STIBO SYSTEMS
MASTER DATA MANAGEMENT

USER GUIDE

Matching, Linking, and Merging

2024.1 – March 2024

Table of Contents

Matching, Linking, and Merging

The STEP Matching, Linking, and Merging component offers powerful functionality for identifying and handling duplicate product, entity, asset, and classification objects in STEP.

The matching, linking, and merging functionality is most commonly used for:

- Cleanup operations, such as during data migration
- Matching of the same product from multiple suppliers
- Matching of the same customer from different source systems
- Consolidation of information from different systems
- Cleansing data after migrating records from various sources

As defined in the following sections, before configuring the functionality, users must:

- Determine the match criteria to define what qualifies two or more objects as duplicates.
- Choose a match action to define what the system should do when it encounters such duplicates.
- Perform match tuning to match as expected and optimize performance.

Determine Match Criteria

As part of a matching algorithm, the match criteria determine what qualifies objects as duplicates, as defined in the **Match Criteria** topic.

Choose a Match Action

As part of a matching algorithm, when setting up a matching solution, users must choose a match action to determine what the system does with duplicates. The match action defines the workflow and the data model around the objects you are matching. Available match actions are included on the **Match Actions** topic.

Identify Duplicates

Users can configure the system to only identify duplicates using the Identify Duplicates match action, or to also act on those matches. The system supports different action strategies like merging records or generating new Link Golden Records. For more information, refer to the **Identify Duplicates** topic.

Match and Link

Match and Link creates and maintains a set of Golden Records as an aggregation of matching Source Records through an asynchronous process.

- In Product MDM, Match and Link automates the creation and maintenance of Sell-Side Products as Golden Records, based on Buy-Side Products as Source Records.
- In Customer MDM, Match and Link resolves Household Entities as Golden Records from Individual Customer Entities as Source Records.

Match and Link uses an event processor to create and update new Link Golden Records that captures the best information from each of the Source Records. The system identifies the new Link Golden Record object with a STEP identifier and links this record to all source records contributing to it. Over time, new information may clarify that some source records that were linked together are no longer valid for linking to a specific Golden Record. The algorithm will then link these Source Records to different Link Golden Records. As a result of this automatic linking and splitting, the STEP identifier of the Link Golden Record linked to a given Source Record may change over time.

Users should **never** edit a Link Golden Record object directly. To edit a Link Golden Record object, users should add the information on a special type of source record, called a 'Silver Record,' and the information is then merged into the Link Golden Record by the matching algorithm. The promotion of information from the Silver Record to the Link Golden Record happens asynchronously through the Matching Event Processor. For more information, refer to the **Match and Link** topic.

Match and Merge

The Match and Merge solution uses criteria to match entity records and merge these incoming records into Golden Records.

In Customer MDM and Supplier MDM, Match and Merge is used to consolidate, enrich, and synchronize duplicate records in surrounding systems.

Note: The Match and Merge solution only works for entities.

Match and Merge works by combining a special importer and an event processor. When the Match and Merge Importer imports a new entity, the importer uses a matching algorithm to compare the incoming entity against an existing Golden Record. If a matching entity already exists, the system promotes the information from the incoming entity to that existing Golden Record through Survivorship Rules. As Golden Records are updated, a Matching Event Processor identifies matching Golden Records and merges information from one of the records into the other and deactivates the non-survivor. For more information, refer to **Match and Merge** topic.

Perform Match Tuning

Defining match criteria that accurately identifies matching records is an iterative process that requires a thorough understanding of the data and collaboration between data owners and the super users defining the match criteria. Match tuning enables you to refine and optimize the matching process.

- **During initial implementation**: tune the match criteria to match the correct records and potentially optimize your match criteria to achieve your performance goals.

- **On a live system**: over the lifetime of the system, the structure of data input can change, or new source systems are introduced. As new insights and/or requirements occur, updates to the matching algorithm match codes, match criteria, and match action are required and are also classified as match tuning.

Important: To ensure the most up-to-date algorithm is applied on all relevant objects, when changing a matching algorithm also republish all objects for the algorithm in the event processor.

For more information about the tools available and the recommended process, refer to the **Match Tuning** topic.

Match Criteria

Within a Matching Algorithm, the match criteria are responsible for matching records against each other to find those that match. When users are only interested in exact matches, the match criteria are reasonably straightforward.

For example, if the SSN (Social Security Number) for two customer objects or the EAN (European Article Number) for two product objects are identical, the records are likely duplicates and the matching criteria should return 100 percent. If the SSN or EAN does not match, the match criteria should probably return 0 percent.

In many cases you cannot work with exact matches; instead, you will deal with approximate matches or a combination of exact and approximate matches. For example, for a customer you do not have a SSN available so you will identify duplicates based on names, mailing addresses, phone numbers, and street addresses. For a product, you will identify duplicates based on the manufacturer and manufacturer part number.

This data can have variations, even in objects that represent the same real-world item. Names and addresses can be spelled differently, middle names could be omitted, abbreviations can be used in names and addresses, the customers could be registered with different phone numbers or mailing addresses, and other options that introduce ambiguity to the records.

This complexity can be handled via a decision table in the match criteria logic, which further divides the functionality into normalizers, matchers, and rules.

The Match Criteria uses a decision table to define how to compare two objects and evaluate to what degree they are similar by producing a match score. For more information, refer to the **Match Scores** topic.

Creating Match Criteria

Match Criteria is comprised of Data Elements, Matchers, Rules, Match Code Generators, and Match Code Filters for a matching algorithm. All are added and configured on the Decision Table dialog.

To create match criteria:

1. Edit the match criteria based on the type of matching algorithm:

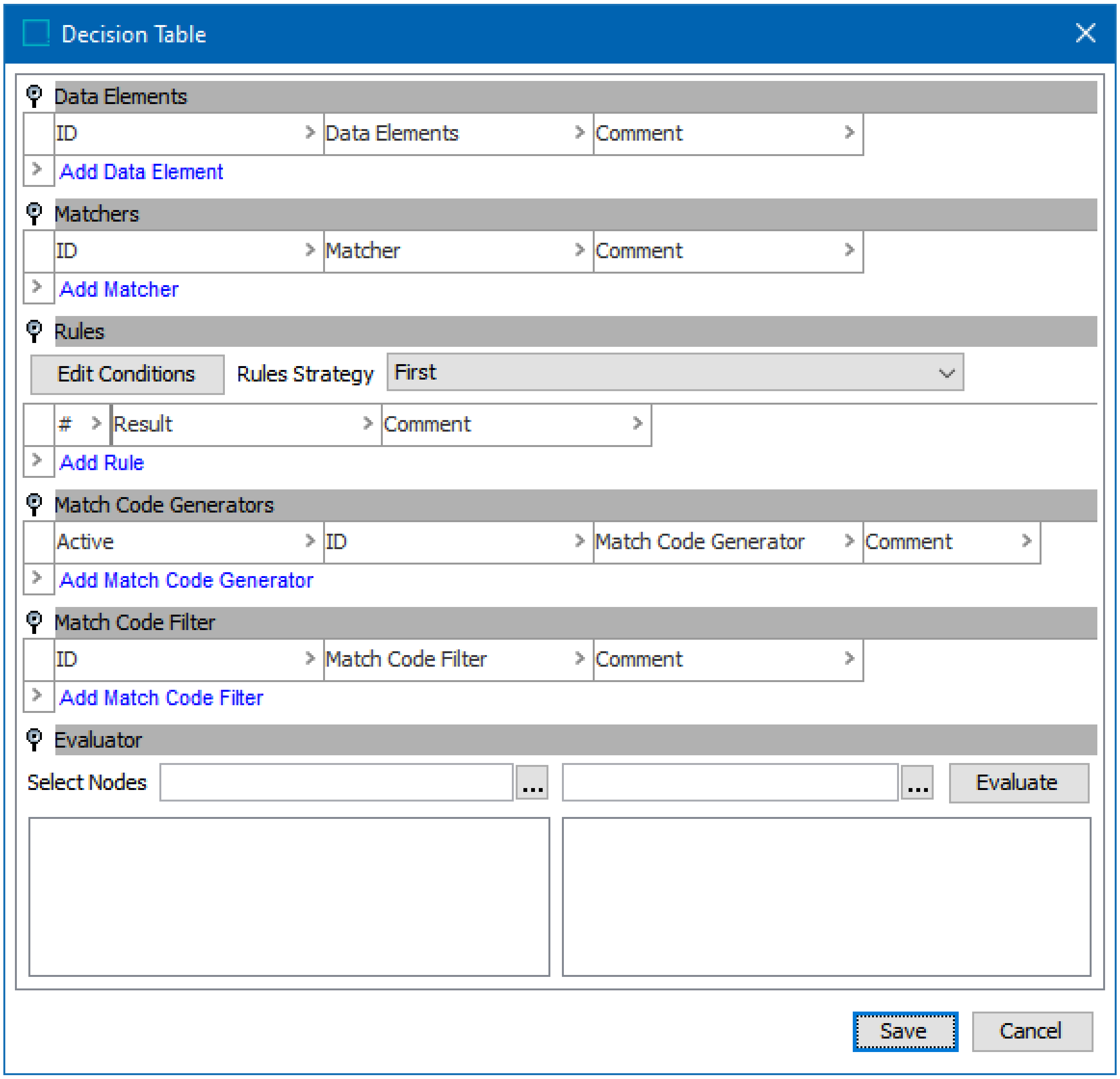
   With embedded match codes: select the Match Criteria tab and click the **Edit Match Criteria** link to display the Decision Table dialog.

   Without embedded match codes (This is a legacy matching algorithm type and has no Match Criteria tab.): on the Matching Algorithm tab open the Match Action flipper, click the **Add Criterion** link, add an ID, select **Decision Table** from the dropdown and click the **Add** button.

   - To edit an existing Decision Table criterion row, click the ellipsis button (...) for the criterion to display the Decision Table dialog.
   - To create a new match criteria click the **Add Criterion** link.

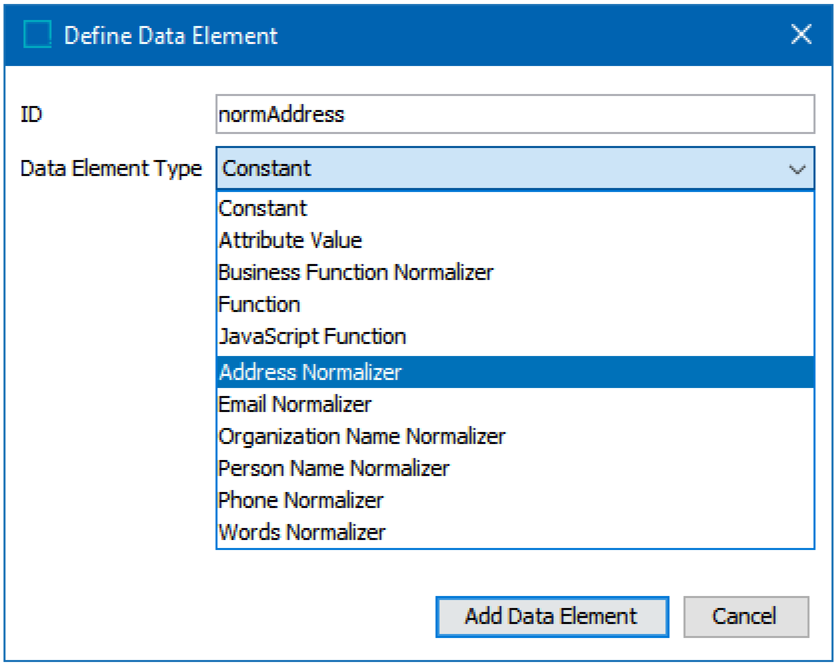
2. In the Decision Table dialog, for each of the following flippers, add one or more rows, and then configure the new row(s) as required:

> **Important:** IDs must be unique across the data elements flipper, the matchers flipper, and the match code generators flipper on the Decision Table dialog.



Rules Strategy: First

- **Data Elements** declare the input for the matchers and match code generators and allow data to be normalized to a format that is easy to compare. Refer to the **Match Criteria Data Elements** topic.

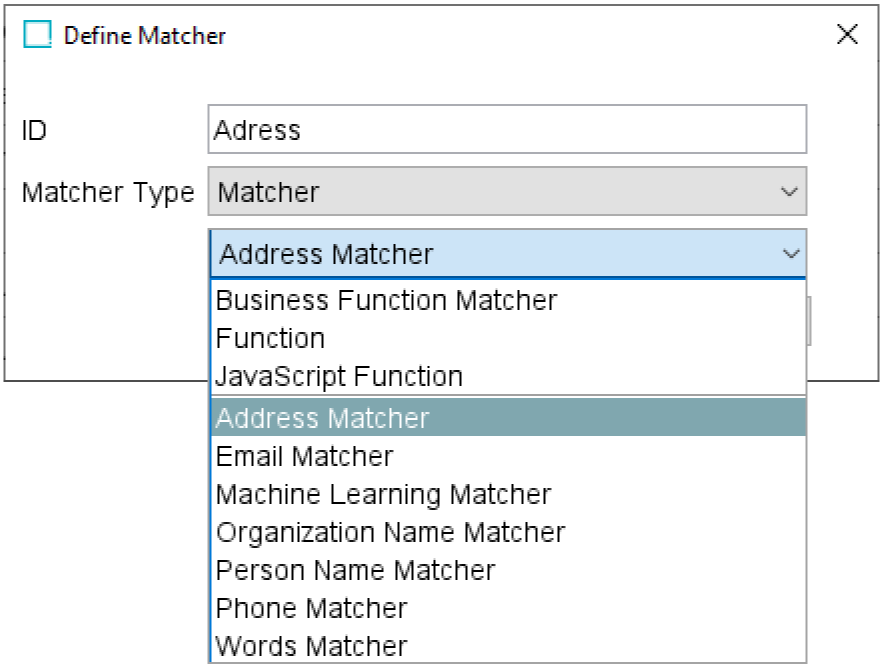


Define Data Element

ID: normAddress

Data Element Type: Constant

- **Matchers** do the actual comparisons of values from the specified data element. A matcher compares one logical aspect of the objects, assigning an equality percentage to that aspect based on the related values. Refer to the **Match Criteria Matchers** topic.



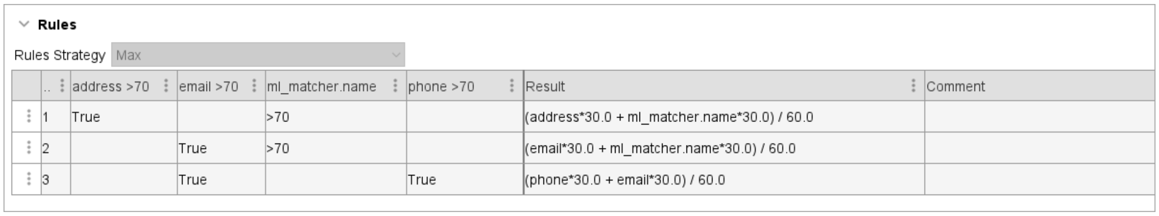
Define Matcher

ID: Adress

Matcher Type: Matcher

Address Matcher

- **Rules** combine the results of matchers into a final match score, which is a percentage that signifies if two objects are a match or are not a match. A new row is added to the flipper and can be configured as needed. Refer to the **Match Criteria Rules** topic.

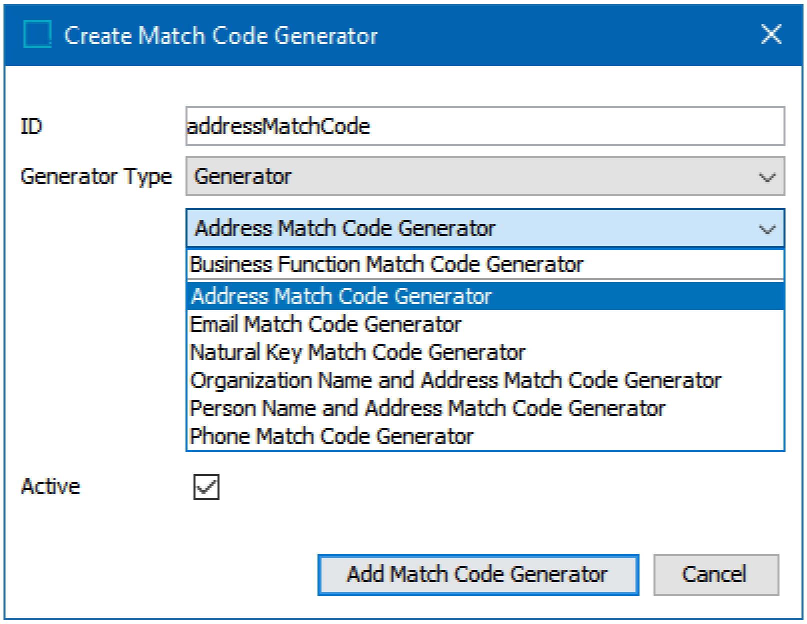


Rules

Rules Strategy: Max

..	address >70	email >70	ml_matcher.name	phone >70	Result	Comment
1	True		>70		(address*30.0 + ml_matcher.name*30.0) / 60.0	
2		True	>70		(email*30.0 + ml_matcher.name*30.0) / 60.0	
3		True		True	(phone*30.0 + email*30.0) / 60.0	

- **Match Code Generators** identify the records that should be compared. Only records with at least one equal match code are passed through the match criteria for evaluation of a match score. This allows efficient matching on a dataset of millions of objects because it prevents comparing every object with every other object. Refer to the **Match Criteria Match Code Generators** topic.



Create Match Code Generator

ID: addressMatchCode

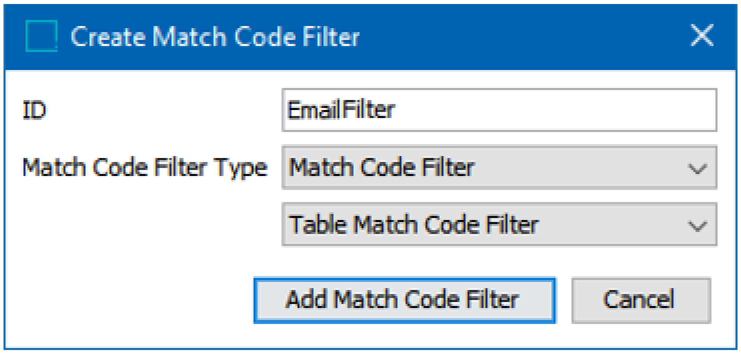
Generator Type: Generator

Address Match Code Generator

Business Function Match Code Generator
Address Match Code Generator
Email Match Code Generator
Natural Key Match Code Generator
Organization Name and Address Match Code Generator
Person Name and Address Match Code Generator
Phone Match Code Generator
Add Match Code Generator
Cancel

Active ☑

- **Match Code Filter** allows users to remove specific match code values based on data exceptions defined in a Transformation Lookup Table. Refer to the **Match Criteria Match Code Filter** topic.



Create Match Code Filter

ID: EmailFilter

Match Code Filter Type: Match Code Filter

Table Match Code Filter

3. In the Decision Table dialog, open the Evaluator flipper and test the configuration.

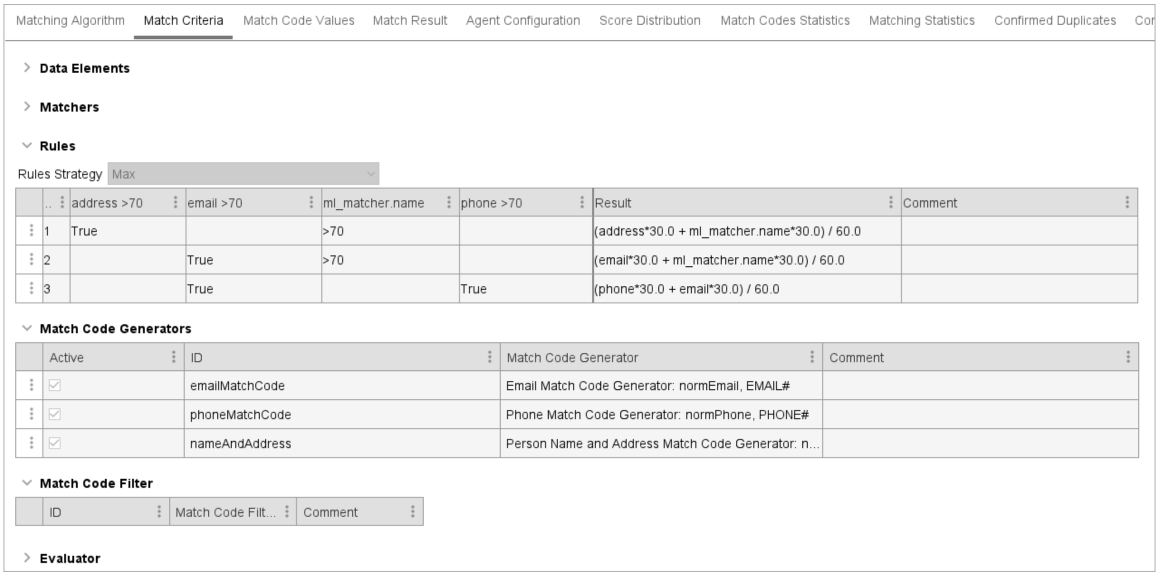
   - For the Select Nodes parameters, click the ellipsis button (...) for each field and select two objects for comparison.

- Click the **Evaluate** button.

An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

4. Click the **Save** button to keep the Match Criteria changes and return to the Matching Algorithm object.

An example set of match criteria elements is shown below.



Rules Strategy: Max

..	address >70	email >70	ml_matcher.name	phone >70	Result	Comment
1	True		>70		(address*30.0 + ml_matcher.name*30.0) / 60.0	
2		True	>70		(email*30.0 + ml_matcher.name*30.0) / 60.0	
3		True		True	(phone*30.0 + email*30.0) / 60.0	

Active	ID	Match Code Generator	Comment
☑	emailMatchCode	Email Match Code Generator: normEmail, EMAIL#	
☑	phoneMatchCode	Phone Match Code Generator: normPhone, PHONE#	
☑	nameAndAddress	Person Name and Address Match Code Generator: n...	

ID	Match Code Filt...	Comment

Matching Algorithms and Match Expressions

Matching is performed by a matching algorithm and can involve the following elements, referred to as 'match expressions':

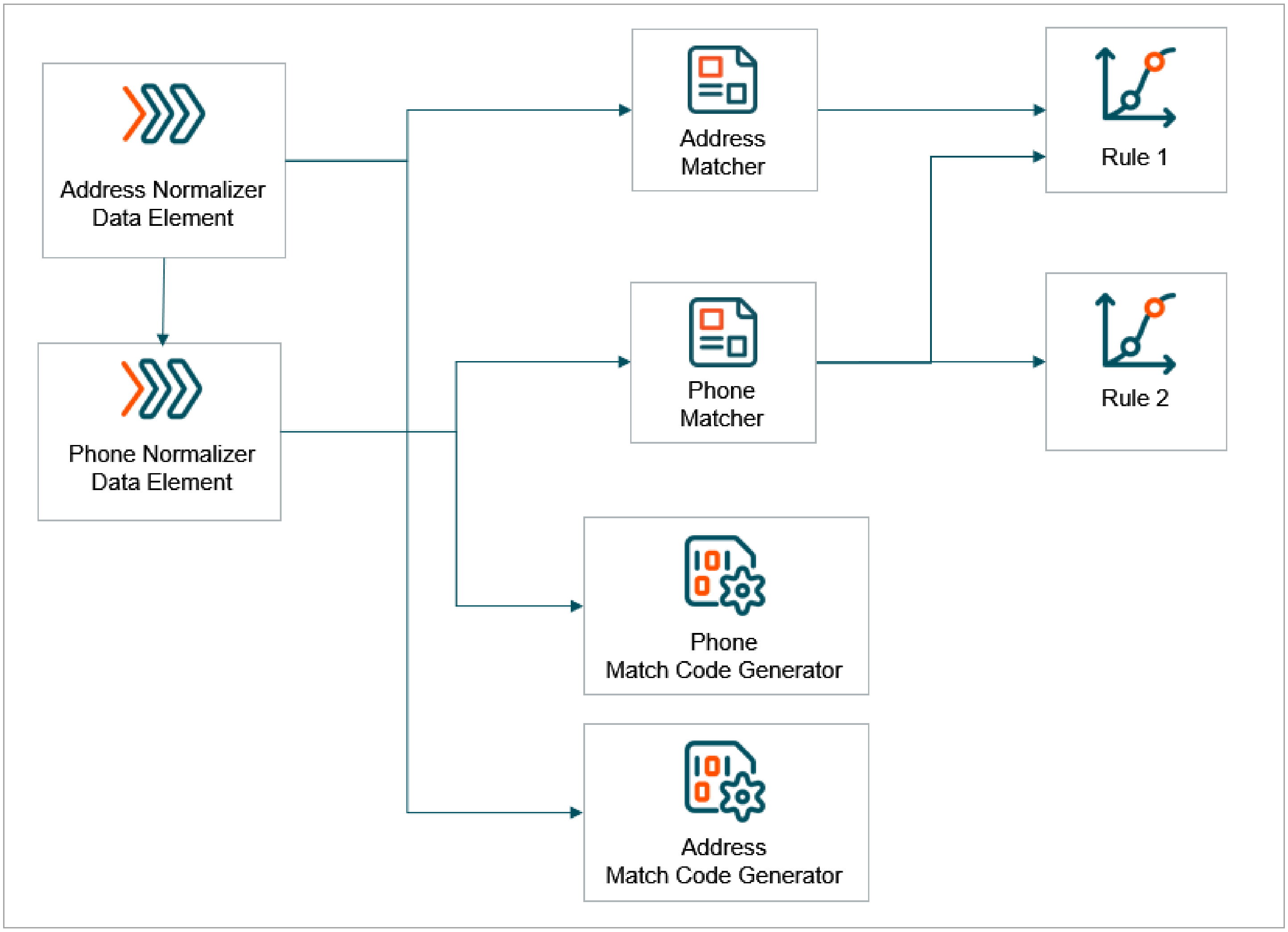
- Global Binds (legacy functionality, use data elements instead)
- Data Elements, with subtypes like the Address Normalizer
- Matchers, with subtypes like the Organization Name and Address Matcher
- Match Code Generators
- Match Code Filters

Match expressions can be thought of as 'functions' available in the match expression context, and each must have a unique user-defined ID.

The 'match expression context' (illustrated below) includes all match expressions identified in the matching algorithm. When a matcher needs the output from a data element (or a legacy global bind), it uses the relevant ID and calls the match expression context to evaluate the match expression.

- If the specified data element was already evaluated, the result is returned.
- If the data element has not yet been evaluated, the context evaluates it.

This means the result of a Data Element is available to a Matcher by the call to **MatchExpressionContext evaluate(dataElementID)**, and that when normalizing the data element, complex computations are performed only one time.


