
- **[19176] Records with scores lower than the min score threshold are returned.**
- **[19221]** A record gets successfully added into the new cluster even when it's score is '-1'.
- **[19337]** It is not possible to return to the **cluster view** after running the **Match suspicious** service while in a perspective.

31.51 Release Note 1.7.7

Release Date: January 18, 2016

Bug fixes

- **[18690]** The progress bar does not update and the **Cancel** button is inactive for a long period of time when executing the **Match at once** service with the **Filter by** property configured.
- **[18682]** An error displays when creating a duplicate record in the workflow when the value of the **Embedded at submit** field is set to **Yes**.
- **[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.

31.52 Release Note 1.7.6

Release Date: November 25, 2015

Bug fixes

- **[18346]** Records with invalid state exist in cluster after running **Unmerge** service on golden or pivot record.

31.53 Release Note 1.7.5

Release Date: November 19, 2015

New features

- It is possible to modify merged record by setting the **Merged record is recycled** property in the process policy.
- The survivorship policy now has new survivorship record selection named **Was golden**. With this selection mode survivorship record will be selected amongst Was golden records and current Golden. If many records with the **Golden** or **Was golden** identifier exist, the **Most recently acquired** record is used.
- The tooltip and user guide for field **Golden is preserved for selection** now is enriched.

- It is now possible to use a survivorship policy when running an **Exact match at once** service.
- An **Exact match at once** service can now be executed in a single in-memory transaction to speed up the matching process. However, if any exception is thrown, all results will be lost.

New API

- It is now possible to change the import mode value for a given process policy. See the `MatchingConfigurationOperations API` for more information.

31.54 **Release Note 1.7.4**

Release Date: November 3, 2015

New features

- The **Profiling** and **Cleansing** buttons can now be displayed or hidden in a workflow by setting the **Display cleansing buttons** property in the workflow configuration.
- The **Accept** button can now be displayed or hidden in a workflow by setting the **Auto complete** property in the workflow configuration.

Bug fixes

- **[17840]** It is not possible to access matching services in closed data spaces.

31.55 **Release Note 1.7.3**

Release Date: October 9, 2015

New features

- Matching based on a multi-value field inside a terminal group is supported.
- The source field's value in the newly created golden record can be customized by using the **Source value** property configured in matching policy.

Bug fixes

- **[17566]** An empty result is returned when searching with a foreign key field using a programmatic label.
- **[17560]** When you launch **Profiling** from the **Full Data Quality Stewardship** service, it raises a null pointer exception and results in a blank screen.

Optimization

Matching process has been optimized.

31.56 **Release Note 1.7.2**

Release Date: September 16, 2015

New features

- EBX® Match and Merge Add-on has been adapted to fulfill requirements from EBX® Insight Add-on and EBX® Add-on for Oracle Hyperion EPM.

31.57 **Release Note 1.7.1**

Release Date: August 24, 2015

Bug fixes

- **[17418]** An exception occurred when running matching on a FK field. The **Localized label** property is empty.

31.58 **Release Note 1.7.0**

Release Date: July 31, 2015

New features

Matching

- You can now execute a **Match table** operation on a suspect record. When a record matches the suspect, it is considered a duplicate if its new score is higher or equal to its current score.
- **Match at once** operations can now use a **Full mode** option to run a match against every record in the table, regardless of any existing configuration settings. This means that any property values which make up a "base" configuration are ignored. The number of matches executed is n*(n-1) for n involved records. This improves matching result relevance, but increases execution time.
- The **Matching policy** table now includes a **Filter by** property that enables you to filter records before executing fuzzy matching. When faced with a large volume of data to match (millions of records), a good practice is to identify criteria on which the data can be grouped. Fuzzy matching is then applied on a suitable subset of records rather than on the whole table.

Matching view

- When displaying a record's matching metadata, the merged record's target is now displayed through a user-friendly pop-up rather than embedded in a frame.
- In a workflow, whether in full or light matching user tasks, the **State** button filter can now be hidden or displayed by setting a property in the workflow configuration.