Address Normalizer Data Element
Phone Normalizer Data Element
Address Matcher
Phone Matcher
Phone Match Code Generator
Address Match Code Generator
Rule 1
Rule 2

The dependency graph between elements in the Decision Table should be forthgoing, that is:

- Global binds should not evaluate other match expressions. Do not use calculated attributes with match expression evaluations as global binds.
- Data elements may evaluate other data elements or global binds.
- Matchers and Match Code Generators may evaluate global binds, data elements, and expressions of their own kind.
- Match code filters may evaluate global binds, data elements, match codes, and other match code filters.

The **MatchExpressionContext evaluate()** function exists in these permutations, where the **contextObject** is either 'first' or 'second':

- **evaluate(expressionID) -** Data Elements have two outputs, one for the first object, another for the second. Evaluation of a data element normally concerns only the normalization of the value, and as such, typically uses this method. When evaluating the first object, it returns the data elements or bind variables related to that first object, disregarding any values from second object.

- **evaluate(expressionID, contextObject) -** Matchers have one output, which is produced by comparing values on the first object with values on the second object. For that reason, the matchers use this method, retrieving the values of both objects, comparing, and returning a result from that comparison.

  For more information, refer to the **Current Object Bind** topic and the **Secondary Object Bind** topic in the **Resource Materials** documentation.

> **Note:** The MatchExpressionContext 'evaluate' function ignores types until runtime. This means that type inconsistencies are only discovered at runtime. When working with **evaluate()**, it is recommended to do small changes in iterations, and test often.

Chaining Match Expressions to Expand Functionality

Chaining match expressions allows individual expressions to run in a defined order to produce the necessary output, which is then evaluated by the next expression in line. Consider the following examples of chaining:

- Business Function runs before an Address Normalizer
- Address Normalizer runs before a Business Function

Business Function runs before an Address Normalizer

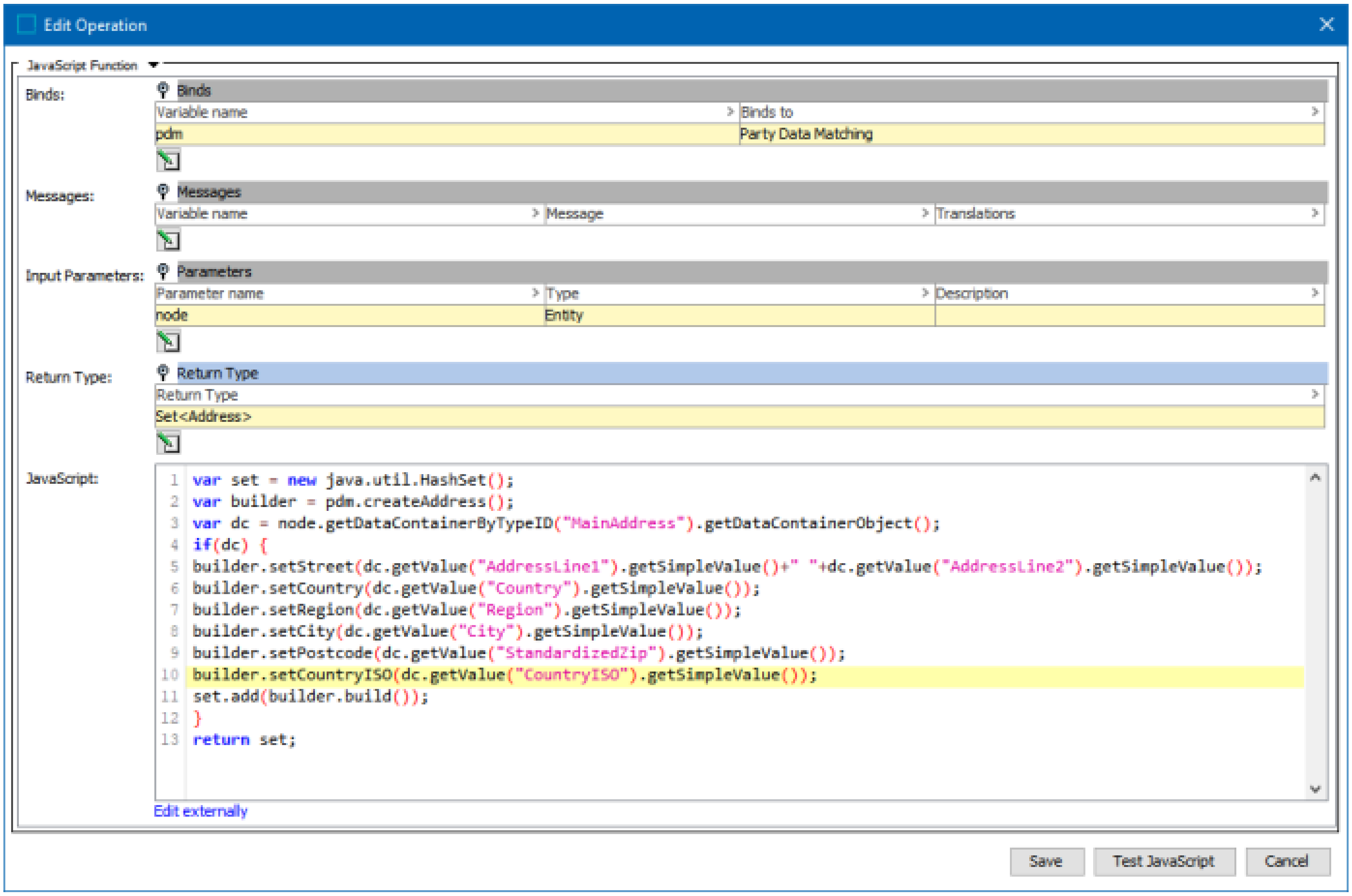
Normalizer output can be expanded or entirely replaced with a JavaScript Business Function run from a Business Function Normalizer. For example, chaining an Address Normalizer with Business Function Normalization allows the business function to run first and that output is used to create input for an address normalizer.

This setup includes the following elements which are illustrated below:

- Create a JavaScript business function.
- Create and configure a Business Function Normalizer data element
- Create and configure an Address Normalizer

1. Create a JavaScript business function.

   This sample custom business function provides additional normalization to the Address Normalizer functionality.



Binds:

Variable name	Binds to
pdm	Party Data Matching

Messages:

Variable name	Message	Translations

Input Parameters:

Parameter name	Type	Description
node	Entity	

Return Type:

Return Type
Set<Address>

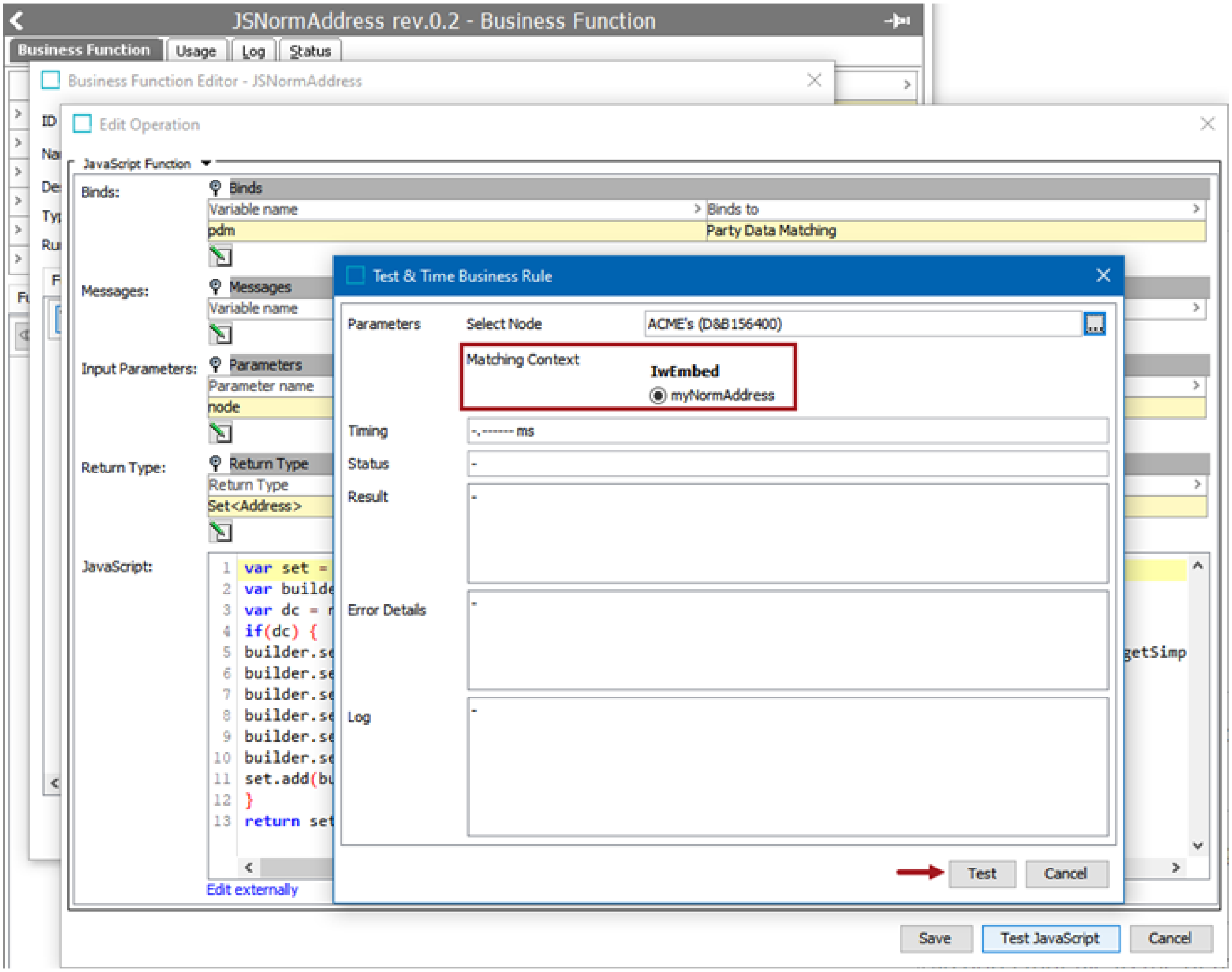
JavaScript:

```
var set = new java.util.HashSet();
var builder = pdm.createAddress();
var dc = node.getDataContainerByTypeID("MainAddress").getDataContainerObject();
if(dc) {
builder.setStreet(dc.getValue("AddressLine1").getSimpleValue()+" "+dc.getValue("AddressLine2").getSimpleValue());
builder.setCountry(dc.getValue("Country").getSimpleValue());
builder.setRegion(dc.getValue("Region").getSimpleValue());
builder.setCity(dc.getValue("City").getSimpleValue());
builder.setPostcode(dc.getValue("StandardizedZip").getSimpleValue());
builder.setCountryISO(dc.getValue("CountryISO").getSimpleValue());
set.add(builder.build());
}
return set;
```

```
        var set = new java.util.HashSet();
        var builder = pdm.createAddress();
        var dc = node.getDataContainerByTypeID("MainAddress").getDataContainerObject();
        if(dc) {
        builder.setStreet(dc.getValue("AddressLine1").getSimpleValue()+" "+dc.getValue
    ("AddressLine2").getSimpleValue());
        builder.setCountry(dc.getValue("Country").getSimpleValue());
        builder.setRegion(dc.getValue("Region").getSimpleValue());
        builder.setCity(dc.getValue("City").getSimpleValue());
        builder.setPostcode(dc.getValue("StandardizedZip").getSimpleValue());
        builder.setCountryISO(dc.getValue("CountryISO").getSimpleValue());
        set.add(builder.build());
        }
        return set;
```

2. Close the JavaScript business function to save it and reopen to test it.

> **Note:** All changes for a JavaScript business function that includes the 'Matching Context' parameter are applied when the business function is closed and reopened.



Binds: Variable name: pdm | Binds to: Party Data Matching

Input Parameters: Parameter name: node

Return Type: Set<Address>

```
var set =
var builde
var dc = r
if(dc) {
builder.se
builder.se
builder.se
builder.se
builder.se
builder.se
set.add(bu
}
return set
```

getSimp

Parameters: Select Node: ACME's (D&B156400)

Matching Context: IwEmbed ● myNormAddress

Timing: -,------ ms

Status: -

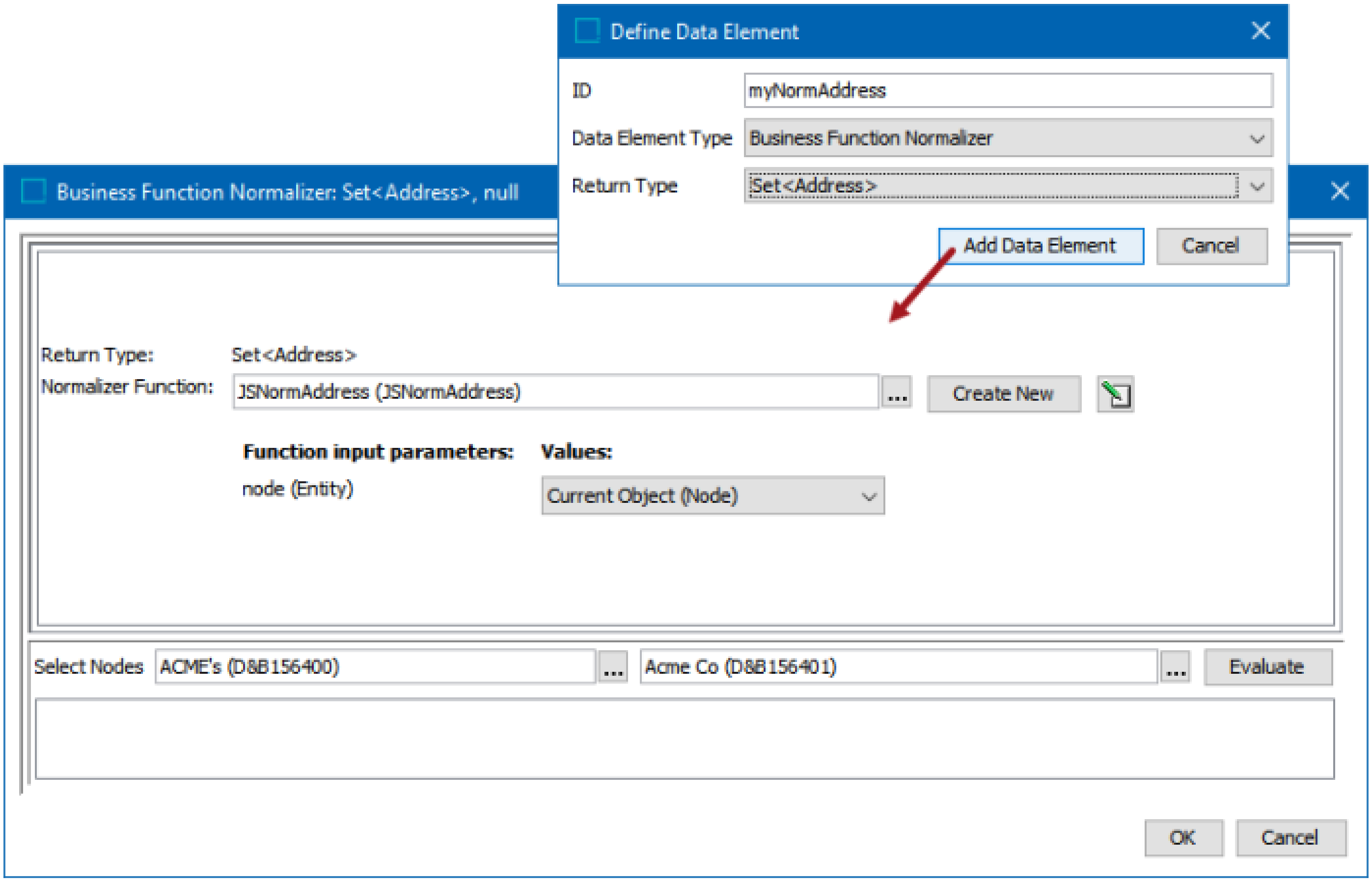
Result: -

Error Details: -

Log: -

3. Create and configure a Business Function Normalizer data element.

The Business Function Normalizer data element type with a Return Type of Set <Address> is created and links the selected 'Function input parameters' and 'Values'.



Define Data Element

ID: myNormAddress

Data Element Type: Business Function Normalizer

Return Type: Set<Address>

Business Function Normalizer: Set<Address>, null

Return Type: Set<Address>

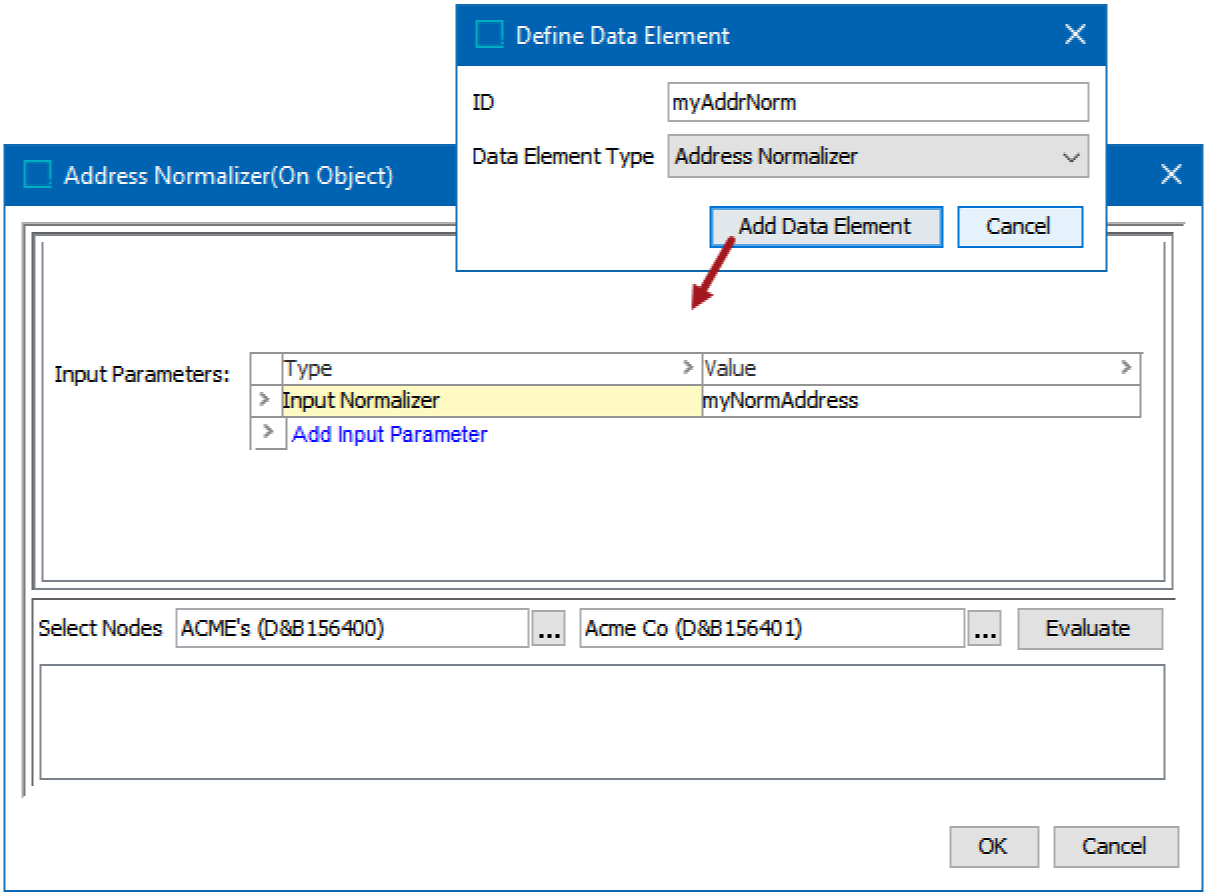
Normalizer Function: JSNormAddress (JSNormAddress)

Function input parameters:	Values:
node (Entity)	Current Object (Node)

Select Nodes: ACME's (D&B156400) | Acme Co (D&B156401)

4. Create and configure an Address Normalizer data element.

   The Address Normalizer is configured with the Input Normalizer type to use the Business Function Normalizer output.



ID: myAddrNorm

Data Element Type: Address Normalizer

Input Parameters:

Type	Value
Input Normalizer	myNormAddress

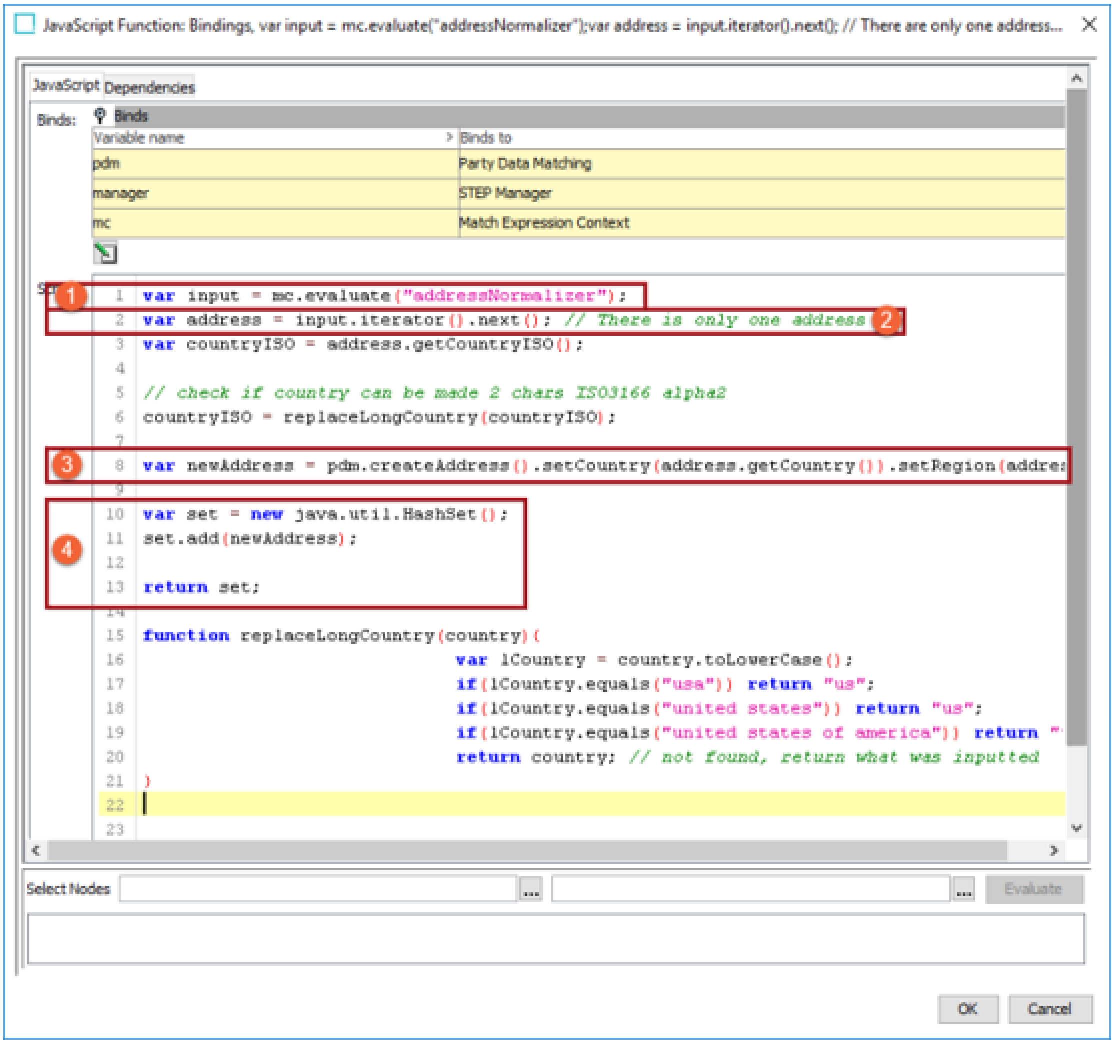
Select Nodes: ACME's (D&B156400) | Acme Co (D&B156401)

Address Normalizer runs before a Business Function

Normalizer output can be expanded or entirely replaced with a JavaScript Function. For example, chaining a JavaScript Function to run after an Address Normalizer, using the output of the standard normalizer as its input.

This setup includes the following elements which are illustrated below:

1. Uses the evaluate function on a Match Expression Context, 'mc.evaluate' in the screenshot below, to retrieve the output of a desired normalizer.
2. Uses an iterator to access the set of values / strings.
3. Uses a builder pattern to create new values / strings from the iterated data.
4. Inserts the new values / strings into the return set.


JavaScript Function: Bindings, var input = mc.evaluate("addressNormalizer");var address = input.iterator().next(); // There are only one address...
JavaScript
Dependencies
Binds:
Binds
1
2
3
4
Select Nodes
...
...
Evaluate
OK
Cancel

Variable name	Binds to
pdm	Party Data Matching
manager	STEP Manager
mc	Match Expression Context

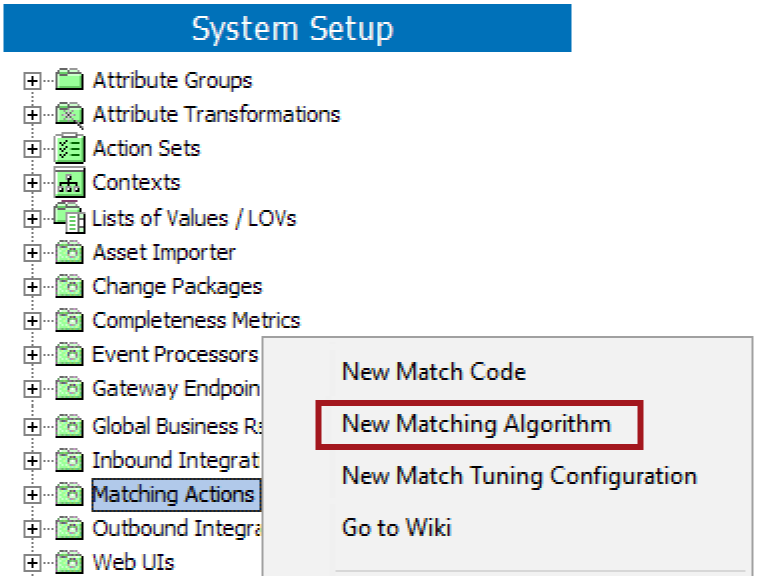
```
var input = mc.evaluate("addressNormalizer");
var address = input.iterator().next(); // There is only one address
var countryISO = address.getCountryISO();

// check if country can be made 2 chars ISO3166 alpha2
countryISO = replaceLongCountry(countryISO);

var newAddress = pdm.createAddress().setCountry(address.getCountry()).setRegion(addre:

var set = new java.util.HashSet();
set.add(newAddress);

return set;

function replaceLongCountry(country){
                                var lCountry = country.toLowerCase();
                                if(lCountry.equals("usa")) return "us";
                                if(lCountry.equals("united states")) return "us";
                                if(lCountry.equals("united states of america")) return "
                                return country; // not found, return what was inputted
}

```

Initial Setup for Matching Algorithms

This one-time setup is required to define a matching algorithm group type which then holds the matching algorithms you create for use in match, link, and merge solutions.

The matching algorithm is typically configured first, followed by the match codes.

Review your System Setup tab to determine if a matching algorithm node already exists. Right-click on the node and verify that the 'New Matching Algorithm' option is enabled. The name of the node on your system is not required to match the one in the image below.



If you do not have a node to hold matching algorithms, complete the following one-time setup steps.

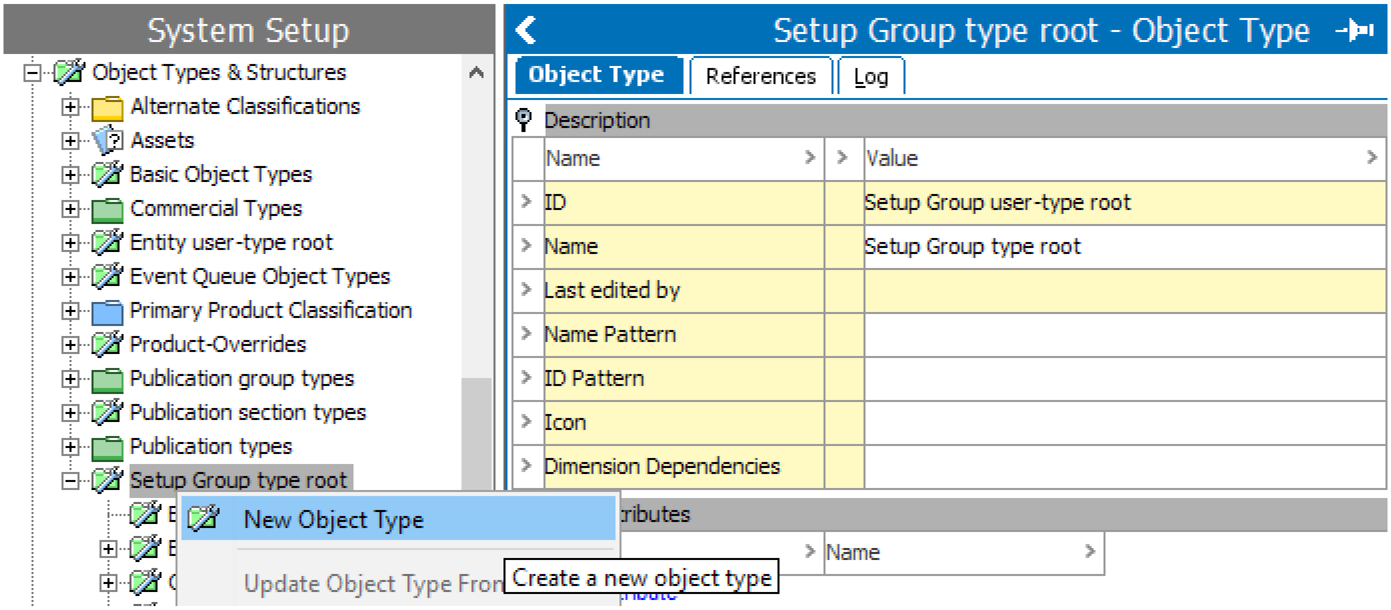
1. Create setup group type for matching algorithms.
2. Link matching algorithm object types to setup group type.
3. Create a matching algorithm setup group.

Once the setup has been completed, the steps in this section are only needed if you want additional levels of organization.

Create Setup Group Type for Matching Algorithms

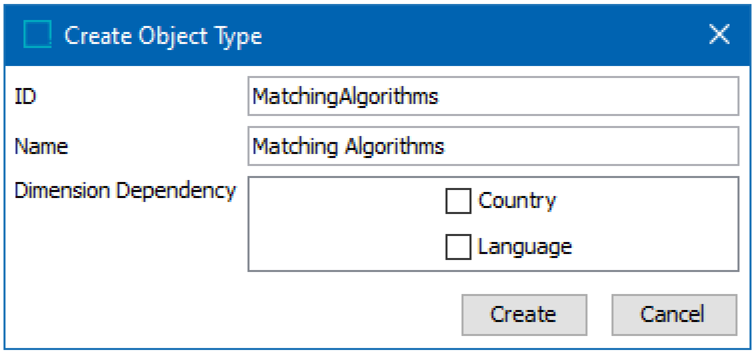
A matching algorithm group type defines the structure and allowed locations of a matching algorithm.

1. Go to System Setup > Object Types & Structures > select **Setup Group type root**.
2. Right-click **Setup Group type root**, and the New Object Type option will display.



Name	Value
ID	Setup Group user-type root
Name	Setup Group type root
Last edited by	
Name Pattern	
ID Pattern	
Icon	
Dimension Dependencies	

3. Click **New Object Type**, and the Create Object Type dialog will display.



ID: MatchingAlgorithms

Name: Matching Algorithms

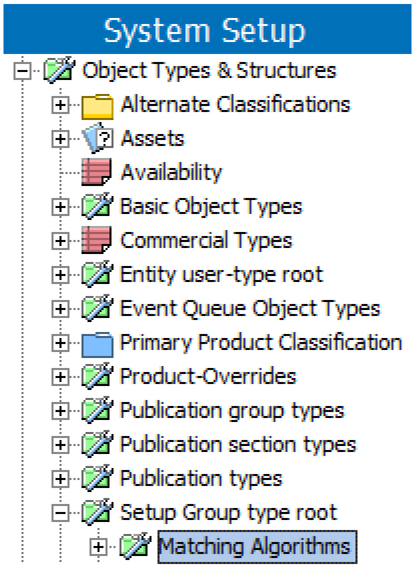
Dimension Dependency: ☐ Country ☐ Language

4. Enter an **ID**.

5. Enter a **Name**.

6. Click **Create**.

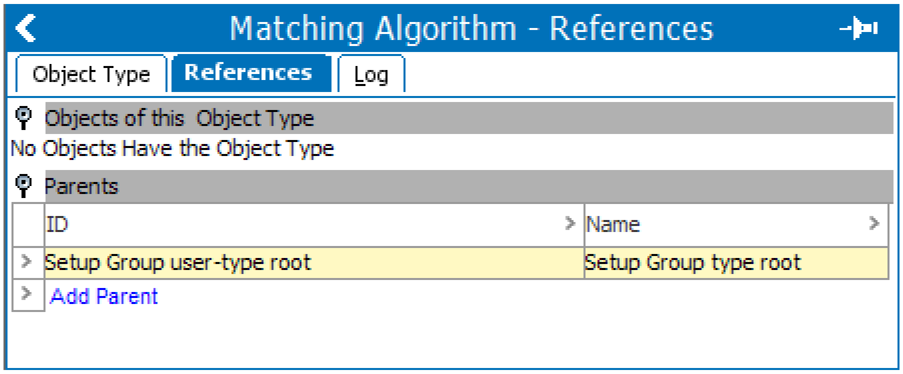
   The Create Object Type dialog closes, and the newly created object type for the matching algorithm displays beneath the Setup Group type root.



System Setup

- Object Types & Structures
  - Alternate Classifications
  - Assets
  - Availability
  - Basic Object Types
  - Commercial Types
  - Entity user-type root
  - Event Queue Object Types
  - Primary Product Classification
  - Product-Overrides
  - Publication group types
  - Publication section types
  - Publication types
  - Setup Group type root
    - Matching Algorithms

7. Select the newly added Setup Group type > References tab > open the Parents flipper.



Matching Algorithm - References

Objects of this Object Type

No Objects Have the Object Type

Parents

ID	Name
Setup Group user-type root	Setup Group type root
Add Parent	

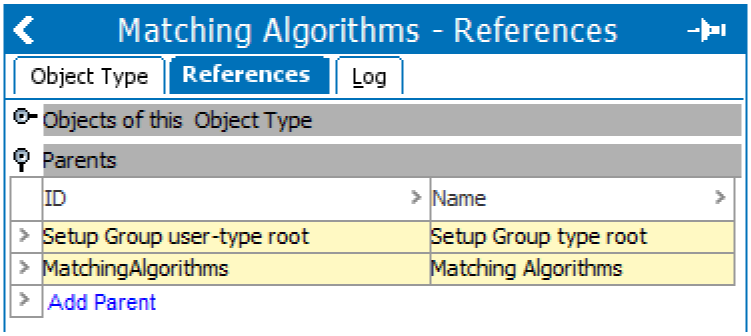
Important: By default, the Setup Group type root is listed as the parent. Optionally add the newly created setup group type as a parent of itself so that additional matching algorithm group types can be added below the main level.

8. Click **Add Parent**, and the Select New parent dialog displays.

9. Browse or search to select **the relevant setup group type**.

10. Click the **Select** button.

    The dialog will close, and the newly created setup group type (i.e., Matching Algorithms) is listed as a parent along with the Setup group user-type root.

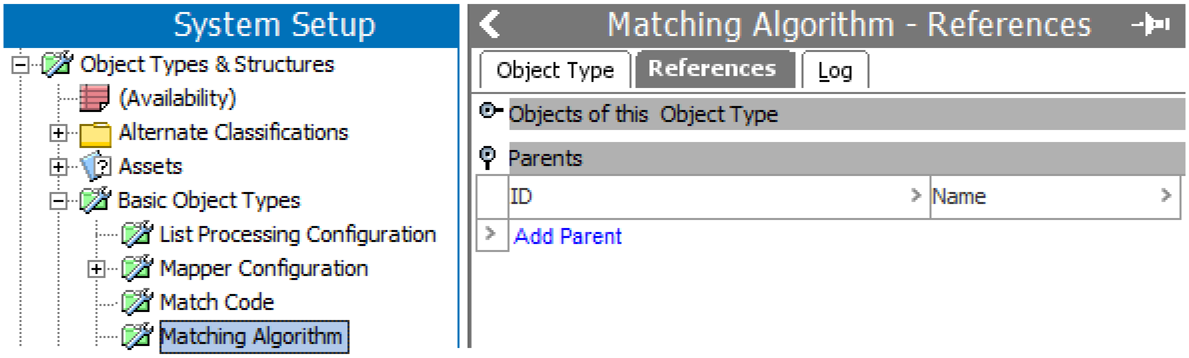


ID	Name
Setup Group user-type root	Setup Group type root
MatchingAlgorithms	Matching Algorithms

Link Matching Algorithms Object Types to Setup Group Types

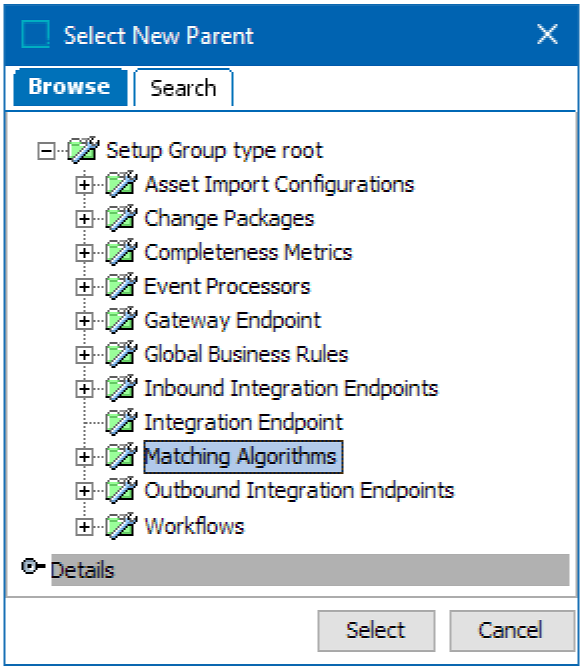
Linking determines the object types that can be displayed at each level of a hierarchy.

1. Go to System Setup> Object Types & Structures > **Basic Object Types**.
2. Select **your matching algorithm object type** to display the editor.



3. Click the **References** tab.
4. Open the **Parents** flipper.
5. Click the **Add Parent** link, and the Select New Parent dialog displays.
6. Browse or search to select **the relevant setup group type**.

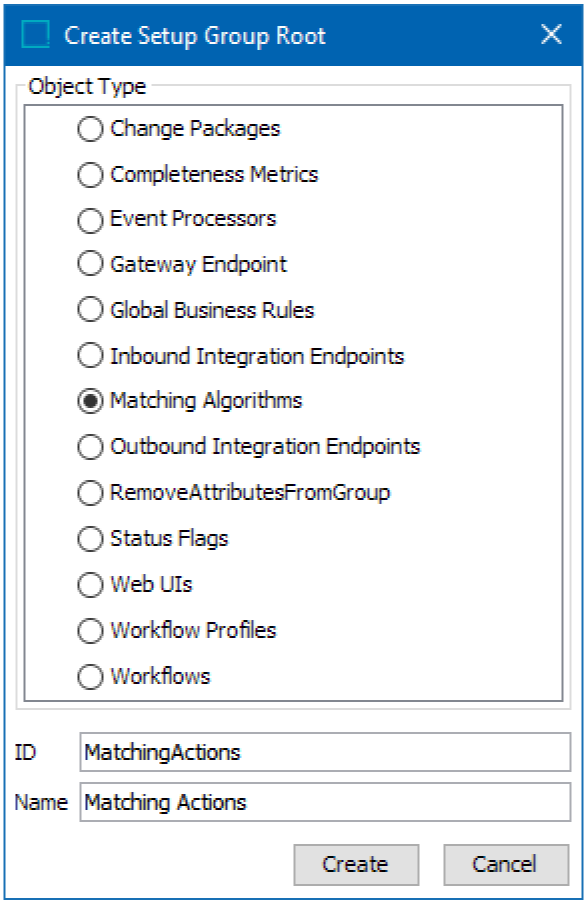
7. Click the **Select** button.



Create a Matching Algorithm Setup Group

Creating a setup group allows your matching items (including a matching algorithm setup group type) to appear as a node in the System Setup hierarchy.

1. Go to System Setup > select **any object in the hierarchy**.
2. On the menu bar, select **Maintain** > **Insert** > **Setup Group Root**, and the Create Setup Group Root dialog will display.



Create Setup Group Root

Object Type

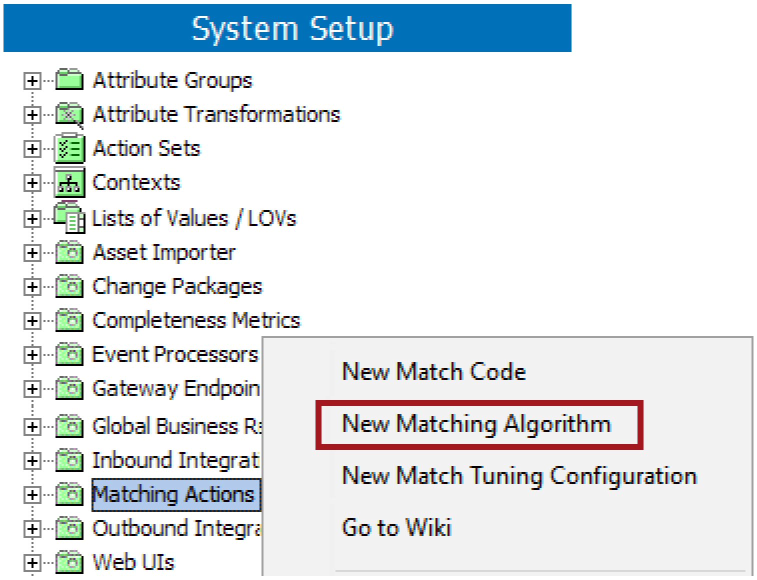
○ Change Packages
○ Completeness Metrics
○ Event Processors
○ Gateway Endpoint
○ Global Business Rules
○ Inbound Integration Endpoints
● Matching Algorithms
○ Outbound Integration Endpoints
○ RemoveAttributesFromGroup
○ Status Flags
○ Web UIs
○ Workflow Profiles
○ Workflows

ID: MatchingActions

Name: Matching Actions

3. Select **your matching algorithm object type**.
4. Enter an **ID**.
5. Enter a **Name**.
6. Click **Create**.

   The setup group is created and appears as a node in the System Setup hierarchy and allows the creation of matching algorithms.



7. Continue with the **Configuring Matching Algorithms** topic.

Configuring Matching Algorithms

A matching algorithm allows a user to define:

- The **match criteria**, which is what qualifies objects as duplicates.
- The **match action**, which is what the system should do with such duplicates.

Matching algorithms are used in **Match and Merge** and **Match and Link** solutions.

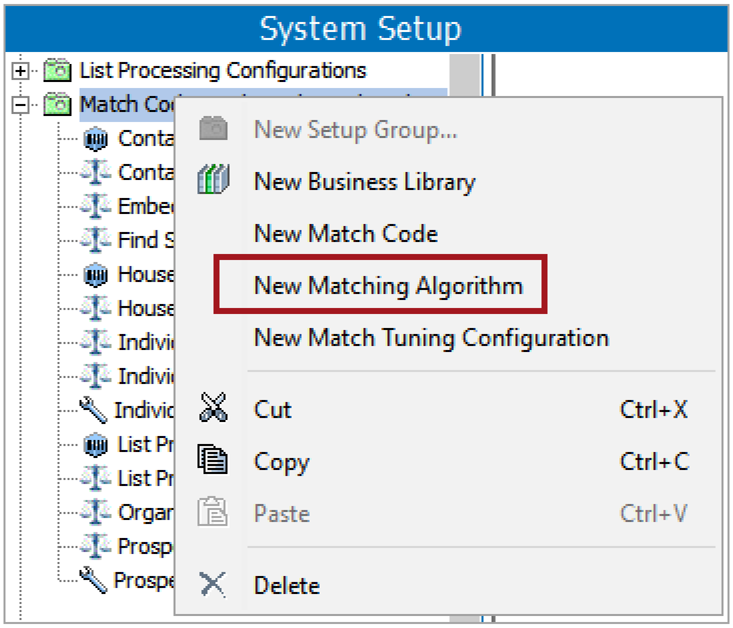
Tools available for tuning and monitoring the results of the matching algorithm are defined in the **Match Tuning** topic.

> **Note:** A matching algorithm definition can be exported as comments and submitted to an external source control system for comparison purposes. For details, refer to the **Configuration Management** documentation.

Create a Matching Algorithm

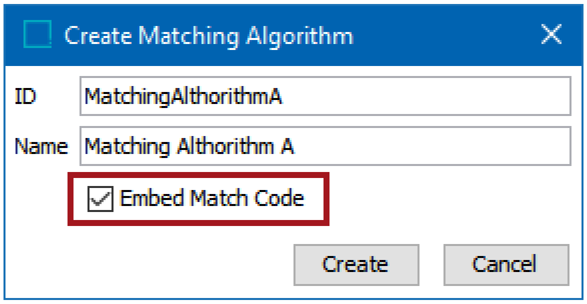
Use the following steps to create a matching algorithm:

1. In System Setup, right-click the node configured to house matching algorithms and select **New Matching Algorithm**.



2. In the Create Matching Algorithm dialog, define an **ID** and **Name** for the matching algorithm.

- Check the **Embed Match Codes** checkbox so the match codes are embedded in the algorithm. This is the recommended method.
- If the **Embed Match Codes** checkbox is not checked (legacy functionality), you must manually create a match code and link it to the matching algorithm. For more information, refer to the **Match Codes** topic.



Create Matching Algorithm

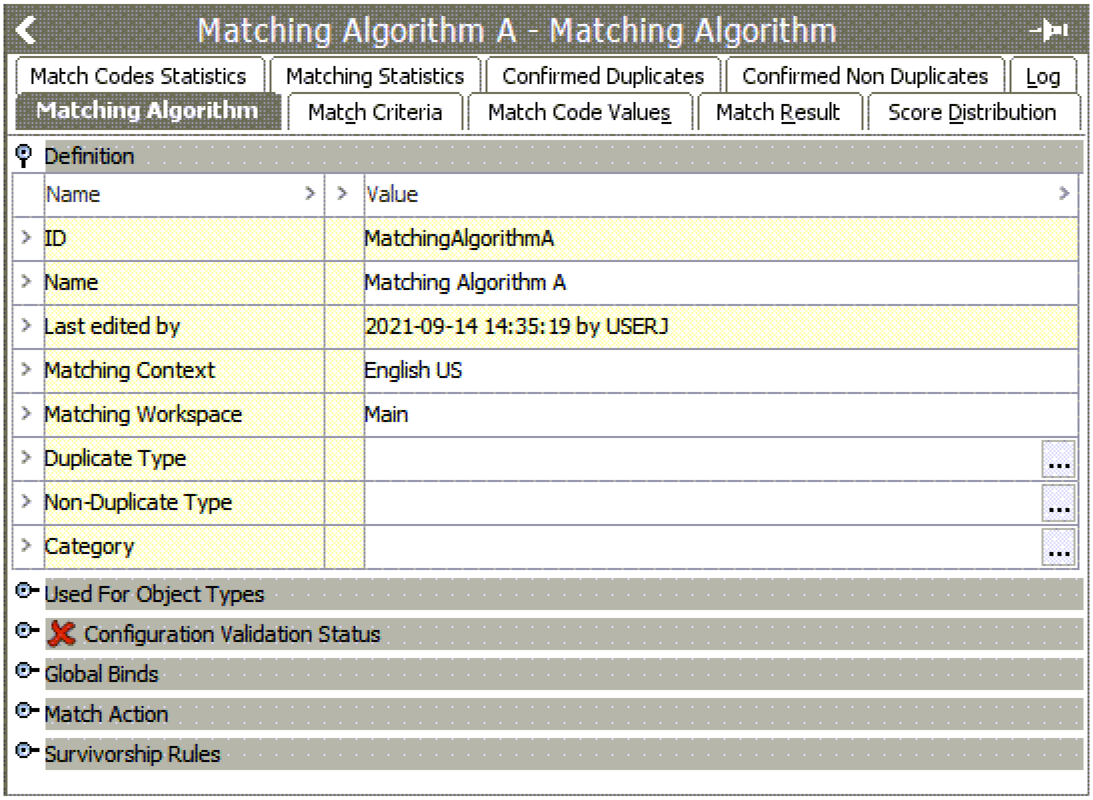
ID: MatchingAlthorithmA

Name: Matching Althorithm A

☑ Embed Match Code

- Click **Create** to display the Matching Algorithm object.

  Initially, the Configuration Validation Status flipper shows a red X. That indicator changes as the required elements are provided and configured correctly.



Matching Algorithm A - Matching Algorithm

Definition

Name	Value
ID	MatchingAlgorithmA
Name	Matching Algorithm A
Last edited by	2021-09-14 14:35:19 by USERJ
Matching Context	English US
Matching Workspace	Main
Duplicate Type	...
Non-Duplicate Type	...
Category	...

Used For Object Types

✗ Configuration Validation Status

Global Binds

Match Action

Survivorship Rules

Configure a Matching Algorithm

Use the following steps to configure a matching algorithm:

1. Open the matching algorithm and on the Matching Algorithm tab, in the Definition flipper:

   - For the **Matching Context** parameter, specify the context to run the matching algorithm. By default, the current context is set.

   - For the **Matching Workspace** parameter, specify the workspace to run the matching algorithm. By default, the Main workspace is selected.

   - For the **Duplicate Type** parameter, click the ellipsis button (...). In the 'Select a Duplicate Reference Type' dialog, select the appropriate reference type as defined in the component model. For more information, refer to the **Configuring Matching Component Model** topic.

     Note: The Duplicate Reference Types must be multivalued, None inheritance, no dimension dependencies, and not externally maintained.

   - For the **Non-Duplicate Type** parameter, click the ellipsis button (...). In the 'Select a Duplicate Reference Type' dialog, select the appropriate reference type as defined in the component model. For more information, refer to the **Configuring Matching Component Model** topic.

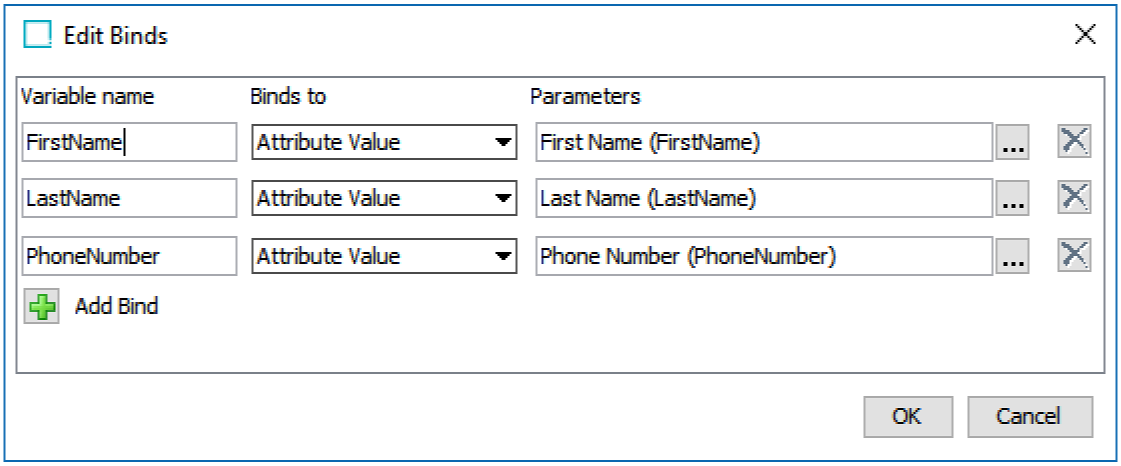
     Note: The Non-Duplicate Reference Types must be multivalued, None inheritance, no dimension dependencies, and not externally maintained. In Match and Link solutions, a valid 'Confirmed Justification Attribute' can be made valid on the reference type.

   - For the **Category** parameter, click the ellipsis button (...). In the 'Select Category' dialog that displays, select a parent node the matching algorithm will function under. If selected, the matching algorithm will **only** work under that node.

     Important: The match code category uses the default parent ID of the IIEP and/or matching algorithm. It does **not** depend on the parent ID of the individual entities in the incoming request / IIEP file. If the incoming record parent ID attempts to store the entity outside the match code category, STEP reports an error and adds the source record to the error file.

2. In the **Used for Object Types** flipper, click the **Add Object Type** link and select the object types to be considered by this algorithm.

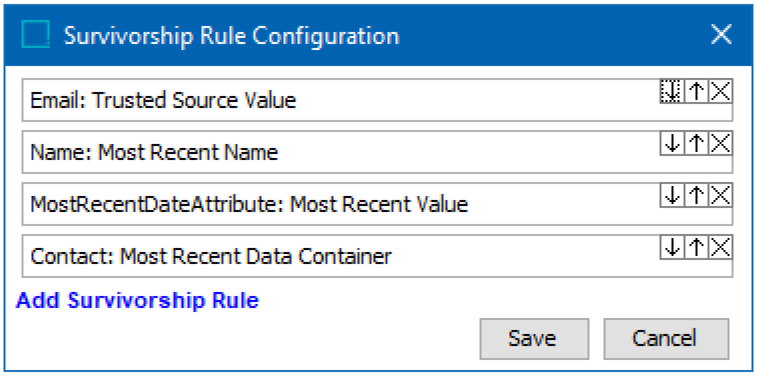
3. In the **Global Binds** flipper, while the **Legacy Global Binds** functionality can potentially improve the performance, the **Match Criteria Data Elements** functionality is preferred.