Simple matching view

- In a workflow (simple matching user task), the **Keep in workflow** service can now be hidden using a property in the workflow configuration.
- The **Cluster view** button and the pop-up menu at the level of the suspect record can be hidden using a property in the configuration.
- A new **Display record** service displays the selected duplicate record through a pop-up. This helpful feature allows you to make a decision on the record before the next stewardship operation.

Merge view

- The **Cluster view** button and the **Suspect** record drop-down menu can be hidden using configuration options.
- A new button allows you to hide/show fields not used in the merging process such as associations, computed attributes, selection nodes, etc.
- Field values that aren't an exact are now highlighted in light coral color.

Hook

- You can now configure an **event listener** java class at the table level. This allows you to add bespoke treatment when a matching state value change occurs on a record.

31.59 Release Note 1.6.2

Release Date: June 29, 2015

New features

- You can now match on a field of any simple type.

Improvement

- When using property **Golden is preserved for selection**, the **Pivot record selection mode** will prioritize records identified as **was golden** and records with **Golden** state existing in the cluster.

31.60 Release Note 1.6.1

Release Date: June 17, 2015

New features

Matching

- In Simple matching workflow, when re-opening the simple matching task once the record is no more in a supicious state, the UI will adapt in order to redirect the user on the accurate screen.

31.61 Release Note 1.6.0

Release Date: June 10, 2015

New features

Matching

- You can now separate the UI of Matching and Cleansing services by using the properties in **Table** configuration. For example, in the **(Full) Data quality stewardship** view, you can create a configuration that displays only Matching services, only Cleansing services or both.
- The relation match feature can now apply to multiple relationships. For instance, the **Address** and **Sales** tables each have a relationship to the **Party** table. You can run a matching operation

against the **Party** table by applying two matching configurations based on the relationships from the **Address** and **Sales** tables.

- Relationship records involved in a merge process can now be fixed (merge, deletion, alignment). For instance, an **Address** table has a relationship to a **Party** table. At the time the parties are merged, you can merge, delete and realign the related addresses.
- A new matching policy **Is forced to void** property allows you to short-circuit matching execution. This makes it possible to define an active matching policy that does not actually execute. This mechanism is useful when many matching policies are defined and depend on different contexts. For certain contexts, you may want to short-circuit matching.
- When configuring a table, you no longer need to refer to a dataspace and dataset. Table configuration now relies on the list of data models to reach the desired tables.
- You can now configure automatic execution of a workflow depending on the matching operation's result (per state value of the matching record: golden, pivot, suspect, etc.) or in the case of survivorship execution.
- The groups of synonyms can now be arranged in a hierarchy mode. When matching uses the synonym mechanism, all groups of child synonyms can be used to look for a synonym.
- The data views used for matching can now be configured by user-profile: full view matching up, full view matching bottom, light view matching, merge view matching, simple matching view.
- A new option-**Automatically align foreign keys**-is now available in the merge UI. This option allows you to automatically execute alignment of foreign keys after the merge. The existing relationships to the merged records are updated to link to the related pivot or golden record.
- In the simple matching view, you can now hide the suspicious record in the list of duplicate records.
- On a table's tabular view, you can execute the **Display metadata** service on a record. This service provides you with the full matching and cleansing metadata.
- You can now configure a **null** value for the **Exclude records from matching** feature.
- The **Merge view** and **Back to table view** buttons are now displayed in the matching view in a workflow procedure.

Cleansing

- New cleansing procedures:
  - **Foreign key fixing** - The broken foreign keys are automatically identified and a UI allows you to fix the defective relationships.
  - **Deprecated records** - This procedure allows you to get the list of the deprecated records (based on a time configuration) and decide to delete or keep the records.
  - **Unused records** - This procedure provides you with a set of rules to detect the unused records and process them (delete, keep, etc.). An unused record is not referenced by any other records in the repository.
- The results from cleansing procedure execution (profiling and fixing) can now be kept over time to use in data analytics.

Crosswalk

- The crosswalk feature is a new domain in the EBX® Match and Merge Add-on. It allows you to create a cross-reference table based on a matching policy executed over many tables. For instance, the **Client**, **Party** and **Customer** tables have duplicate records and you want to create a cross-reference table that shows how the same record is represented in multiple tables. This feature is not used to merge the records but provides a convenient place to view the linked records.