Edit Binds

Variable name	Binds to	Parameters
FirstName	Attribute Value	First Name (FirstName)
LastName	Attribute Value	Last Name (LastName)
PhoneNumber	Attribute Value	Phone Number (PhoneNumber)

4. In the **Survivorship Rules** flipper, configure as defined in the **Configuring Survivorship Rules** topic.

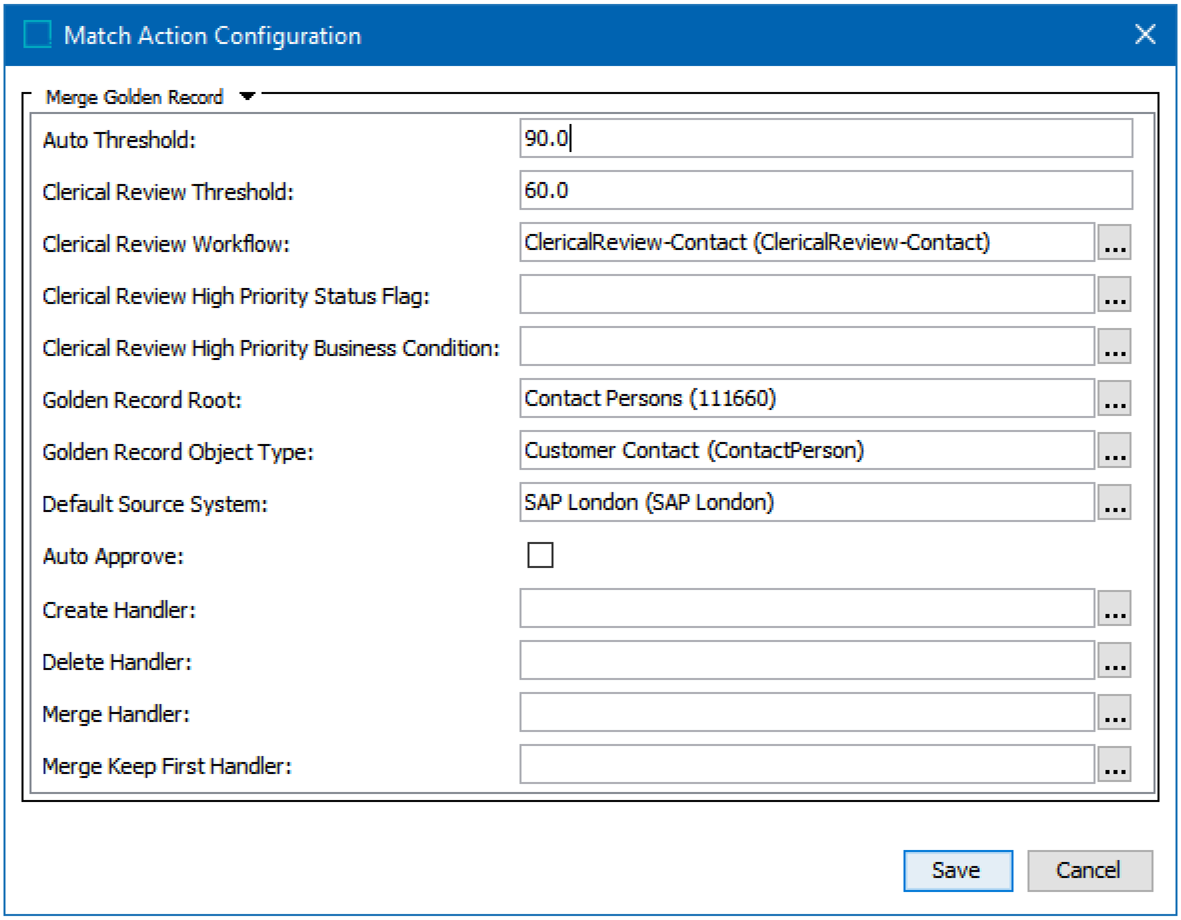


Survivorship Rule Configuration

- Email: Trusted Source Value
- Name: Most Recent Name
- MostRecentDateAttribute: Most Recent Value
- Contact: Most Recent Data Container

5. On the Match Criteria tab, configure the elements as defined in the **Match Criteria** topic.

6. Return to the Matching Algorithm tab, and in the **Match Action** flipper, set up the match action as defined in the **Match Actions** topic.



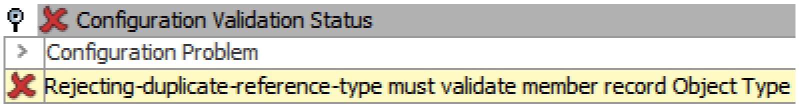
Match Action Configuration

Merge Golden Record

Auto Threshold:	90.0
Clerical Review Threshold:	60.0
Clerical Review Workflow:	ClericalReview-Contact (ClericalReview-Contact)
Clerical Review High Priority Status Flag:	
Clerical Review High Priority Business Condition:	
Golden Record Root:	Contact Persons (111660)
Golden Record Object Type:	Customer Contact (ContactPerson)
Default Source System:	SAP London (SAP London)
Auto Approve:	□
Create Handler:	
Delete Handler:	
Merge Handler:	
Merge Keep First Handler:	

7. On the Matching Algorithm tab, verify your matching algorithm configuration status and take any necessary action.

- A red 'X' displays when the configuration is invalid. Open the flipper to view the errors that must be addressed. Correct any errors shown before running the matching algorithm.

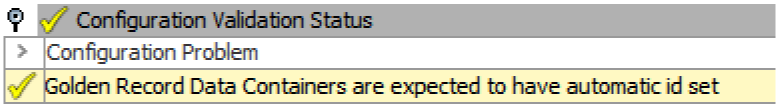


Configuration Validation Status

Configuration Problem

Rejecting-duplicate-reference-type must validate member record Object Type

- A yellow checkmark indicates warnings that should be addressed.

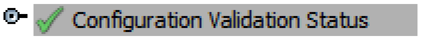


Configuration Validation Status

Configuration Problem

Golden Record Data Containers are expected to have automatic id set

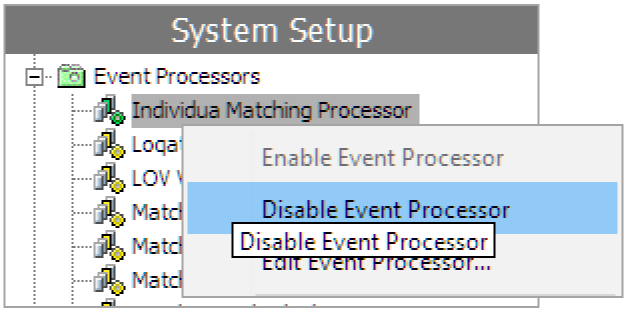
- A green checkmark indicates the matching algorithm has a valid configuration.



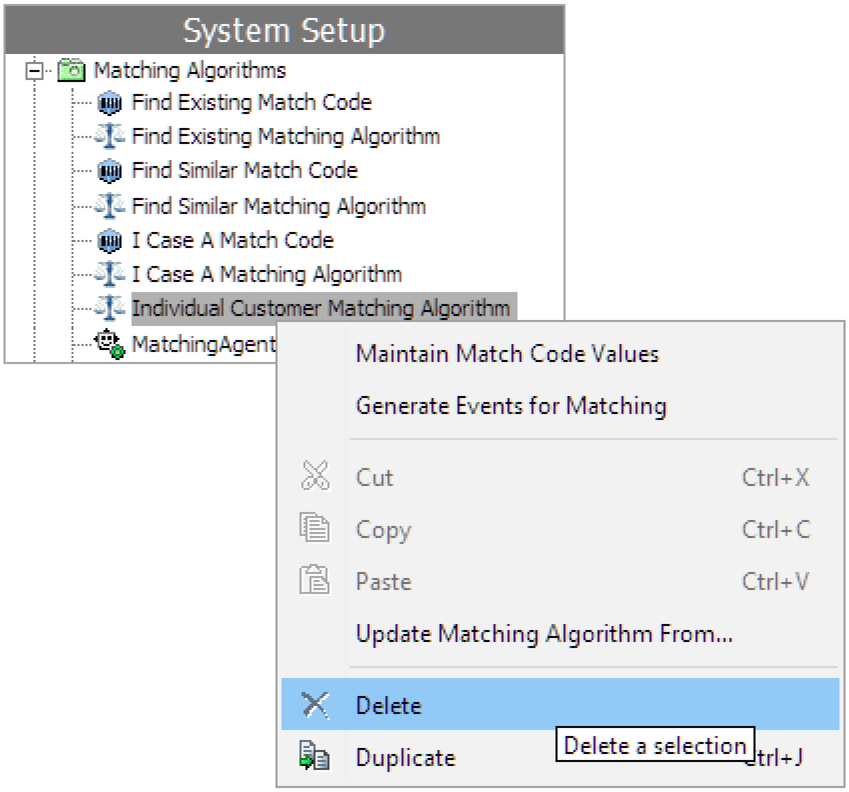
Delete a Matching Algorithm

To delete a matching algorithm, follow the below steps:

1. In System Setup, navigate to the event processor tied to the matching algorithm. Right-click on the event processor and select 'Disable Event Processor.'



2. Navigate to the matching algorithm you wish to delete. Right-click on the algorithm and select 'Delete.'


System Setup
Matching Algorithms
Find Existing Match Code
Find Existing Matching Algorithm
Find Similar Match Code
Find Similar Matching Algorithm
I Case A Match Code
I Case A Matching Algorithm
Individual Customer Matching Algorithm
MatchingAgent
Maintain Match Code Values
Generate Events for Matching
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Update Matching Algorithm From...
Delete
Duplicate
Delete a selection
trl+J

Note: If you attempt to delete a matching algorithm with existing tasks in a workflow, an error appears stating, 'Cannot delete match algorithm with existing workflow tasks.' If you receive this error, finish your workflow tasks and try again. If your Task List contains more tasks than can be completed manually, use a bulk update to remove all items from the workflow, as defined in the **Remove Objects from Workflows** topic in the **Workflows** documentation.

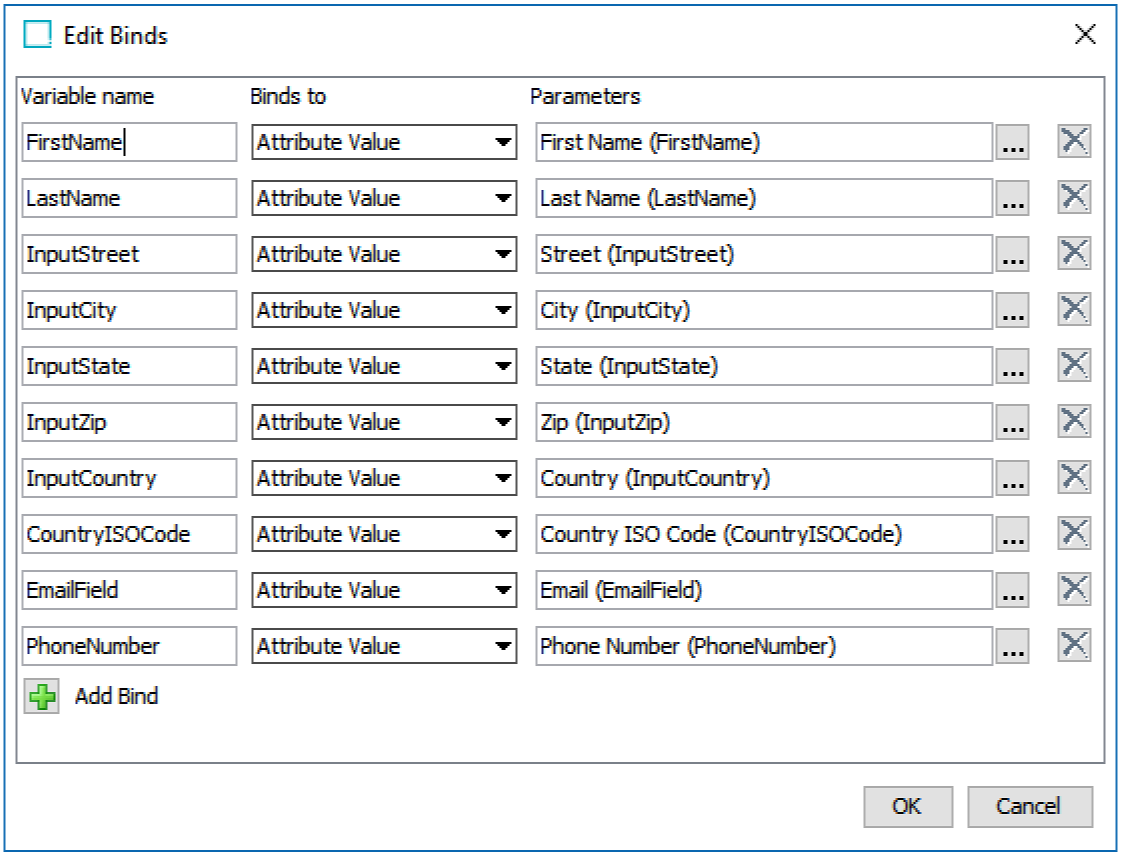
Legacy Global Binds

The matching process can strain performance. When processing large sets of data, there is potentially a significant performance gain if the matching functionality can fetch the values for matching before the matching process begins. This fetching of data is possible via global binds configured on the matching algorithm, where the matching algorithm logic uses attributes that are bound to specific variable names. The system fetches the values for the attributes used in the decision table comparison before the match criteria logic is applied and can be referenced from both JavaScript and STEP functions.

> **Important:** Global binds are legacy functionality and are not optimized for use with In-Memory. Use Data Elements instead, as defined in the **Match Criteria Data Elements** topic.

1. On a matching algorithm object, open the Global Binds flipper and click the **Edit Global Binds** link.

2. Click the **Edit Global Binds** link to open the 'Edit Binds' dialog.



Variable name	Binds to	Parameters
FirstName	Attribute Value	First Name (FirstName)
LastName	Attribute Value	Last Name (LastName)
InputStreet	Attribute Value	Street (InputStreet)
InputCity	Attribute Value	City (InputCity)
InputState	Attribute Value	State (InputState)
InputZip	Attribute Value	Zip (InputZip)
InputCountry	Attribute Value	Country (InputCountry)
CountryISOCode	Attribute Value	Country ISO Code (CountryISOCode)
EmailField	Attribute Value	Email (EmailField)
PhoneNumber	Attribute Value	Phone Number (PhoneNumber)

3. Click the **Add Bind** button (➕) to create a new bind.

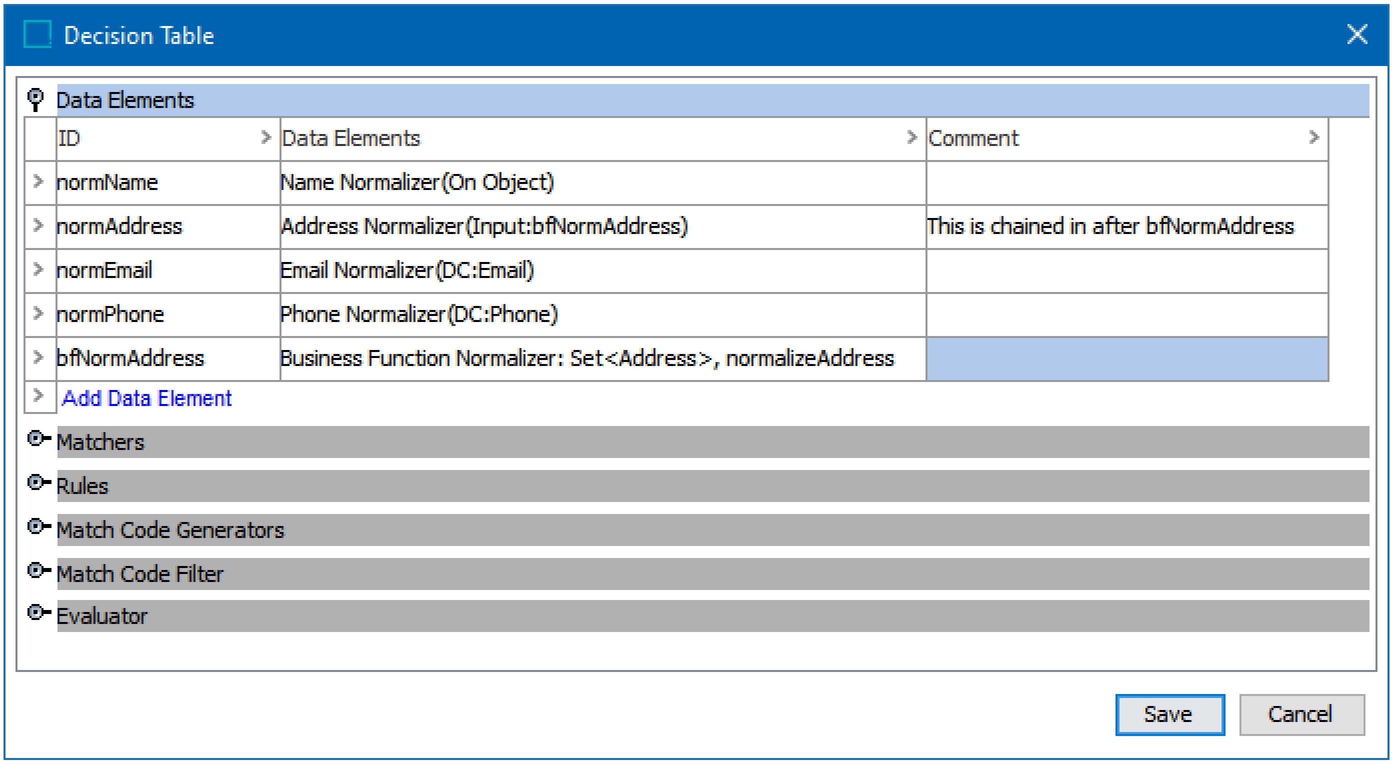
- For **Variable name**, specify a variable name for the bind.
- For **Binds to**, select a bind from the dropdown (some binds are displayed within a group).
- For **Parameters**, when available, click the ellipsis button (...) to specify an object to bind.
- Click the delete button (☒) to remove a bind as needed.

4. Click **OK** to close the dialog and return to the Matching Algorithm object.

Match Criteria Data Elements

Data Elements retrieve data from source objects and standardize it for use in matchers and match code generators.

The Data Elements flipper of a decision table defines the initial input data for the match criteria. The output of a data element is used by the selected Matchers (as defined in the **Match Criteria Matchers** topic). The data elements required are determined by the kind of data being matched, for example, the following image shows data elements for matching on individuals.



ID	Data Elements	Comment
normName	Name Normalizer(On Object)	
normAddress	Address Normalizer(Input:bfNormAddress)	This is chained in after bfNormAddress
normEmail	Email Normalizer(DC:Email)	
normPhone	Phone Normalizer(DC:Phone)	
bfNormAddress	Business Function Normalizer: Set<Address>, normalizeAddress	

A data element retrieves the data required for comparison during matching. This often involves the reduction (or 'normalizing') of data to a matchable form, like making letters lower case in a text string, removing spaces from phone numbers, or expanding abbreviations. Some data elements retrieve data for matching without normalizing.

Data elements can be chained so that the output of one data element can be used as input to another data element. Examples are included in the **Chaining Match Expressions to Expand Functionality** section of the **Matching Algorithms and Match Expressions** topic.

Data Element	Data Element Type	Object Type Allowed
Attribute Value	General Purpose	Entities Products
Business Function Normalizer	General Purpose	Entities Products
Constant	General Purpose	Entities Products
Function	General Purpose	Entities Products
JavaScript Function	General Purpose	Entities Products
Address Normalizer	Preconfigured	Entities
Email Normalizer	Preconfigured	Entities
Organization Name Normalizer	Preconfigured	Entities
Person Name Normalizer	Preconfigured	Entities
Phone Normalizer	Preconfigured	Entities
Words Normalizer	Preconfigured	Entities

Data Element: Address Normalizer

The address normalizer produces a normalized set of addresses for use in the corresponding address matcher.

Prerequisites

Configure the address component model (defined in the **Address Component Model** section of the **Data Integration** documentation).

Input

The following data is provided by the input address element attributes mapped to the address component model. The address object uses the input attribute values or, if the Standardized Zip is set, the address object is populated using the standardized attributes.

Input Attributes	Standardized Attributes
• Input City • Input Country • Input State • Input Street • Input Zip	• Standardized City • Standardized Country • Standardized Country ISO Code • Standardized State • Standardized Street • Standardized Zip

When configuring the data element, the Input Parameters field allows selection of:

- 'Use Attribute on Object' – by default, this option is set to ‘True’ and indicates to read attributes on the object itself. Click the Value dropdown to manually set it to 'False' when using information from a Data Container or an Input Normalizer.
- 'Data Container' – read attributes from the data container.
- ‘Input Normalizer’ – read outputs from the selected Match Expression, as defined in the **Matching Algorithms and Match Expressions** topic.

Output

The output of the address normalizer is a class:
java.util.Set<com.stibo.partydatamatching.domain.address.Address>

For more information on the contents of the class, refer to the **STEP API Documentation** on the STEP Start Page and review the documents linked from within the **Scripting API** section.

Functionality

The address normalizer automatically makes the following modifications to the address in the order listed for comparison purposes only:

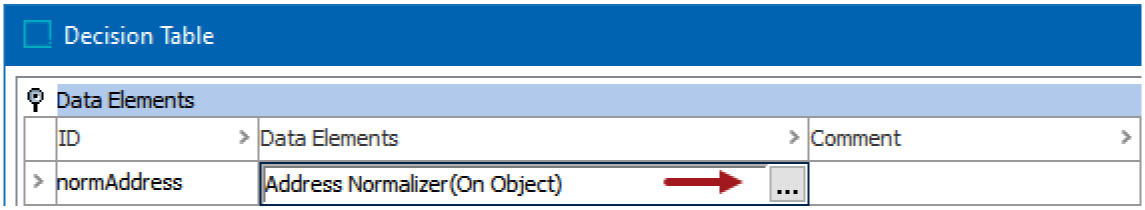
1. All elements – remove leading and trailing white space
2. Country – lower-case text
3. Country ISO – remove characters other than Latin letters and numbers
4. Region – lower-case text
5. City – lower-case text
6. Postal code – remove all spaces, remove dash (-) characters, lower-case text
7. Street – lower-case text

Because address information varies between systems and countries, it is sometimes necessary to chain address normalizers. For an example of adding a custom address normalizer business function that further normalizes the address after the standard normalizer runs, refer to the **Data Elements: Business Function Normalizer** topic.

Configuring an Address Normalizer Data Element

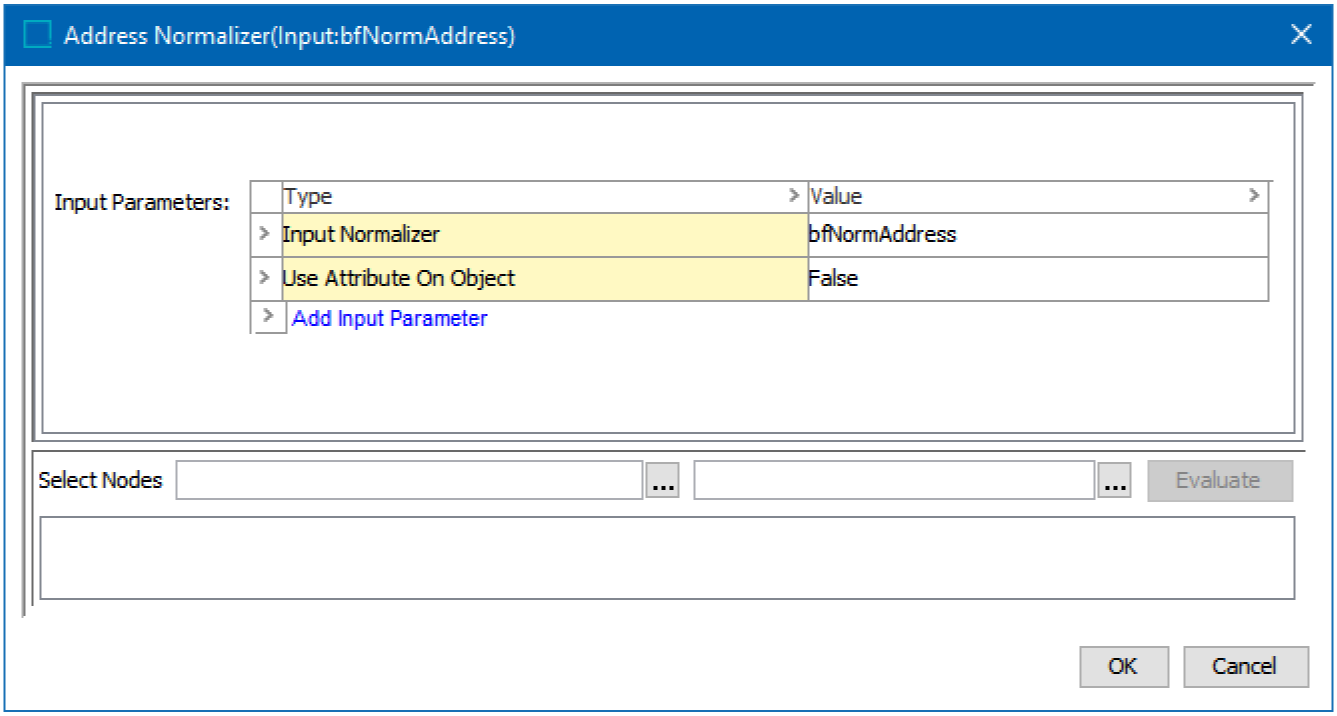
After adding the address normalizer in the Data Elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Data Elements column and click the ellipsis button (**...**) to access the configuration dialog.



ID	Data Elements	Comment
normAddress	Address Normalizer(On Object) ...	

2. On the address normalizer dialog:



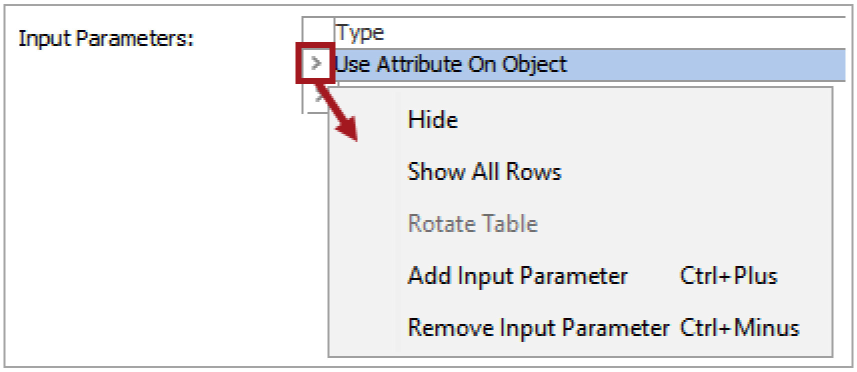
Address Normalizer(Input:bfNormAddress)

Input Parameters:

Type	Value
Input Normalizer	bfNormAddress
Use Attribute On Object	False

- For the **Input Parameters**, define the source of the data to be normalized. Refer to the **Input** section above for details.

  Right-click the arrow in the first column of the Input Parameters table for additional display and edit options. Although it appears that the default 'Use Attribute On Object' parameter can be removed, after closing the dialog it will continue to display. Instead, if a different input parameter is used, click the Value dropdown and manually set 'Use Attribute On Object' option to 'False.'

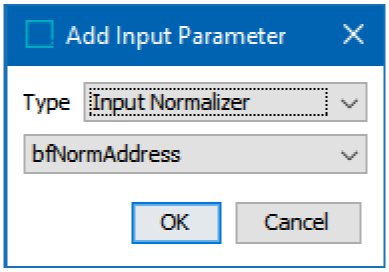


Input Parameters:

Type
Use Attribute On Object

Hide
Show All Rows
Rotate Table
Add Input Parameter Ctrl+Plus
Remove Input Parameter Ctrl+Minus

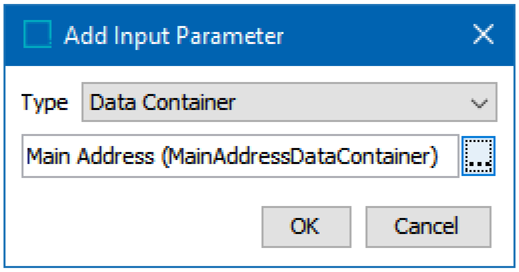
Click the **Add Input Parameter** link to add other input parameters.



Add Input Parameter

Type: Input Normalizer

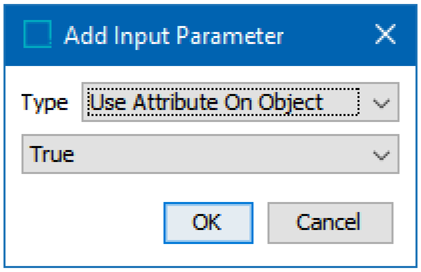
bfNormAddress



Add Input Parameter

Type: Data Container

Main Address (MainAddressDataContainer)

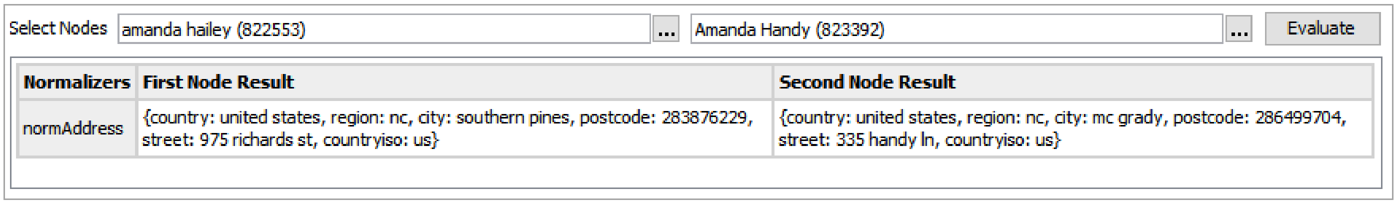


Add Input Parameter

Type: Use Attribute On Object

True

3. To test the configuration, for the Select Nodes parameters:



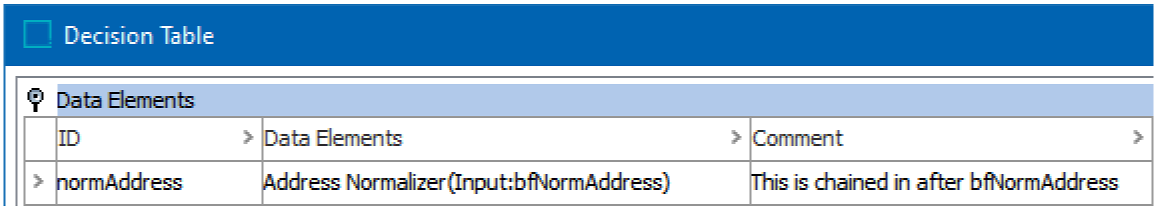
Select Nodes: amanda hailey (822553) ... Amanda Handy (823392) ...

Normalizers	First Node Result	Second Node Result
normAddress	{country: united states, region: nc, city: southern pines, postcode: 283876229, street: 975 richards st, countryiso: us}	{country: united states, region: nc, city: mc grady, postcode: 286499704, street: 335 handy ln, countryiso: us}

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

4. Click **OK** to save and display the configuration in the Data Elements flipper. Click into a Comment cell to add relevant information as desired.



Decision Table

Data Elements

ID	Data Elements	Comment
normAddress	Address Normalizer(Input:bfNormAddress)	This is chained in after bfNormAddress

Data Element: Attribute Value

The attribute value data element allows users to specify a single attribute and output its value. This data element does not normalize its output.

Input

The value from any attribute can be used as input.

Output

The output of the Attribute data element is a Java String.

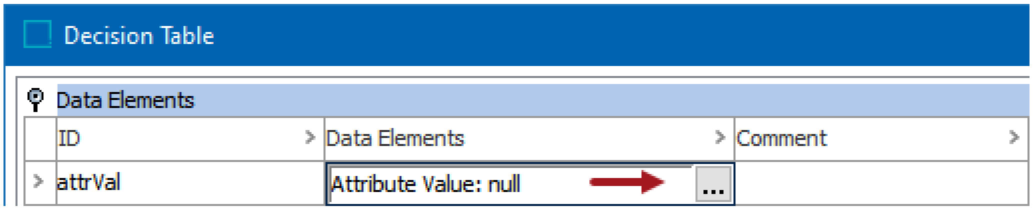
Functionality

Returns the value of the attribute as a simple string. If the attribute is multivalued, the values are separated by '<multisep/>'. Refer to the **Inline References and Multi-Valued Attributes** section of the **Inline References in Attribute Values** topic of the **Getting Started** documentation.

Configuring an Attribute Value Data Element

After adding an attribute value data element in the Data Elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Data Elements column and click the ellipsis button (...) to access the configuration dialog.



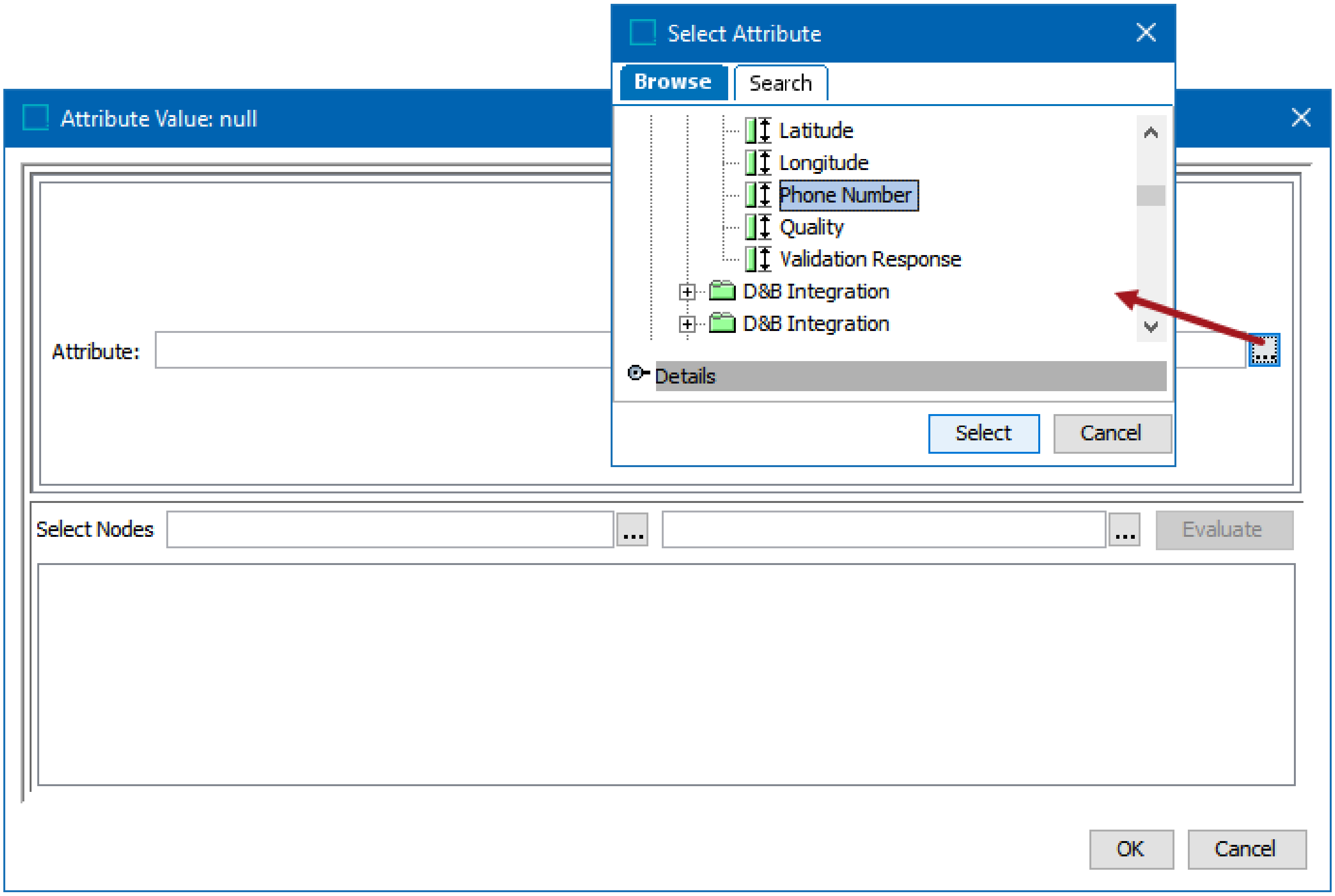
Decision Table

Data Elements

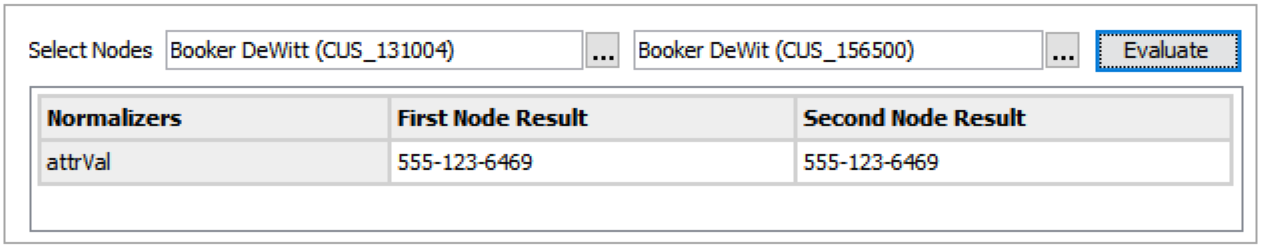
ID	Data Elements	Comment
attrVal	Attribute Value: null	

2. On the Attribute Value dialog, click the ellipsis button (...) to display the Select Attribute dialog.

3. Use browse or search, choose the desired attribute, and click the **Select** button.



4. To test the configuration, for the Select Nodes parameters:



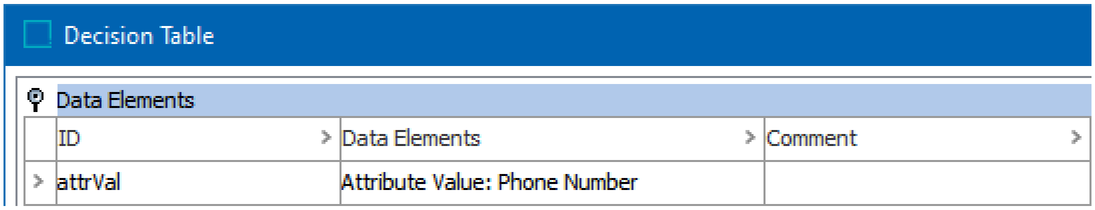
Select Nodes: Booker DeWitt (CUS_131004) ... Booker DeWit (CUS_156500) ... Evaluate

Normalizers	First Node Result	Second Node Result
attrVal	555-123-6469	555-123-6469

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

5. Click **OK** to save and display the configuration in the Data Elements flipper. Click into a Comment cell to add relevant information as desired.



ID	Data Elements	Comment
attrVal	Attribute Value: Phone Number	

Data Element: Business Function Normalizer

The Business Function Normalizer is the most versatile normalizer. Because it uses a business function, it can take any number of values from the source records and produce a normalized data element. You can also use it to expand the functionality of Party Data normalizers as illustrated in the **Chaining Match Expressions to Expand Functionality** section of the **Matching Algorithms and Match Expressions** topic.

For more information, refer to the **Business Functions** topic of the **Business Rules** documentation.

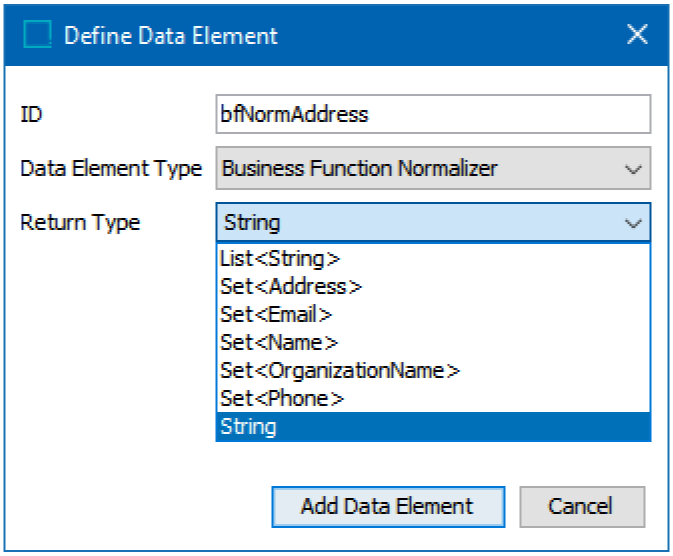
Input

The Business Function Normalizer can take as inputs the object that is to be compared, or the output from another data element / normalizer. For details, refer to the **Matching Algorithms and Match Expressions** topic.

Function Input Parameters on the Business Function are mapped to the compared objects or other data elements as part of the configuration.

Output

The output of the Business Function Normalizer can be used by any Data Element, Matcher, or Match Code Generator. The Return Type selected when creating the Business Function Normalizer must match the expected input of the consumers of the output.



Define Data Element

ID: bfNormAddress

Data Element Type: Business Function Normalizer

Return Type: String

The Business Function normalizer returns one of these output types:

- java.util.List<java.lang.String>
- java.util.Set<com.stibo.partydatamatching.domain.address.Address>
- java.util.Set< com.stibo.partydatamatching.domain.email.Email>
- java.util.Set< com.stibo.partydatamatching.domain.organizationname.OrganizationName>
- java.util.Set< com.stibo.partydatamatching.domain.name.Name>
- java.util.Set<com.stibo.partydatamatching.domain.phone.Phone>
- java.lang.String

Note: The Business Function Normalizer should output a completely new set of strings / values and should not overwrite existing strings / values.

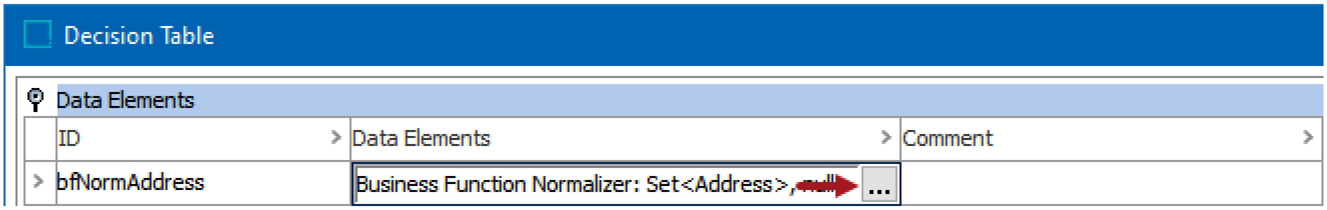
Functionality

The functionality of the Business Function Normalizer is configured using JavaScript.

Configuring a Business Function Normalizer Data Element

After selecting the Return Type (discussed above) and adding Business Function Normalizer in the Data Elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Data Elements column and click the ellipsis button (...) to access the configuration dialog.

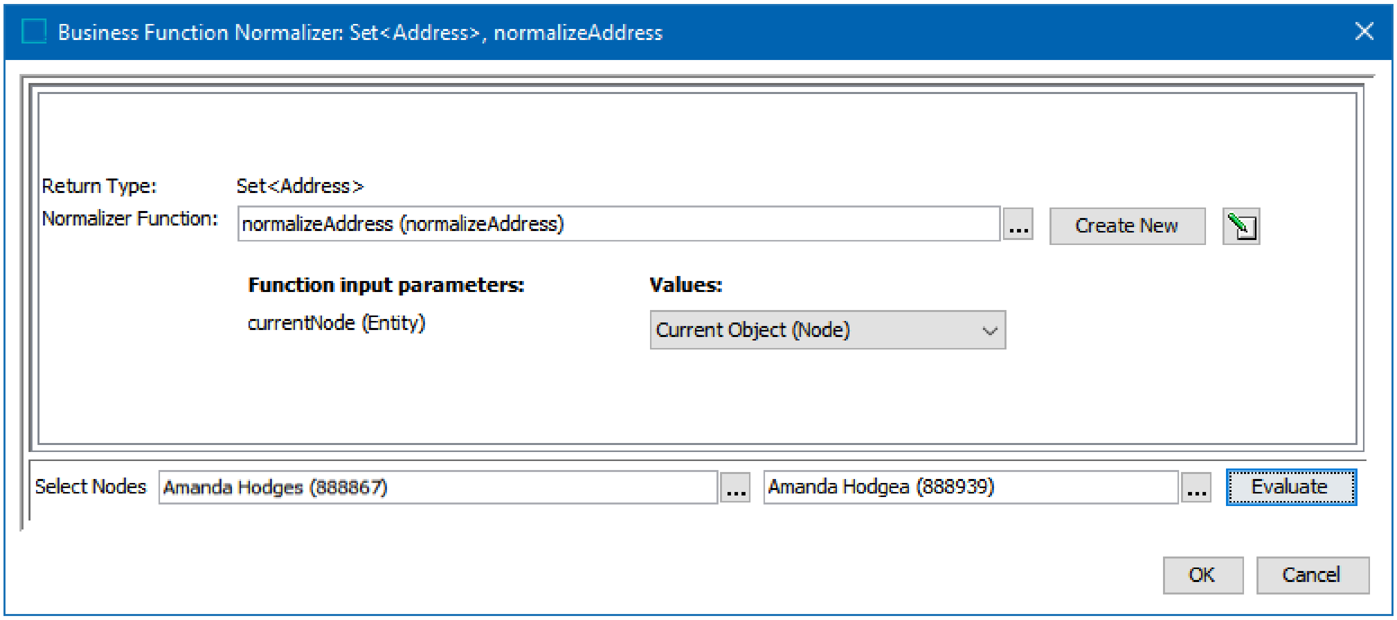


Decision Table

Data Elements

ID	Data Elements	Comment
bfNormAddress	Business Function Normalizer: Set<Address>, ...	

2. On the Business Function Normalizer dialog, the previously selected Return Type is displayed.
3. Choose a option to identify the normalizer function:
   - Click the ellipsis button (...) to display the Select Business Function dialog. If necessary, click the edit button (✎) to modify the selected global business function.
   - Click the **Create New** button to create a new business function.
4. For the displayed 'Function input parameters' select the appropriate values for the normalizer function.



Business Function Normalizer: Set<Address>, normalizeAddress

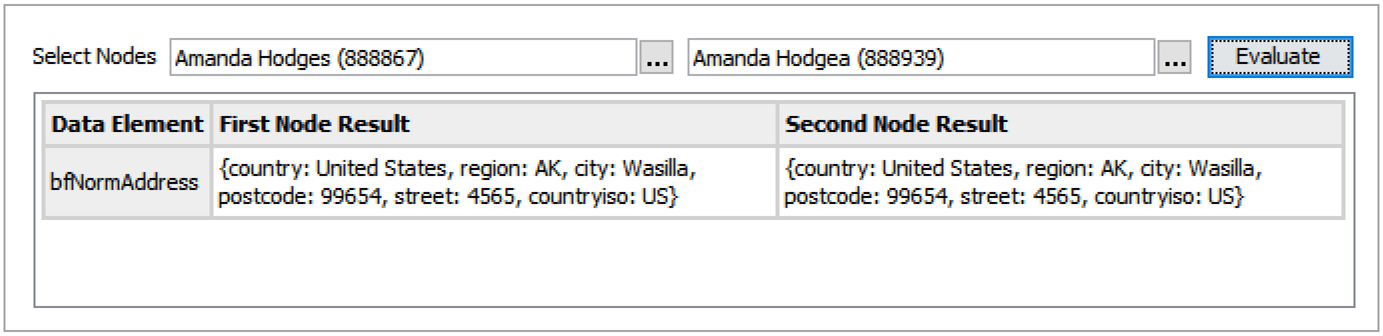
Return Type: Set<Address>

Normalizer Function: normalizeAddress (normalizeAddress)

Function input parameters:	**Values:**
currentNode (Entity)	Current Object (Node)

Select Nodes: Amanda Hodges (888867) | Amanda Hodgea (888939)

5. To test the configuration, for the Select Nodes parameters:



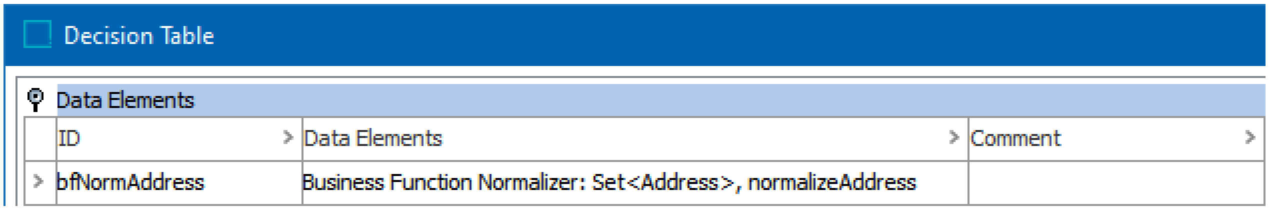
Select Nodes: Amanda Hodges (888867) | Amanda Hodgea (888939)

Data Element	First Node Result	Second Node Result
bfNormAddress	{country: United States, region: AK, city: Wasilla, postcode: 99654, street: 4565, countryiso: US}	{country: United States, region: AK, city: Wasilla, postcode: 99654, street: 4565, countryiso: US}

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

6. Click **OK** to save and display the configuration in the Data Elements flipper.



Decision Table

Data Elements

ID	Data Elements	Comment
bfNormAddress	Business Function Normalizer: Set<Address>, normalizeAddress	

Data Element: Constant

A constant can be used to make a default value or a setting more obvious in the matching algorithm. This can be done by declaring the constant in the data elements and then adding it as an input parameter to another data element, a matcher, or a match code generator. Constants can also improve the reuse of business functions across matching algorithms.

For example, a business function adds a specific normalization to values only from a configured source system. The constant can list those source systems, separated by semicolon or similar. This makes the normalization business function reusable across more matching algorithms and makes the source system list easier to maintain.

Input

None

Output

The output of the attribute data element is a string.

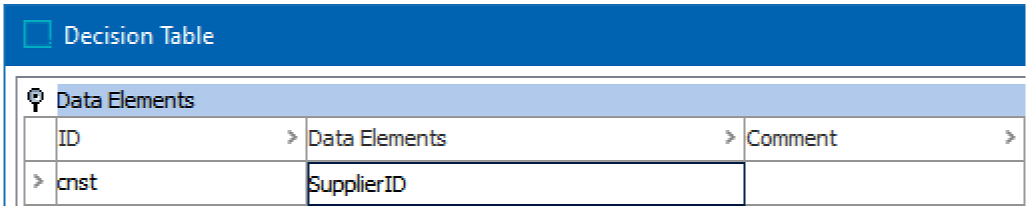
Functionality

Makes the entered constant string available to other data elements, matchers and match code generators.

Configuring a Constant Data Element

After adding the constant in the data elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

Click into the Data Elements column and add the desired text.



Data Elements

ID	Data Elements	Comment
cnst	SupplierID	

Data Element: Email Normalizer

An email normalizer can normalize email data for use in the corresponding email matcher.

Input

When configuring the data element:

1. The **Input Attribute** field defines an attribute to be used as input.
2. The **Input Parameters** field allows selection of:
   - 'Use Attribute on Object' – by default, this option is set to ‘True’ and indicates to read attributes on the object itself. Click the Value dropdown to manually set it to 'False' when using information from a Data Container or an Input Normalizer.
   - 'Data Container' – read attributes from the data container.
   - ‘Input Normalizer’ – read outputs from the selected Match Expression, as defined in the **Matching Algorithms and Match Expressions** topic.

Output

The output of an email normalizer is a java.util.Set< com.stibo.partydatamatching.domain.email.Email>

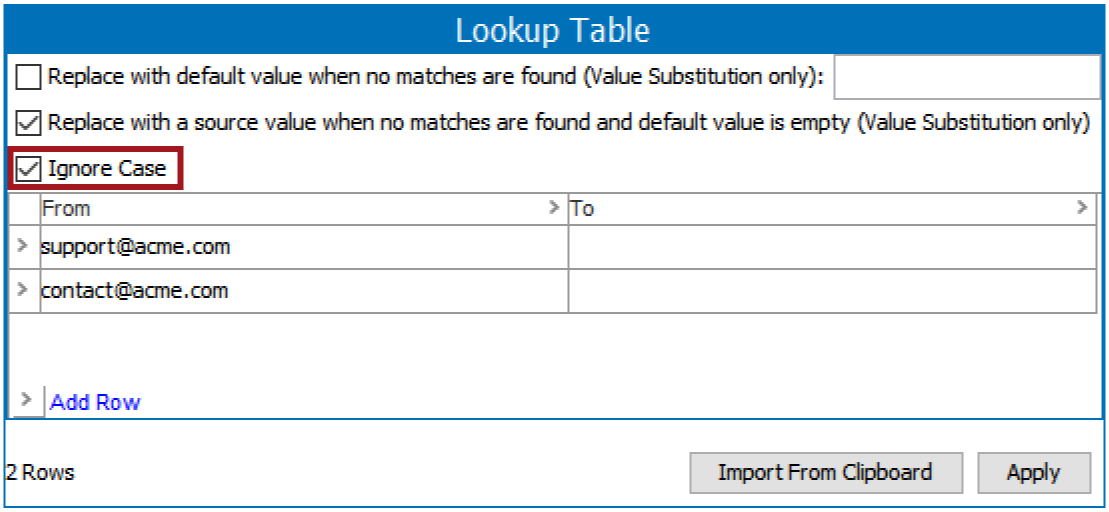
Functionality

The email normalizer automatically makes the following modifications to email for comparison purposes only in the order listed:

1. Applies the Replacement Lookup Table (the 'Ignore Case' option must be enabled)
2. Lower-case text
3. Remove leading and trailing white space

Optional

Transformation Lookup Tables can replace generic email values with empty strings, which prevents them from being used in matching records. The 'Ignore Case' option must be enabled. To create a Transformation Lookup Table for use in the email normalizer, refer to the **Transformation Lookup Tables** topic in the **Resource Materials** section of online help.