31.62 Release Note 1.5.8

Release Date: April 27, 2015

Bug fixes

- **[15351]** Matching policy code meta-data is not updated correctly when using API: `MatchingOperations.matchTable()` with workflow context.

31.63 Release Note 1.5.7

Release Date: March 17, 2015

Bug fixes

- **[14810]** Exception when using API: `MatchingOperations.simulateMatchTable()` to search with Date field.
- **[14948]** Matching policy code meta-data is not updated correctly when created/updated records become Golden after no duplicate matching.

31.64 Release Note 1.5.6

Release Date: January 26, 2015

New API

- Ability to define the workflow context when simulating matching.

31.65 Release Note 1.5.5

Release Date: December 8, 2014

New features

- Ability via the UI to merge without changing the records states.

31.66 Release Note 1.5.4

Release Date: November 10, 2014

New features

- Add an algorithm to enable matching in Chinese.
- New Java API are available.

Bug fixes

- **[13149]** Matching context is not used correctly when matching context field is a Boolean field.

31.67 **Release Note 1.5.3**

Release Date: October 10, 2014

New features

- Java API are available to run **match at once**, **group at once** and **set at once** operations.

31.68 **Release Note 1.5.2**

Release Date: September 30, 2014

Bug fixes

- **[12599]** Java script error occurs after running **Show cluster**, State/Unmatched in group, State/To be matched in group in the light view.
- **[12587]** Exception occurs after running Align foreign keys of all merged records/merged records with multi-value foreign key field.

31.69 **Release Note 1.5.1**

Release Date: September 12, 2014

Bug fixes

- **[11137] Log out** screen is displayed after running **Remove and set golden** service on greater than or equal to 4 suspect records.
- **[12481]** Exception occurs when using **Not match when same source**.
- **[12526]** Exception when searching with null foreign key using **Exact algorithm**.

31.70 **Release Note 1.5.0**

Release Date: August 29, 2014

New features

Data cleansing

- The EBX® Match and Merge Add-on integrates a new functional domain to manage data cleansing. New UI operations and an API are available to declare and customize a data cleansing procedures portfolio.
- A predefined cleansing procedure that checks for and fixes missing values in tables is provided.
- An API allows you to develop any bespoke cleansing procedures that are integrated automatically in the EBX® Match and Merge Add-on UI.

New service available for suspect records

- A new service (**Remove and set golden**) is now available to handle suspect records. When running this service, the record is set to a golden state and moved to the golden cluster. The cluster is not modified.

New services on the Make golden action in the simple matching UI

- The **Make golden and force all records to merged** service forces all suspicious records to the **Merged** state with their target record linked to the golden record.
- The **Make golden and force selected records to merged** service forces only records that have been selected in the list of suspicious records to the **Merged** state.

Record selection on the Simple matching UI

- It is now possible to configure a record selection policy to automatically identify the **best record** among the list of the potential duplicate records. Stewardship can be executed on this **best record** that is considered the pivot record.

Matching fields

- It is now possible to manage the field order used in a matching policy. It is useful to prioritize the fields that are the most important in the matching execution, in particular when the **funneling** mode is activated.

Custom distance algorithms

- It is now possible to create custom distance algorithms.

New matching metadata

- **Date of last operation code** gives the matching last execution date on a record.
- **Last survivorship policy code** gives the last survivorship policy executed on a record.

31.71 **Release Note 1.4.0**

Release Date: April 1, 2014

New features

Relation match

- It is now possible to run matching on tables that have indirect relationships. For example, a table **Address** has a relationship to the table **Party**. Matching can be configured to de-duplicate the parties based on a policy defined on the **Address**. The relation match works also through join tables. Based on the previous example, a join table **Party-Address** is used to link the **Party** with the **Address**. The matching on **Party** now refers to the matching of **Address** through the join table **Party-Address**.
- A new UI service **Display relation records** allows showing the records involved in the relation match. For instance, applied to a **Party** record, the **Display relation records** gives all the **Address** records for this party.