Lookup Table

□ Replace with default value when no matches are found (Value Substitution only):

☑ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

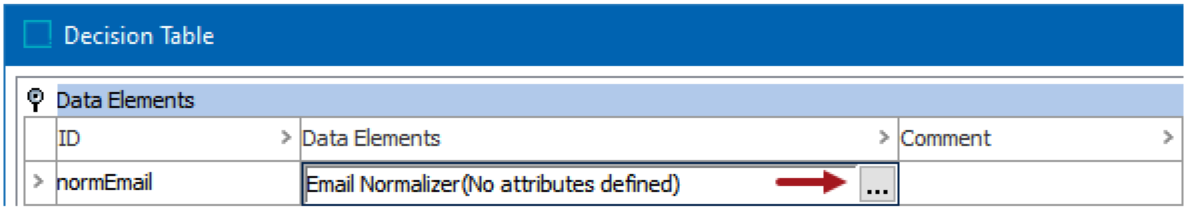
☑ Ignore Case

From	To
support@acme.com	
contact@acme.com	

Configuring an Email Normalizer Data Element

After adding the email normalizer in the Data Elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Data Elements column and click the ellipsis button (...) to access the configuration dialog.

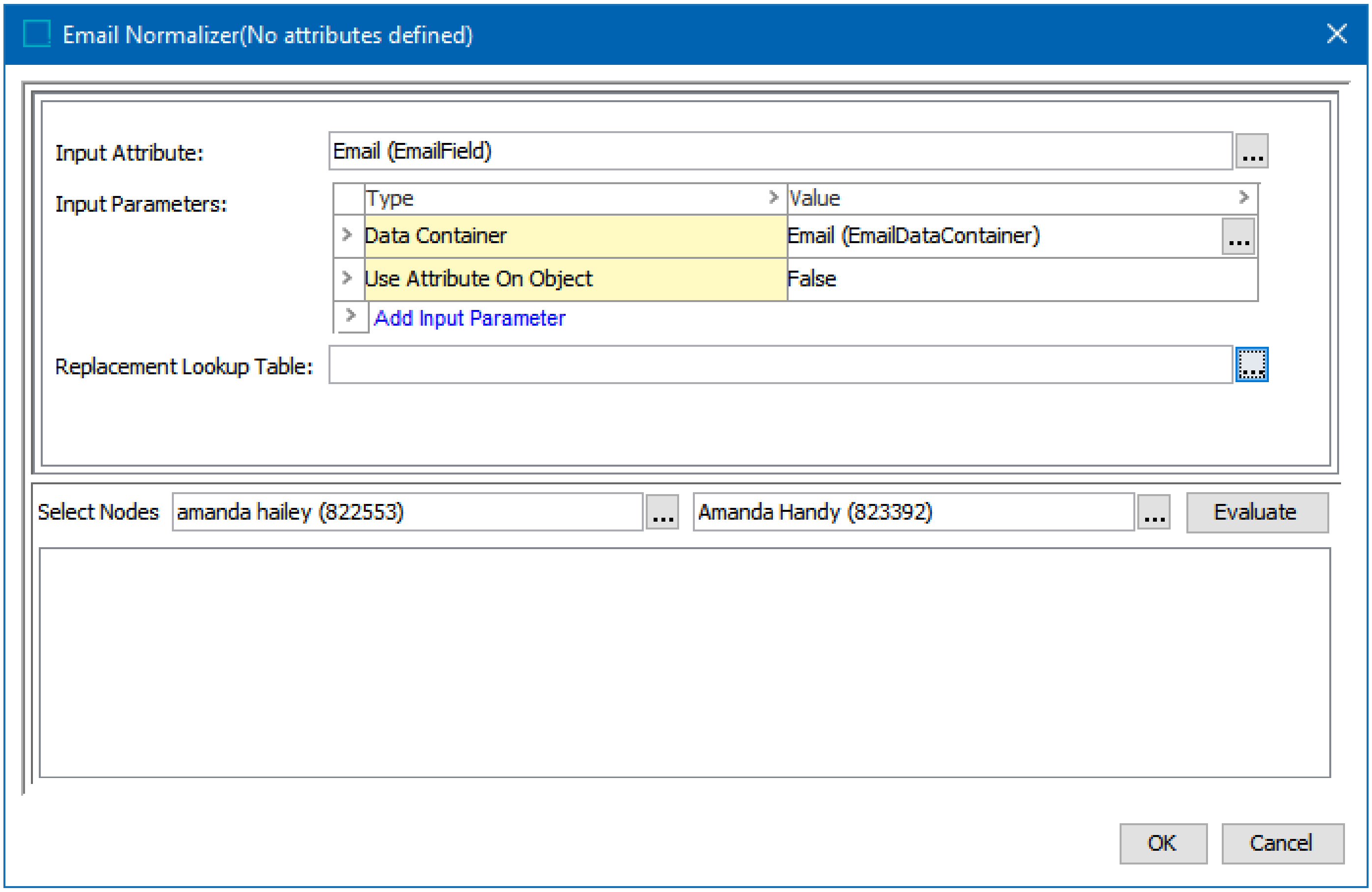


Decision Table

Data Elements

ID	Data Elements	Comment
normEmail	Email Normalizer(No attributes defined)	

2. On the Email Normalizer dialog:



Email Normalizer(No attributes defined)

Input Attribute: Email (EmailField)

Input Parameters:

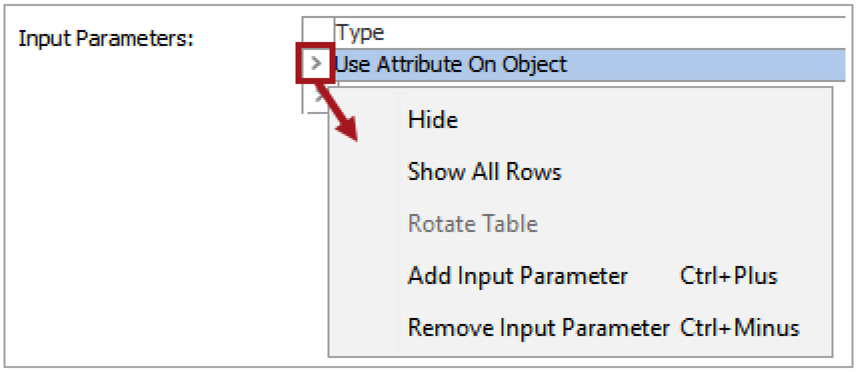
Type	Value
Data Container	Email (EmailDataContainer)
Use Attribute On Object	False

Replacement Lookup Table:

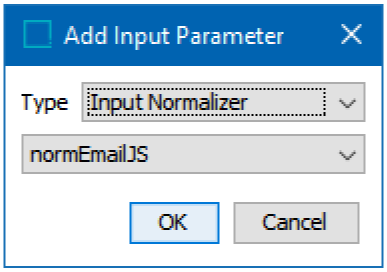
Select Nodes: amanda hailey (822553) | Amanda Handy (823392)

- For the **Input Attribute**, click the ellipsis button (...) and select the email attribute.
- For the **Input Parameters**, define the source of the data to be normalized. Refer to the **Input** section above for details.

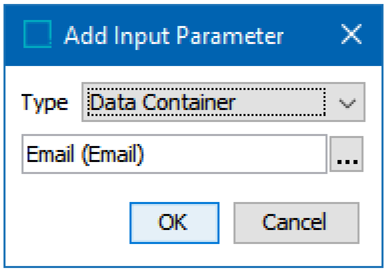
  Right-click the arrow in the first column of the Input Parameters table for additional display and edit options. Although it appears that the default 'Use Attribute On Object' parameter can be removed, after closing the dialog it will continue to display. Instead, if a different input parameter is used, click the Value dropdown and manually set 'Use Attribute On Object' option to 'False.'



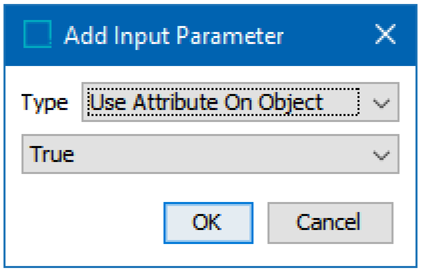
Click the **Add Input Parameter** link to add other input parameters.



Type: Input Normalizer
normEmailJS



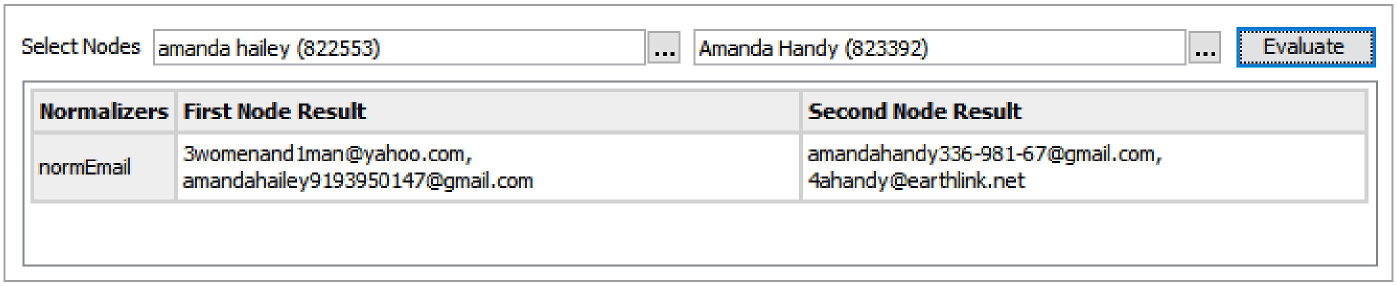
Type: Data Container
Email (Email)



Type: Use Attribute On Object
True

- For the **Replacement Lookup Table**, click the ellipsis button (**...**) and select the transformation lookup table asset.

3. To test the configuration, for the Select Nodes parameters:



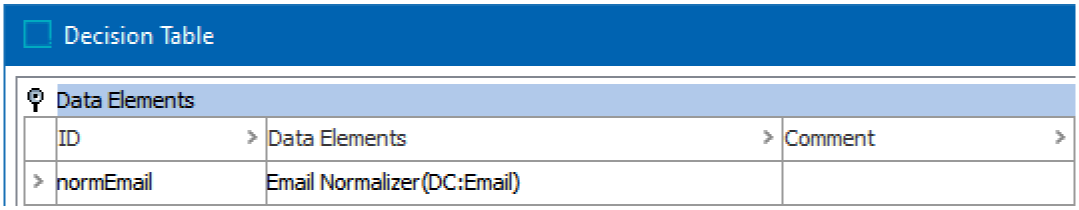
Select Nodes: amanda hailey (822553) | Amanda Handy (823392)

Normalizers	First Node Result	Second Node Result
normEmail	3womenand1man@yahoo.com, amandahailey9193950147@gmail.com	amandahandy336-981-67@gmail.com, 4ahandy@earthlink.net

- Click the ellipsis button (**...**) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

4. Click **OK** to save and display the configuration in the Data Elements flipper.



ID	Data Elements	Comment
normEmail	Email Normalizer(DC:Email)	

Data Element: Function

The function element normalizes values via built-in STEP functions. For more information, refer to the **Function Editor** topic of the **Resource Materials** in online help.

Input

The selected STEP function defines the types of values allowed as input.

Output

The output of a function is a java.lang.String.

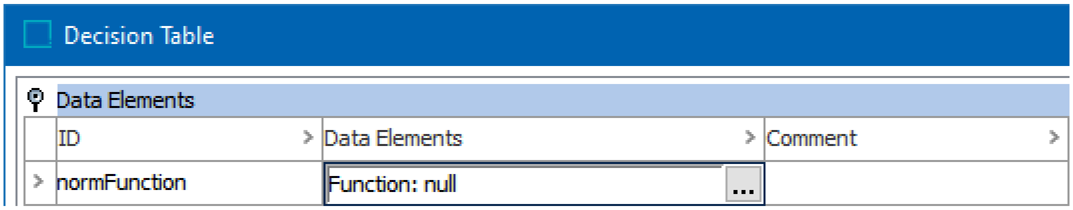
Functionality

The function data element uses the same editor and language as calculated attributes.

Configuring a Function Data Element

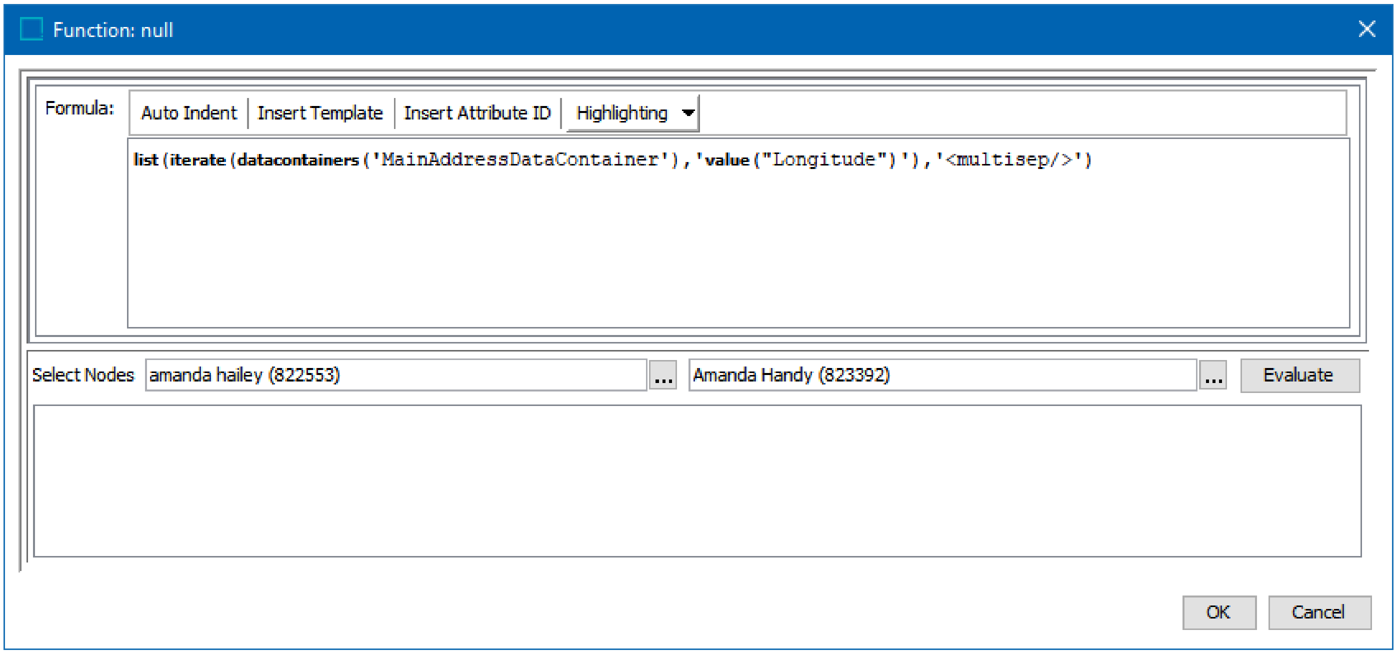
After adding the function in the Data Elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Data Elements column and click the ellipsis button (...) to access the configuration dialog.



ID	Data Elements	Comment
normFunction	Function: null ...	

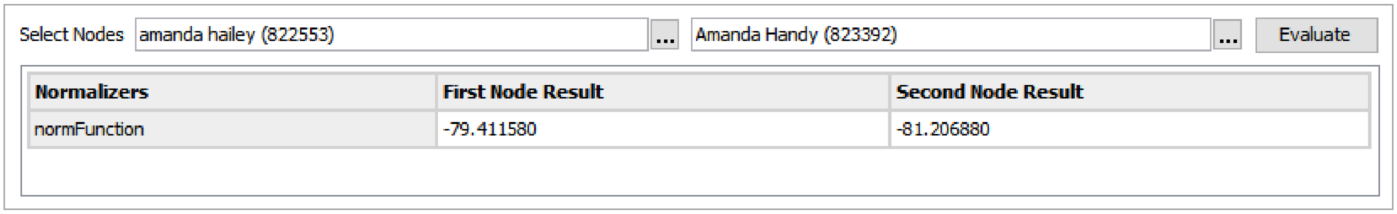
2. On the Function dialog, click the **Insert Template** tab to review the available functions and use a template. Add your code to the Formula section, as shown below.



```
list(iterate(datacontainers('MainAddressDataContainer'),'value("Longitude")'),'<multisep/>')
```

Select Nodes: amanda hailey (822553) | Amanda Handy (823392)

3. To test the configuration, for the Select Nodes parameters:



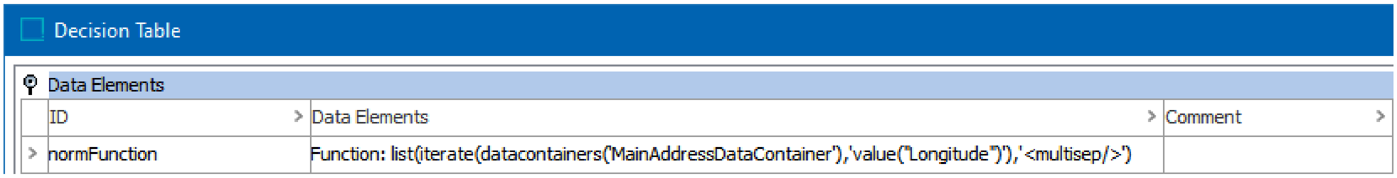
Select Nodes: amanda hailey (822553) | Amanda Handy (823392)

Normalizers	First Node Result	Second Node Result
normFunction	-79.411580	-81.206880

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

4. Click **OK** to save and display the configuration in the Data Elements flipper.



ID	Data Elements	Comment
normFunction	Function: list(iterate(datacontainers('MainAddressDataContainer'),'value("Longitude")'),'<multisep/>')	

Data Element: JavaScript Function

The JavaScript function allows normalized values to be produced entirely via a JavaScript function, including the use of external libraries through dependencies.

Important: Using a business function instead of a JavaScript function, allows code to be separated from configuration and allows complex logic to be reused across multiple matching algorithms.

Input

Use the bind 'Match Expression Context' to access the output of other data elements or bind variables. Use the evaluate(String expressionID) method.

Output

Although the output type of a JavaScript function is not required, the type must be known by downstream expressions.

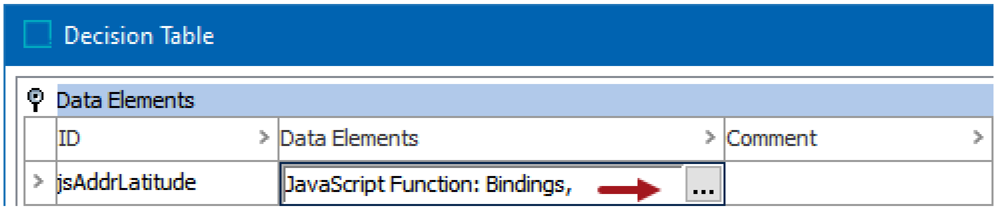
Functionality

The JavaScript function is constructed around the Match Expression Context. For more information, refer to **Match Expressions and Match Expression Context** topic.

Configuring a JavaScript Function Data Element

After adding the JavaScript function in the Data Elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

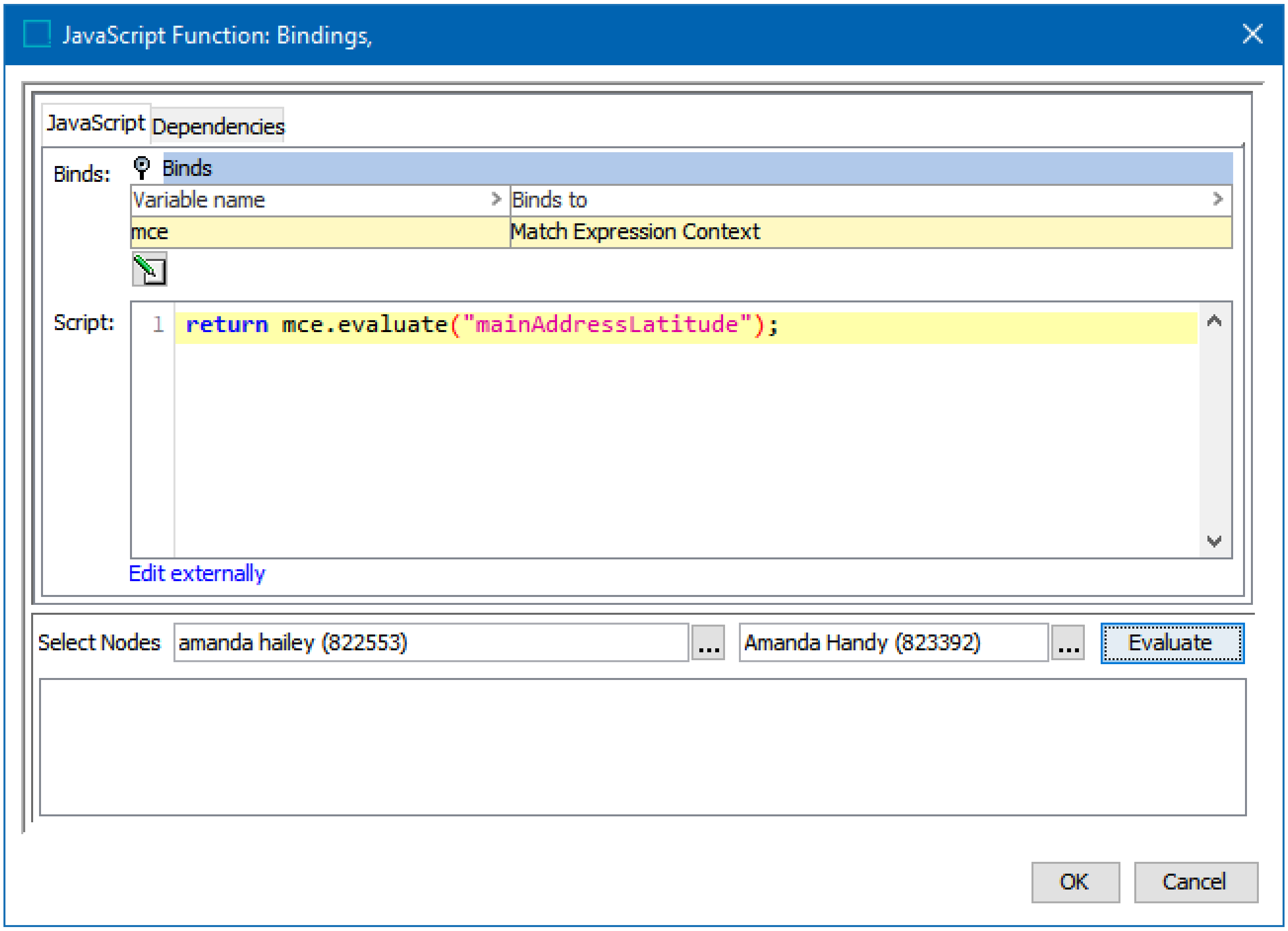
1. Click into the Data Elements column and click the ellipsis button (...) to access the configuration dialog.



ID	Data Elements	Comment
jsAddrLatitude	JavaScript Function: Bindings, ...	

2. On the JavaScript Function dialog, add the function data.

- For **Binds**, click the **Edit** button (✎) to add the necessary binds for your JavaScript.
- For **Script**, add your JavaScript code.



Binds:

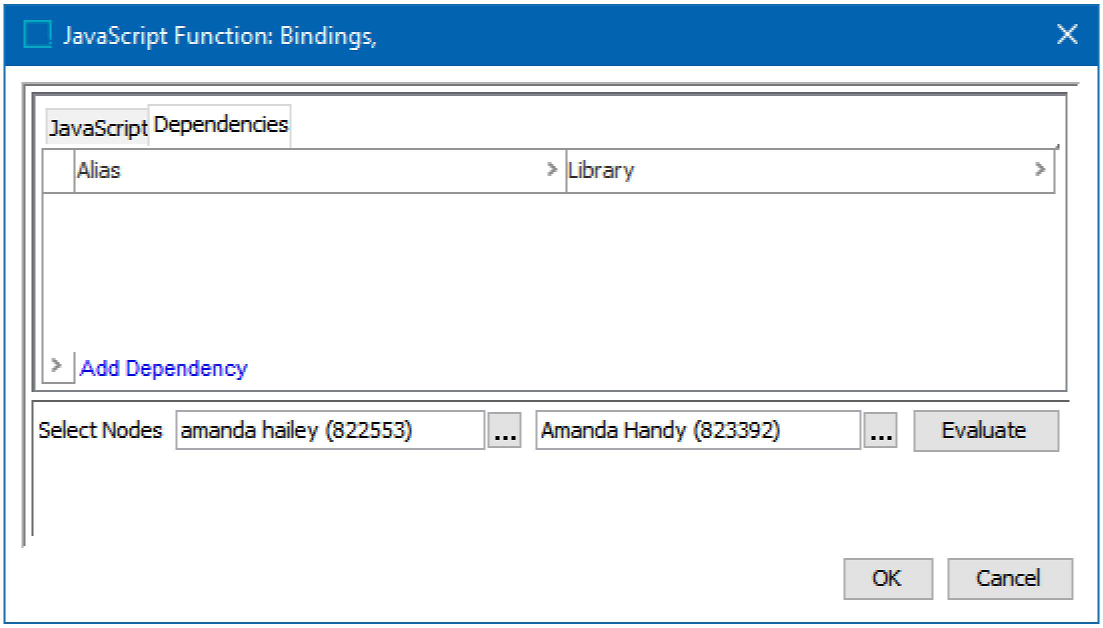
Variable name	Binds to
mce	Match Expression Context

Script:

```
return mce.evaluate("mainAddressLatitude");
```

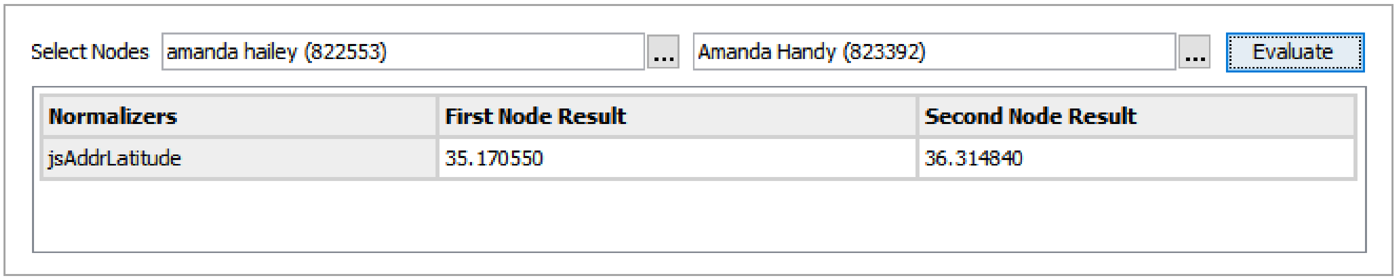
Select Nodes: amanda hailey (822553) ... Amanda Handy (823392) ...

3. Click the Dependencies tab, click the **Add Dependency** link to select any libraries required for the script added on the JavaScript tab.



Select Nodes: amanda hailey (822553) | Amanda Handy (823392)

4. To test the configuration, for the Select Nodes parameters:



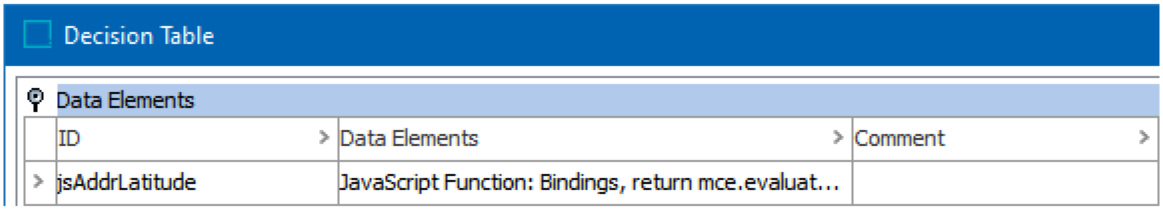
Select Nodes: amanda hailey (822553) | Amanda Handy (823392)

Normalizers	First Node Result	Second Node Result
jsAddrLatitude	35.170550	36.314840

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

5. Click **OK** to save and display the configuration in the Data Elements flipper.



ID	Data Elements	Comment
jsAddrLatitude	JavaScript Function: Bindings, return mce.evaluat...	

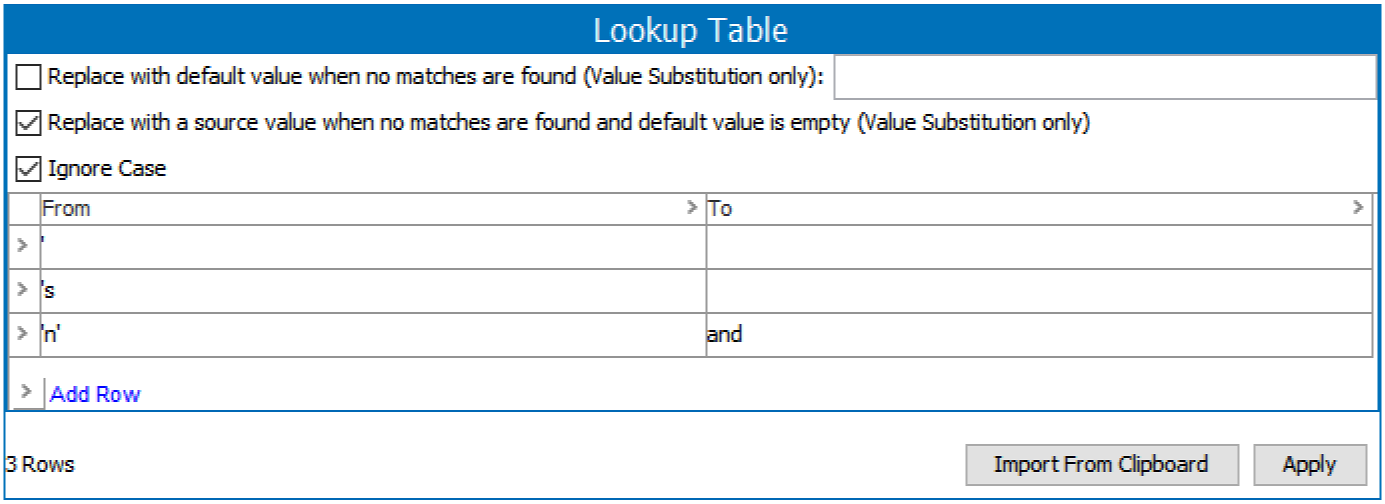
Data Element: Organization Name Normalizer

An organization name normalizer can normalize organization name data for use in the corresponding organization name matcher.

Considerations

As needed, create the following:

- **Replacement String Lookup Table -** This lookup table should account for inconsistencies in organization names by defining semantically equivalent strings (especially in the usage of apostrophes and quotation marks). It is often a good way to remove accent, quotation and apostrophe characters and normalize non-Latin characters. Refer to the **Transformation Lookup Tables** topic in the **Resource Materials** section of online help. For example:



Lookup Table

☐ Replace with default value when no matches are found (Value Substitution only):

☑ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

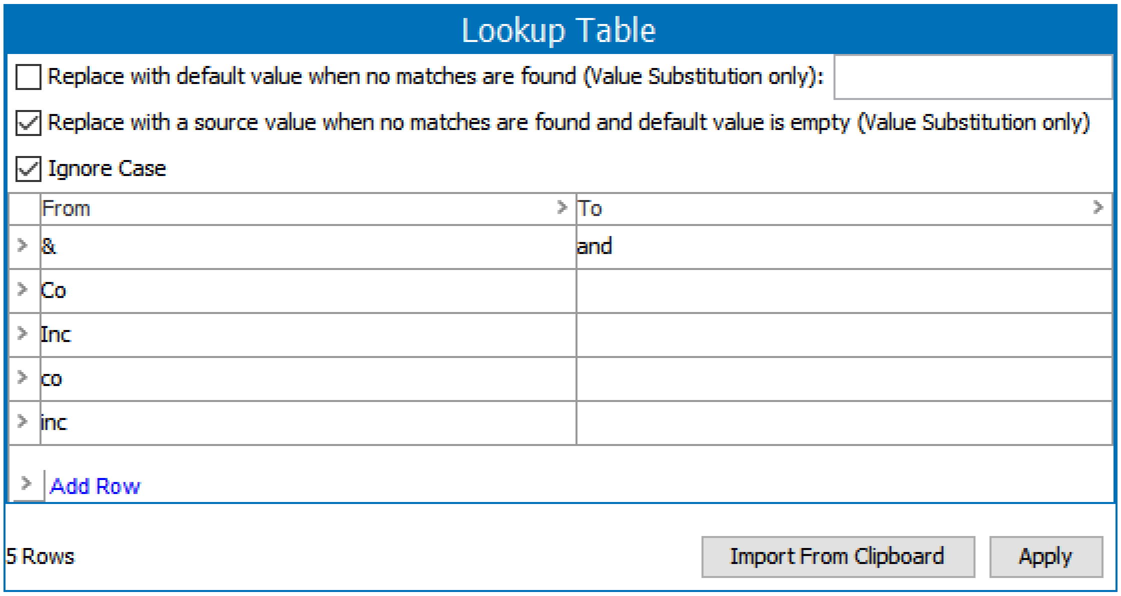
☑ Ignore Case

From	To
'	
's	
'n'	and

> **Note:** Although Transformation Lookup Tables can be manually ordered in the workbench, regardless of the order of the rows, system processing transforms punctuation first, followed by alphabetic characters. For example, one row with ' and another row with 's processes the ' first, meaning that the 's entry is not processed because the ' has been removed already.

- **Replacement Word Lookup Table -** This lookup table is used for replacing or removing parts of an organization name that forms full words, like 'Inc' or 'Co'. When the normalizer runs, it replaces entire word occurrences of a 'From' entry to the 'To' entry sequentially from the first row to the last row. Refer to the

Transformation Lookup Tables topic in the **Resource Materials** section of online help. For example:



Lookup Table

☐ Replace with default value when no matches are found (Value Substitution only):

☑ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

☑ Ignore Case

From	To
&	and
Co	
Inc	
co	
inc	

- **Name Split Regex** - The default (\s+) splits on any 'white space' character like space, tab or line change, but can be modified to split on comma, semicolons or even '<multisep/>', depending on the source data.

Input

When configuring the data element:

1. The **Organization Name Attribute** field defines an attribute to be used as input.
2. The **Input Parameters** field allows selection of:
   - 'Use Attribute on Object' – by default, this option is set to ‘True’ and indicates to read attributes on the object itself. Click the Value dropdown to manually set it to 'False' when using information from a Data Container or an Input Normalizer.
   - 'Data Container' – read attributes from the data container.
   - ‘Input Normalizer’ – read outputs from the selected Match Expression, as defined in the **Matching Algorithms and Match Expressions** topic.

Output

The output of an organization name normalizer is a java.util.Set< com.stibo.partydatamatching.domain.organizationname.OrganizationName>.

Functionality

The organization name normalizer automatically makes the following modifications to the organization name in the order listed for comparison purposes only:

1. Lower-case text

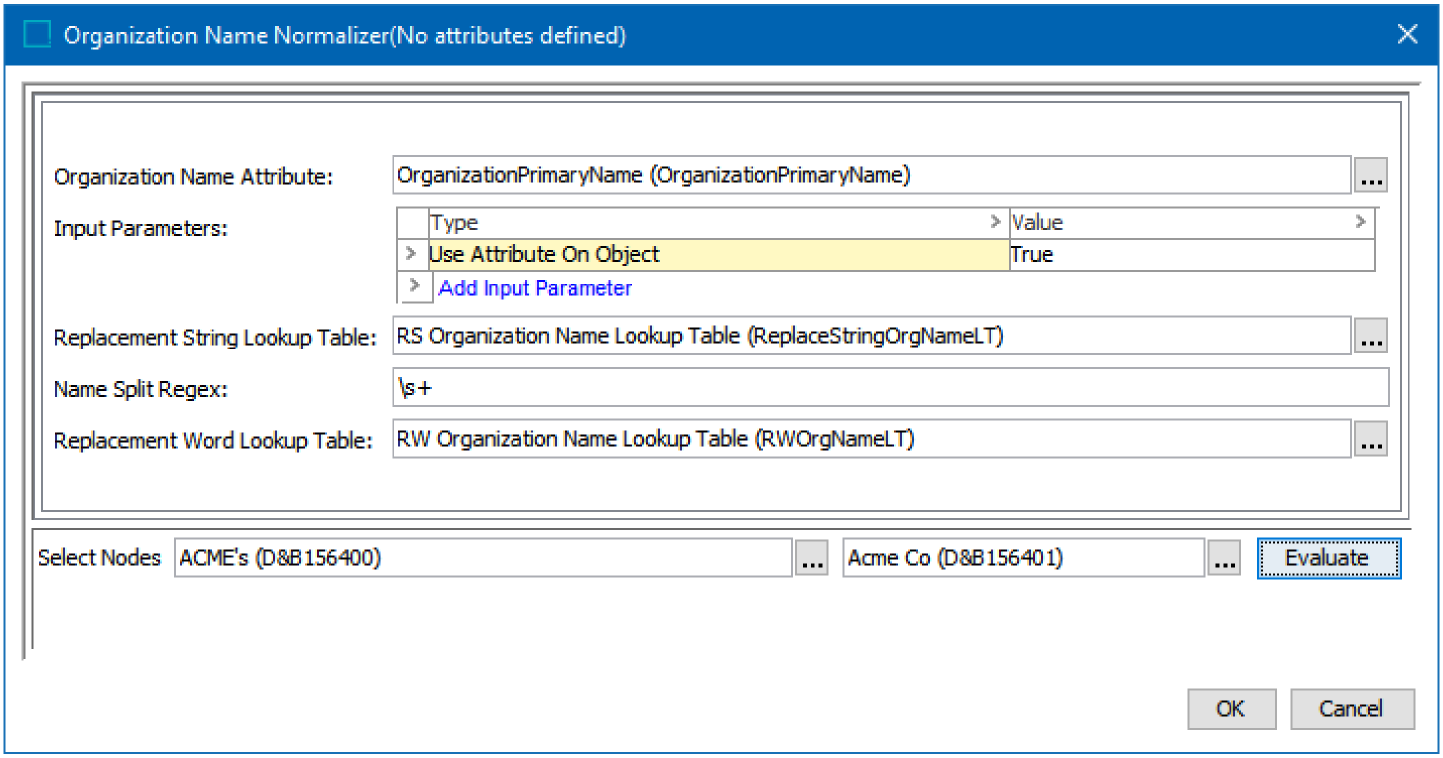
2. Apply the selected Replacement String Lookup Table - which case-insensitively replaces every substring occurrence of a 'From' entry in the Replacement String Lookup Table with the 'To' entry. Replacement is performed in the order of the table. This allows removal of characters, accents, quotations, or apostrophes. It can also be used to Romanize non-Latin characters.

3. Apply the selected Replacement Word Lookup Table, respecting the Name Split Regex as a word divider - which only makes replacements when entire words match the replacement table 'From' entry. Word divisions are defined by the Name Split Regex, to handle separation in the input by comma, <multisep/> tags, space, tabs, line feed, etc.

For example, consider the setup illustrated in the **Considerations** section:

Replacement String Lookup Table	From ['s] To [] From [‘n’] To [and]
Replacement Word Lookup Table	From [Inc] To [] From [Co] To [] From [&] To [and]
Name Split RegEx	\s+

Organization name normalization with this setup results in the following:



Organization Name Normalizer(No attributes defined)

Organization Name Attribute: OrganizationPrimaryName (OrganizationPrimaryName)

Input Parameters:

Type	Value
Use Attribute On Object	True

Replacement String Lookup Table: RS Organization Name Lookup Table (ReplaceStringOrgNameLT)

Name Split Regex: \s+

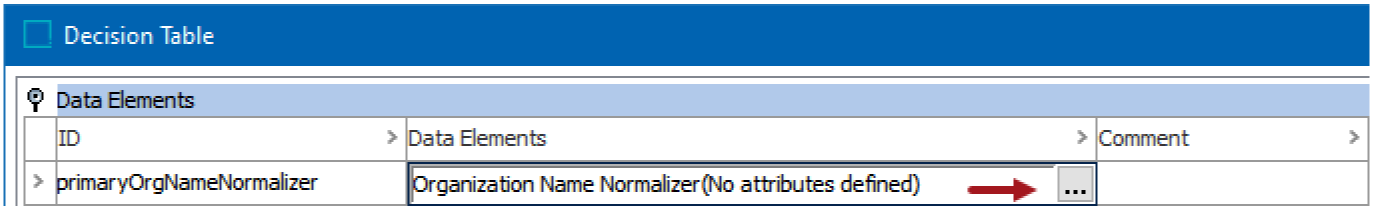
Replacement Word Lookup Table: RW Organization Name Lookup Table (RWOrgNameLT)

Select Nodes ACME's (D&B156400) Acme Co (D&B156401)

Configuring an Organization Name Normalizer Data Element

After adding the organization name normalizer in the Data Elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Data Elements column and click the ellipsis button (...) to access the configuration dialog.

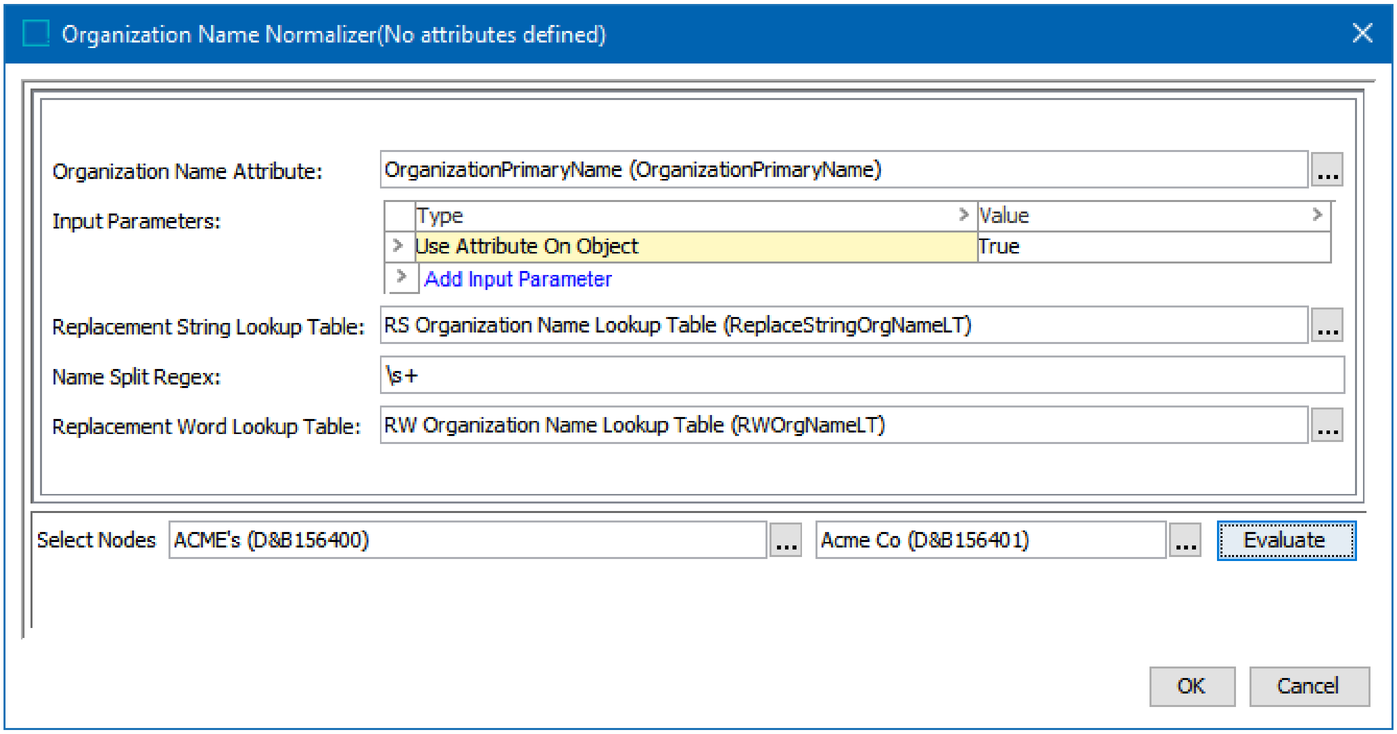


Decision Table

Data Elements

ID	Data Elements	Comment
primaryOrgNameNormalizer	Organization Name Normalizer(No attributes defined)	

2. On the Organization Name Normalizer dialog:



Organization Name Normalizer(No attributes defined)

Organization Name Attribute: OrganizationPrimaryName (OrganizationPrimaryName)

Input Parameters:

Type	Value
Use Attribute On Object	True

Add Input Parameter

Replacement String Lookup Table: RS Organization Name Lookup Table (ReplaceStringOrgNameLT)

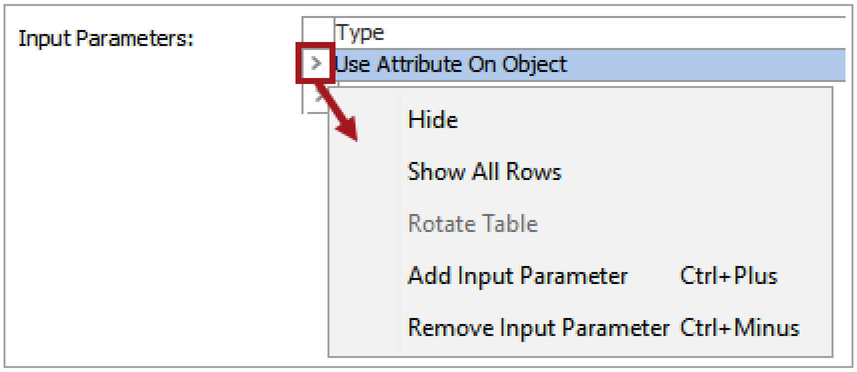
Name Split Regex: \s+

Replacement Word Lookup Table: RW Organization Name Lookup Table (RWOrgNameLT)

Select Nodes: ACME's (D&B156400) | Acme Co (D&B156401)

- For the **Organization Name Attribute**, click the ellipsis button (...) and select the organization name attribute.
- For the **Input Parameters**, define the source of the data to be normalized. Refer to the **Input** section above for details.

  Right-click the arrow in the first column of the Input Parameters table for additional display and edit options. Although it appears that the default 'Use Attribute On Object' parameter can be removed, after closing the dialog it will continue to display. Instead, if a different input parameter is used, click the Value dropdown and manually set 'Use Attribute On Object' option to 'False.'



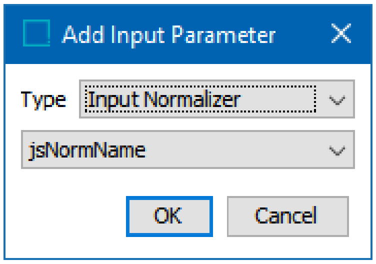
Input Parameters:

Type

Use Attribute On Object

Hide
Show All Rows
Rotate Table
Add Input Parameter Ctrl+Plus
Remove Input Parameter Ctrl+Minus

Click the **Add Input Parameter** link to add other input parameters.

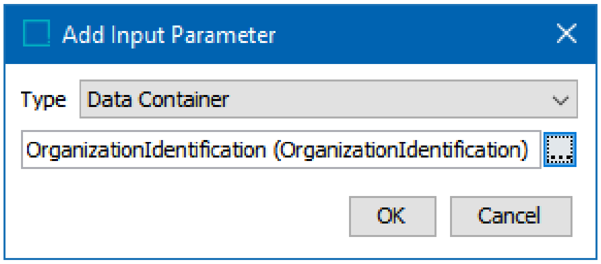


Add Input Parameter

Type: Input Normalizer

jsNormName

OK | Cancel

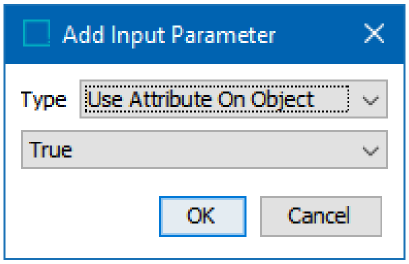


Add Input Parameter

Type: Data Container

OrganizationIdentification (OrganizationIdentification) ...

OK | Cancel



Add Input Parameter

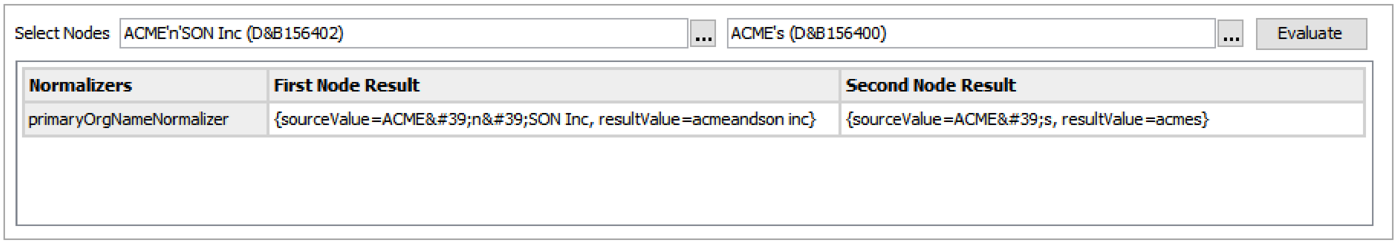
Type: Use Attribute On Object

True

OK | Cancel

- For the **Replacement String Lookup Table**, click the ellipsis button (...) and select the Transformation Lookup Table asset created as defined in the **Considerations** section above.
- For the **Name Split Regex**, click the ellipsis button (...) add regular expression to split the value of the organization name attribute into words. Leave the default (removes any whitespace character zero or more times, such as spaces, tabs, and new lines) or add your own RegEx. For more information, refer to the **Regular Expression** topic in the **Resource Materials** section of online help.
- For the **Replacement Word Lookup Table**, click the ellipsis button (...)) and select the Transformation Lookup Table asset created as defined in the **Considerations** section above.

3. To test the configuration, for the Select Nodes parameters:



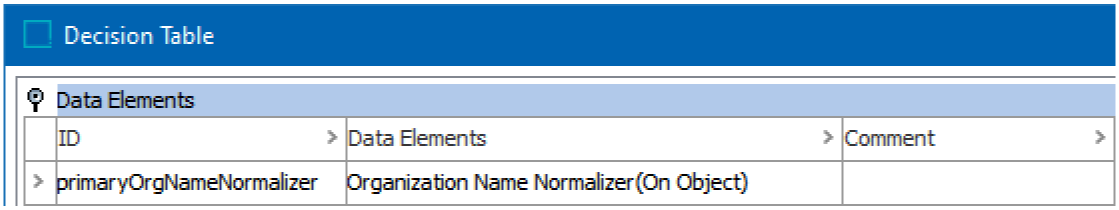
Select Nodes: ACME'n'SON Inc (D&B156402) ... | ACME's (D&B156400) ... | Evaluate

Normalizers	First Node Result	Second Node Result
primaryOrgNameNormalizer	{sourceValue=ACME&#39;n&#39;SON Inc, resultValue=acmeandson inc}	{sourceValue=ACME&#39;s, resultValue=acmes}

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

4. Click **OK** to save and display the configuration in the Data Elements flipper.



Decision Table

Data Elements

ID	Data Elements	Comment
primaryOrgNameNormalizer	Organization Name Normalizer(On Object)	

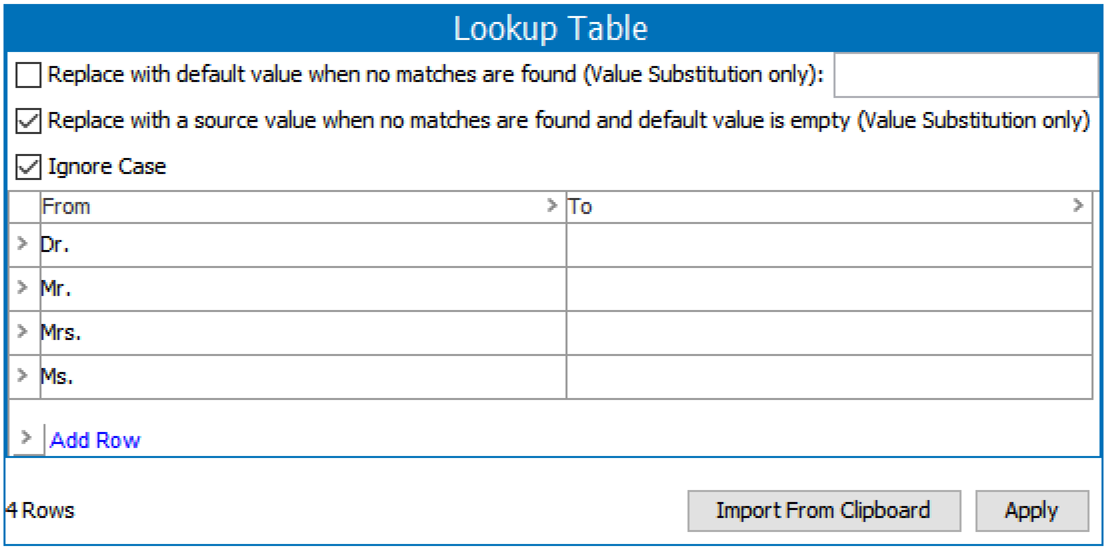
Data Element: Person Name Normalizer

A person name normalizer can normalize names of individuals for use in the corresponding Person Name Matcher.