Synonym management

- A repository of synonyms can now be configured. If a field value in the suspect record does not match with the pivot, then the synonym values are used to find new matches rather than the initial value in the suspect field. The first positive match is used to get the final score.

Match inside list of values

- It is now possible to match within lists. Two lists of values match when at least one value is exactly the same in the two lists. For instance (John, Carl, Paul) will match with (Frank, Theo, John).

UI merge improvement

- A new button **Cluster view** allows one to get the list of the **future merged records** corresponding to the current selection for the merge.

New UI service

- **Show cluster for merge**. From the full matching view, a selected cluster containing a pivot record can now be loaded in the bottom panel of the view so that the **Merge** button is directly available.

31.72 **Release Note 1.3.1**

Release Date: January 3, 2014

Bug fixes

- **[9162]** After a match at once, the simple matching cannot launch a workflow until the user logs out.

31.73 **Release Note 1.3.0**

Release Date: December 17, 2013

New functionalities

- **Display metadata**. This service opens a pop-up with all the matching metadata of the current record.

- Metadata **Last process policy code** and **Last matching policy code** has been added.
- New partial merge options have been added: **Keep unmerged record(s) as suspect**, **Set golden and set unmerged record(s) as not suspect with**, **Set golden and set unmerged record(s) as suspicious**.
- **No merge** option has been added to cancel the merge procedure.
- New services have been addeed: **Exact match at once (to be matched)** and **Exact match at once (unmatched)**. These services apply a fast matching policy to group records in large set of data.
- New operation **Match suspicious** has been adeed to allow the matching of a suspicious record.
- New operation **Create new golden** has been added for pivot records. This service opens a pop-up to create a new record that is considered as the pivot. The former pivot record becomes a suspect and the merge view is displayed automatically. If all primary key fields are auto-incremented, then the new golden is created automatically and the merge view is displayed.

Enhancement of existing operations

- A score is now displayed in the simple matching view to get the matching results of a potential duplicate against a suspicious record.
- Progression bars are now displayed for the services match at once and set at once.
- During a merge, when a matching field does not match, its icon is displayed in red.

Configuration

Data life cycle context

The new property **dataset is applied to any data space** has been added.

Table

The new property **Disable matching trigger** deactivates the matching trigger and allows the direct use of the EBX® Match and Merge Add-on API. This implementation is useful when the EBX® Match and Merge Add-on triggers must coexist with other trigger actions, such as when integrated with the EBX® Insight Add-on.

Matching policy

The following new properties have been added to the matching policy:

- **Automatically create new golden**, to force the creation of a new golden record in the case of a match. This feature is only available if the system can automatically supply the primary key of the new record. That is, if an auto-incremented primary key is used.
- **No match records when same source**, to deactivate matching when two records come from the same source. The source is configured at the table level.

Process policy

The following has been added to the process policy:

- Property to configure whether or not EBX® permissions are applied to a merge performed through the user interface.

- Property to configure bespoke data views for **Full view matching up**, **Full view matching bottom**, **Light view matching**, **Merge view matching** and **Simple view matching**.
- Configuration of which operations the user can execute to manage a suspicious record in the service **Match suspicious**.
- Configuration of direct matching merge through the user interface, independant of the configuration of the user interface merge at submission. This allows specifying which operations the user can execute to manage the pivot and suspect records during the merge procedure.

API

New API has been added for **simulate match table**, which can be applied on either a record or a set of search criteria.

Other enhancements

- The EBX® Match and Merge Add-on operations available in EBX® views have been simplified. There are now only two operations, **full view** and **light view**. The **full view** operation is displayed at the table level to allow matching that is applied on all records. The **light view** operation is displayed at the record level. It is now possible to select a record and directly open the **light view** on the corresponding cluster.
- The tab **Create view** in **full view** and **light view** has been removed. Creation is now handled at the level of regular EBX® views. These views can be enriched with embedded inline matching at submission time to detect potential duplicate records on the fly, a feature already available in version 1.2.0.

31.74 **Release Note 1.2.1**

Release Date: September 19, 2013

Bug fixes

- Record creation is blocked when configuration is erroneous.
- Programmatic label renderers on foreign keys trigger an exception when used during matching operation.

31.75 **Release Note 1.2.0**

Release Date: September 17, 2013

New operations

- Embedded matching. This feature allows to launch the data stewardship after the submit button directly.
- Unmerge.
- Reinitialization of the latest cluster number.

Enhancement of existing operations

- Suspect record retention strategy can be configured. When the retention strategy is not applied, the suspect records that no longer match with the pivot record are pushed out the cluster.

- In the merge UI, the matching fields are indicated with a dedicated icon. A bubble help gives the name of the matching algorithm used.
- A matching policy context can now define a list of fields contexts (previously two fields were available only).

Configuration

- A survivorship policy can be configured by contexts of dataspace and dataset.
- A matching policy can be configured by contexts of dataspace and dataset.
- A matching policy can be configured with matching thresholds that overwrite ones stemming from the process policy.
- A matching field can be configured to exclude records.
- A matching field can be configured to filter field values before the match execution.

Other

- The add-on runs on the tables that are persisted in the relational mode. Some limitations must be taken into consideration compared to the semantic mode.
- Doublemetaphone algorithm has been improved in order to produce matching scores more accurate.

31.76 **Release Note 1.1.0**

Release Date: June 3, 2013

New operations

- Group at once (unmatched) and group at once (to be matched). These operations are used to group 'unmatched' or 'to be matched' records in order to apply the match operation on the scope of a group. Two operations group collapse (unmatched) and group collapse (to be matched) are available to undo the groups.
- Match at once (suspicious).
- **Set definitive golden** on suspicious and unmatched record.
- **Add into cluster** on a suspicious record.
- **Not suspect with** to decide that a record is no longer a suspect against a pivot record until a new scoring reaches a certain level of similarity or until the property **On not suspect with** is set to **False** in the related process policy.

Enhancement of existing operations

- Match at once on the scope of groups of records collected by the operations group at once (unmatched) and group at once (to be matched).
- **Add into cluster** allows to move the record into a new cluster (creation of the cluster).

Configuration

- A matching policy can be configured for the operations match at once.
- A matching policy can be configured for the operations group at once.
- A matching policy can be configured for a workflow context.
- A matching policy can be configured with fields values that exclude records from the matching.

Matching algorithm and scoring

- Funneling matching.
- New matching algorithms (DoubleMetaPhoneLevenshtein, Soundex).
- **Golden preserved for selection** is available. Then, the **pivot record selection mode** will keep in priority a **was golden** record if it exists in the cluster.
- New matching algorithm **Exact** that allows to force an exact matching (not fuzzy).
- Matching policy with Exact literal score. When more than one Business ID fields is configured, two records are considered as Suspect if any of the Business ID field value is equal (in the previous version, two records are considered as Suspect if all of the business ID fields values are equal).
- When a survivor field function returns more than one value, the record with the latest timestamp is used in priority (in the previous version it is a random selection).

Workflow

- Script: change state of a record to suspicious.
- Script: simulate the match table operation. The result is a list of all records that match but without changing their states.
- Condition: check if a record is in a given state.
- User task: the existing Simple Matching user task is enriched with several new properties (see Java doc directly).
- User task: merge view.

API

- Java API are now available to integrate the EBX® Match and Merge Add-on with external systems.

31.77 Release Note 1.0.0

Release Date: April 8, 2013

EBX® Match and Merge Add-on finds duplicate records in a table and its relations. Based on a user configuration of fields to match in a table, the add-on automatically identifies which records are suspected duplicates.

A rich EBX®-integrated end-user interface provides all the services required to govern duplicate records. Users can reject or merge duplicate records in order to build an accurate record, also known as a golden record.

31.78 Known limitations

The EBX® session automatically times out for security reasons, meaning the user is logged out after a certain period of inactivity. In some situations, this can result in the EBX® login screen opening inside a frame of the EBX® Match and Merge Add-on views. To rectify this situation, the user must select any action outside the frame to display the full EBX® login screen. This limitation has no impact on data integrity or security, however, it is a known ergonomics issue.

The EBX® Match and Merge Add-on is currently only available in English. The French translation will be added in a future version.