Considerations

As needed, create the following:

- **Replacement Word Lookup Table -** This lookup table should be sensitive to diacritics in the dataset and should remove parts of a person name like 'Dr.' or 'Ms.'. When the normalizer runs, it replaces entire word occurrences of a 'From' entry to the 'To' entry sequentially from the first row to the last row. Refer to the **Transformation Lookup Tables** topic in the **Resource Materials** section of online help. For example:



Lookup Table

☐ Replace with default value when no matches are found (Value Substitution only):

☑ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

☑ Ignore Case

From	To
Dr.	
Mr.	
Mrs.	
Ms.	

4 Rows

- **Name Split Regex -** The default (\s+) splits on any 'white space' character like space, tab or line change, but can be modified to split on comma, semicolons or even '<multisep/>', depending on the source data.

Input

This data is provided by the input attributes mapped in the configuration, and includes first name, middle name and last name, which are kept separate while normalizing.

When configuring the data element:

1. The **First Name Attribute** field defines an attribute to be used as input.
2. The **Middle Name Attribute** field defines an attribute to be used as input.

3. The **Last Name Attribute** field defines an attribute to be used as input.

4. The **Input Parameters** field allows selection of:

   - 'Use Attribute on Object' – by default, this option is set to 'True' and indicates to read attributes on the object itself. Click the Value dropdown to manually set it to 'False' when using information from a Data Container or an Input Normalizer.

   - 'Data Container' – read attributes from the data container.

   - 'Input Normalizer' – read outputs from the selected Match Expression, as defined in the **Matching Algorithms and Match Expressions** topic.

Output

The output of a person name normalizer is a java.util.Set< com.stibo.partydatamatching.domain.name.Name>.

Functionality

The person name normalizer automatically makes the following modifications in the order listed to person name for comparison purposes only:

1. Lower-case text.

2. Apply the Replacement Word Lookup Table. Typically, this is used to remove unwanted words from names. For example, 'Mr.,' 'Dr.,' or 'Von.'. This happens before Unicode Canonical Decomposition, meaning the lookup table is sensible to diacritics etc. The Replacement Word Lookup Table makes use of the Name Split Regex to separate words in the names.

3. If the 'Normalize Accents' checkbox is enabled, run the Unicode Canonical Decomposition, which is described in https://www.unicode.org/reports/tr15/tr15-23.html. The most important effect of this to remove diacritics. The actual diacritics removed are those listed in the Unicode segment InCombiningDiacriticalMarks.

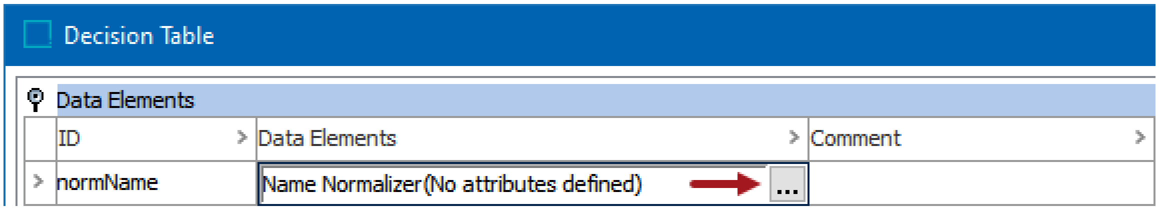
4. Removes any punctuation.

Note: Canonical Decomposition, as defined by Unicode, does a lot of work, but not all characters and substitutions may be normalized sufficiently for specific use cases. Examples are eastern Europe Ł or the Nordic Ø. Such special cases can often be solved by adding a business function normalizer in front of or after the person name normalizer that solves the specific cases. For more information, search the web.

Configuring a Person Name Normalizer Data Element

After adding the person name normalizer in the Data Elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Data Elements column and click the ellipsis button (...) to access the configuration dialog.

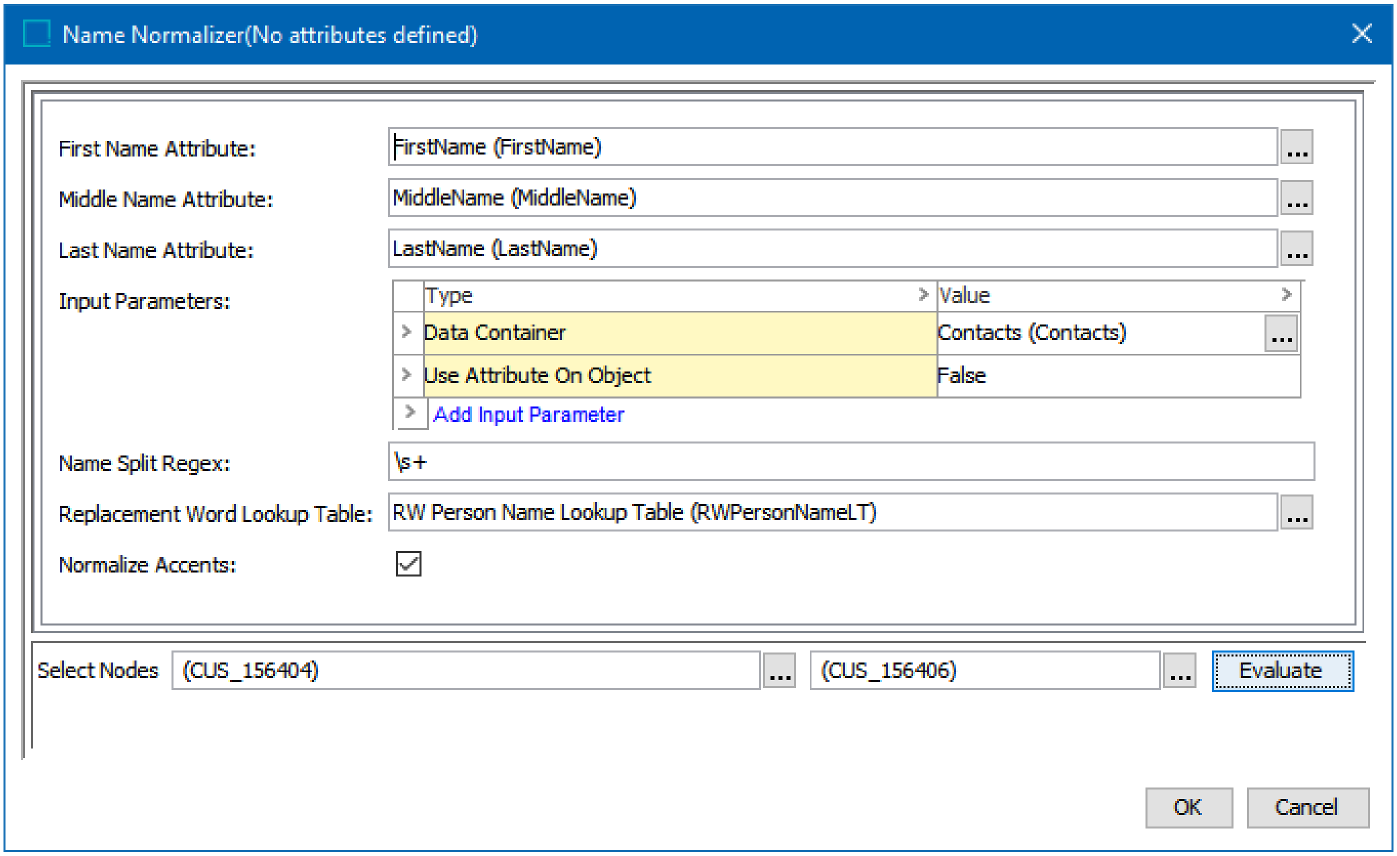


Decision Table

Data Elements

ID	Data Elements	Comment
normName	Name Normalizer(No attributes defined) ...	

2. On the Person Name Normalizer dialog:



Name Normalizer(No attributes defined)

First Name Attribute: FirstName (FirstName) ...

Middle Name Attribute: MiddleName (MiddleName) ...

Last Name Attribute: LastName (LastName) ...

Input Parameters:

Type	Value
Data Container	Contacts (Contacts) ...
Use Attribute On Object	False
Add Input Parameter	

Name Split Regex: \s+

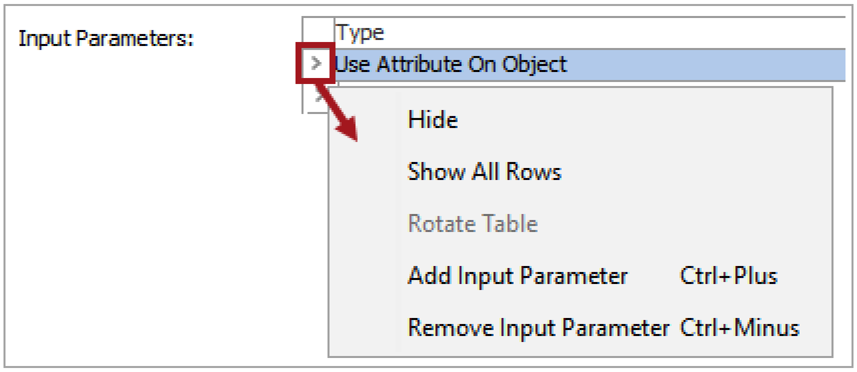
Replacement Word Lookup Table: RW Person Name Lookup Table (RWPersonNameLT) ...

Normalize Accents: ☑

Select Nodes (CUS_156404) ... (CUS_156406) ... Evaluate

- For the **First Name Attribute**, click the ellipsis button (...) and select the appropriate attribute.
- For the **Middle Name Attribute**, click the ellipsis button (...) and select the appropriate attribute.
- For the **Last Name Attribute**, click the ellipsis button (...) and select the appropriate attribute.
- For the **Input Parameters**, define the source of the data to be normalized. Refer to the **Input** section above for details.

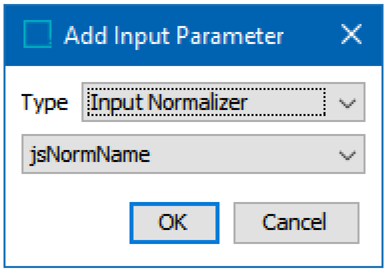
Right-click the arrow in the first column of the Input Parameters table for additional display and edit options. Although it appears that the default 'Use Attribute On Object' parameter can be removed, after closing the dialog it will continue to display. Instead, if a different input parameter is used, click the Value dropdown and manually set 'Use Attribute On Object' option to 'False.'


Input Parameters:

Type
Use Attribute On Object

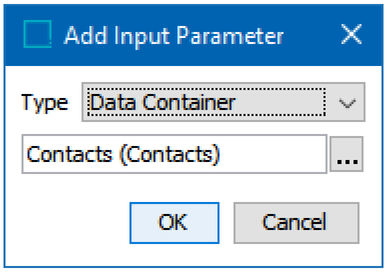
Hide
Show All Rows
Rotate Table
Add Input Parameter Ctrl+Plus
Remove Input Parameter Ctrl+Minus

Click the **Add Input Parameter** link to add other input parameters. Refer to the **Input** section above for details.


Add Input Parameter

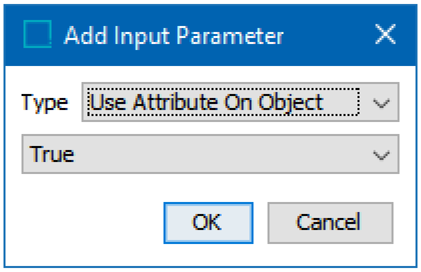
Type: Input Normalizer

jsNormName


Add Input Parameter

Type: Data Container

Contacts (Contacts)

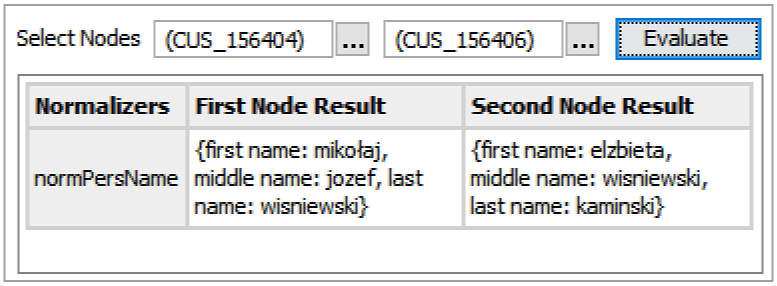

Add Input Parameter

Type: Use Attribute On Object

True

- For the **Replacement String Lookup Table**, click the ellipsis button (...) and select the transformation lookup table asset created as defined in the **Considerations** section above.
- For the **Name Split Regex**, add a regular expression to split the value of the first name, middle name, and last name into words. This allows the replacement table to remove a 'Mr.' included in a name field. Leave the default (removes any whitespace character zero or more times, such as spaces, tabs, and new lines) or add your own RegEx. For more information, refer to the **Regular Expression** topic in the **Resource Materials** section of online help.
- For the **Replacement Word Lookup Table**, click the ellipsis button (...) and select the transformation lookup table asset created as defined in the **Considerations** section above.
- For the **Normalize Accents** checkbox, check to run the Unicode Normalization Forms defined in the **Functionality** section above.

3. To test the configuration, for the Select Nodes parameters:

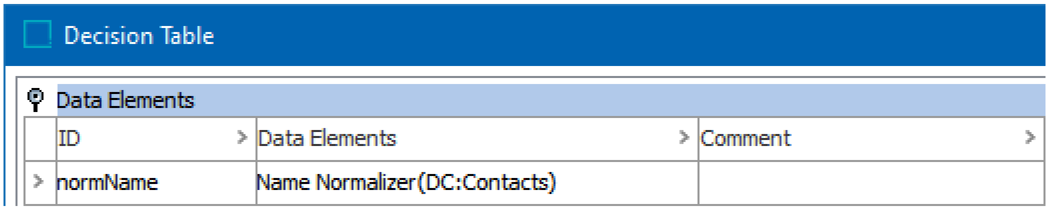


Normalizers	First Node Result	Second Node Result
normPersName	{first name: mikołaj, middle name: jozef, last name: wisniewski}	{first name: elzbieta, middle name: wisniewski, last name: kaminski}

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

4. Click **OK** to save and display the configuration in the Data Elements flipper.



ID	Data Elements	Comment
normName	Name Normalizer(DC:Contacts)	

Data Element: Phone Normalizer

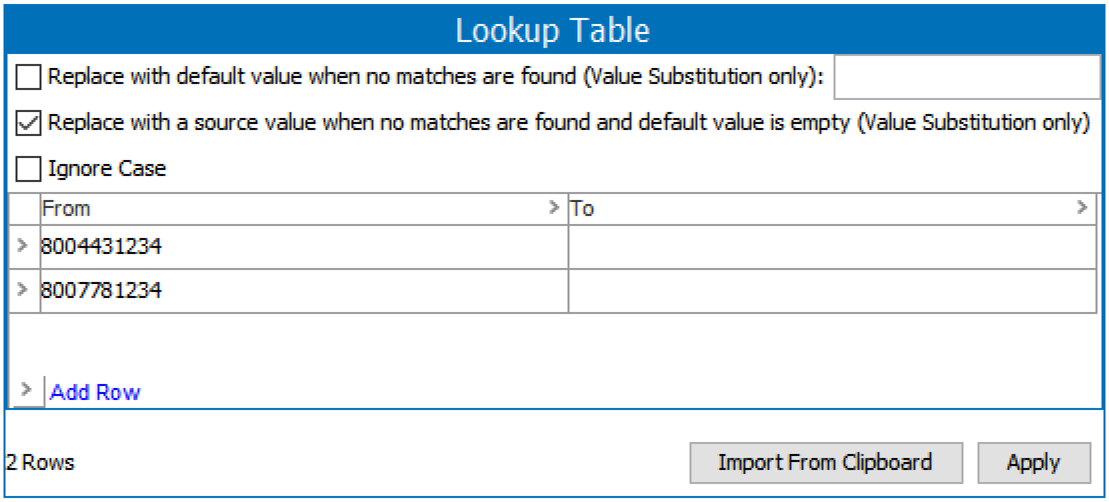
A phone normalizer can normalize phone numbers for use in the corresponding phone matcher.

Considerations

As needed, create the following:

- **Replacement Lookup Table -** This lookup table should remove invalid phone number values, like main phone numbers for close business partners that are often shared between hundreds of contact persons. When the normalizer runs, it replaces occurrences of a 'From' entry with the 'To' entry sequentially from the first row to the last row. Refer to the **Transformation Lookup Tables** topic in the **Resource Materials** section of online help.

  For example, the phone numbers included in this lookup table are known main numbers for companies in the data being matched:



□ Replace with default value when no matches are found (Value Substitution only):

☑ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

□ Ignore Case

From	To
8004431234	
8007781234	

Input

Phone numbers are provided via 'Input Attribute' in the phone normalizer configuration, including formatting such as +, (,), -, as well as phone extension numbers.

The phone number is normalized according to a set of regional rules, determined by the phone country code if it exists, secondary from a normalized address Standardized Country ISO Code or Input Country ISO Code, tertiary by the rules of the default country code as set on the phone normalizer, or finally, as a US phone number.

When configuring the data element:

1. The **Input Attribute** field defines an attribute to be used as input.
2. The **Input Parameters** field allows selection of:
   - 'Use Attribute on Object' – by default, this option is set to 'True' and indicates to read attributes on the object itself. Click the Value dropdown to manually set it to 'False' when using information from a Data Container or an Input Normalizer.
   - 'Data Container' – read attributes from the data container.
   - 'Input Normalizer' – read outputs from the selected Match Expression, as defined in the **Matching Algorithms and Match Expressions** topic.

Output

The output of a phone normalizer is a java.util.Set<com.stibo.partydatamatching.domain.phone.Phone>.

Functionality

The phone normalizer automatically makes the following modifications to a phone number for comparison purposes only in the order listed:

1. **Applies the Replacement Lookup Table** – This is used to remove invalid phone number values, like main phone numbers for close business partners, that are often shared between hundreds of contact persons.
2. **Look for a number in the input text** – The normalization is quite lenient and looks for a number in the input text. It ignores punctuation and white space, as well as any text before the number (e.g., a leading "Tel: ") and trims the non-number characters. It accepts a number in any format (E164, national, international, etc.), assuming it can be interpreted with the country code that is supplied. It also attempts to convert any alpha characters into digits for vanity numbers of the type '1800 ACMECORP'. The input number can contain formatting such as +, () and -, as well as a phone number extension. The normalization can also handle numbers provided in RFC3966 format.

The phone normalizer parses the phone number according to rules for a region, so a country code is needed. The following ordered attempts are made to identify the country code:

1. The normalizer looks for a '+ country code' in the phone number.
2. The normalizer uses the main address country code. This country code must be provided using CLDR two-letter region-code format, and from STEP release 11.0, the phone normalizer can get this from an address normalizer data element (add the data element ID in the main address input).
   - If the address is standardized, the phone normalizer uses the country code from the Standardized Country ISO Code as defined by the address component model.
   - If the address is not standardized, the phone normalizer uses the country code from the Default Country ISO Code.

3. The normalizer uses the default country code set in the normalizer.

4. The normalizer attempts to interpret the number as a US phone number.

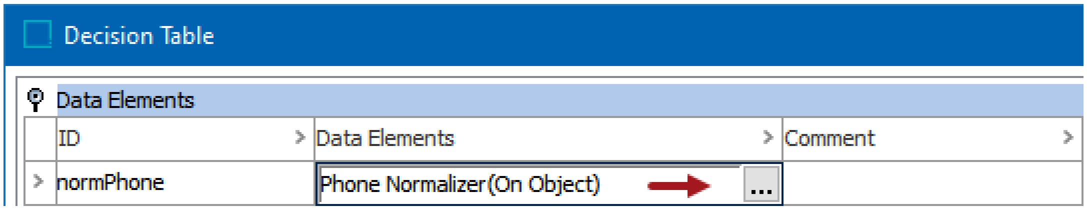
In the examples below, the Brazil phone number is stripped of the long-distance marker (0) and the carrier selection code. The phone letter prefix from RFC3966 is stripped for different languages. Punctuation and parentheses are stripped. The country code is normalized and can be handled externally to the subscriber number.

Input Phone Number	Country Code Provided	Normalized Country Code and Phone Number
+55 0 15 21 5555-5555	BR	countryCode: 55, nationalNumber: 2155555555
tel:+55-00-98765-4321	BR	countryCode: 55, nationalNumber: 987654321
(+45) 9999 9999	DK	countryCode: 45, nationalNumber: 99999999
tlf (45) 99.99,99-99	DK	countryCode: 45, nationalNumber: 99999999
+81 3-3666-7195	JP	countryCode: 81, nationalNumber: 336667195
(03) 1234-5678	JP	countryCode: 81, nationalNumber: 312345678
+8103123456789	JP	countryCode: 81, nationalNumber: 3123456789
+18016954248	US	countryCode: 1, nationalNumber: 8016954248
+1 801.695.4258	US	countryCode: 1, nationalNumber: 8016954248
(890) 514-7258	US	countryCode: 1, nationalNumber: 8016954258
+55 (643) 958-6877	US	countryCode: 55, nationalNumber: 6439586877

Configuring a Phone Normalizer Data Element

After adding the phone normalizer in the Data Elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Data Elements column and click the ellipsis button (**...**) to access the configuration dialog.

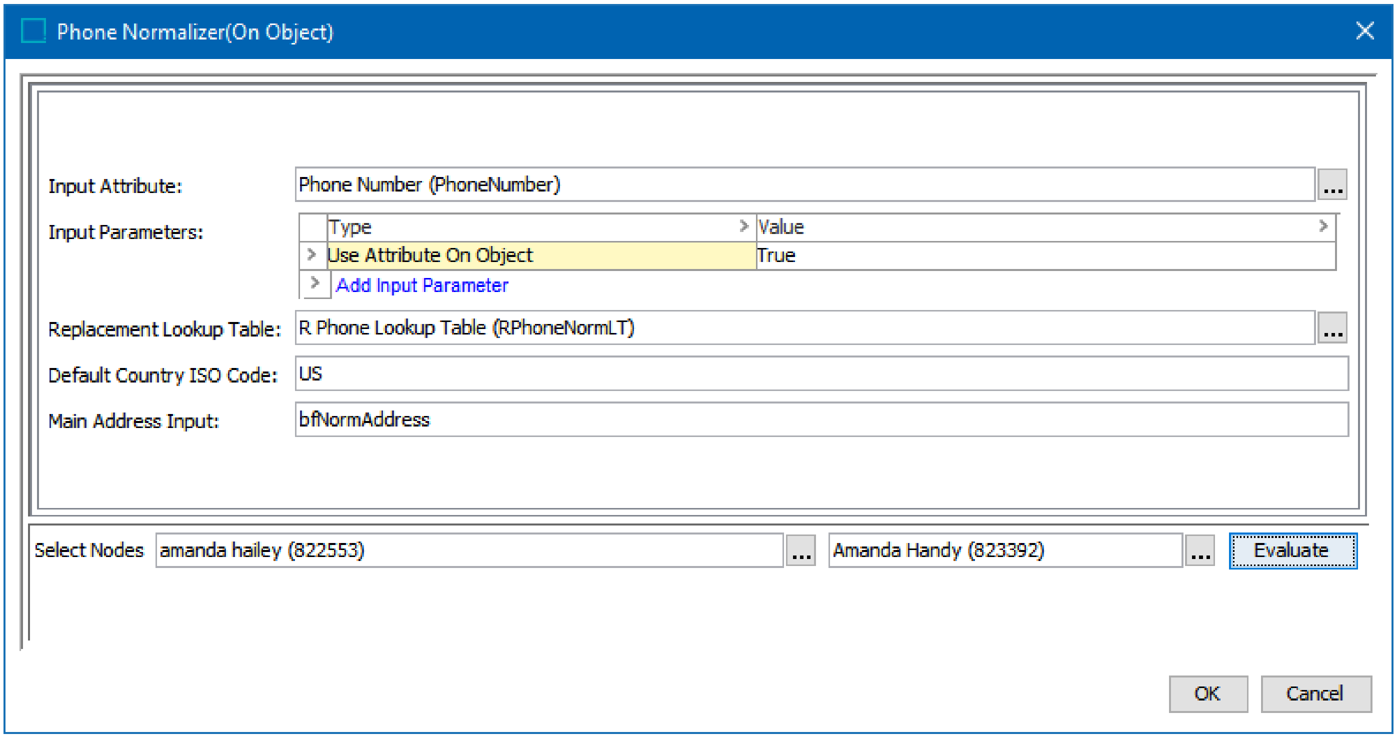


Decision Table

Data Elements

ID	Data Elements	Comment
normPhone	Phone Normalizer(On Object) ...	

2. On the Phone Normalizer dialog:



Phone Normalizer(On Object)

Input Attribute: Phone Number (PhoneNumber) ...

Input Parameters:

Type	Value
Use Attribute On Object	True
Add Input Parameter	

Replacement Lookup Table: R Phone Lookup Table (RPhoneNormLT) ...

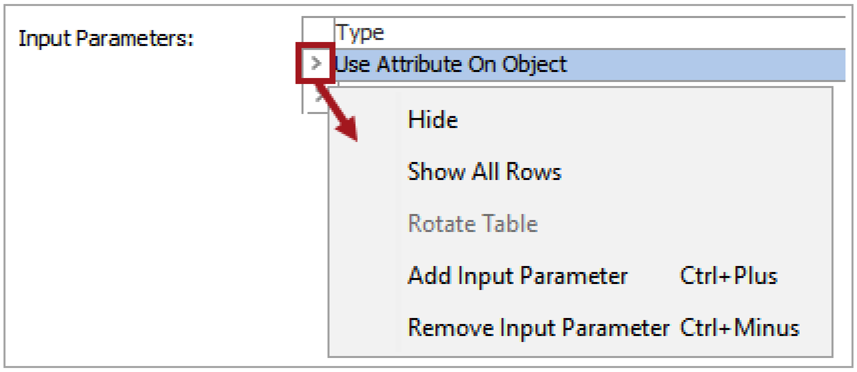
Default Country ISO Code: US

Main Address Input: bfNormAddress

Select Nodes amanda hailey (822553) ... Amanda Handy (823392) ... Evaluate

- For the **Input Attribute**, click the ellipsis button (**...**) and select the phone number attribute.
- For the **Input Parameters**, define the source of the data to be normalized. Refer to the **Input** section above for details.

  Right-click the arrow in the first column of the Input Parameters table for additional display and edit options. Although it appears that the default 'Use Attribute On Object' parameter can be removed, after closing the dialog it will continue to display. Instead, if a different input parameter is used, click the Value dropdown and manually set 'Use Attribute On Object' option to 'False.'

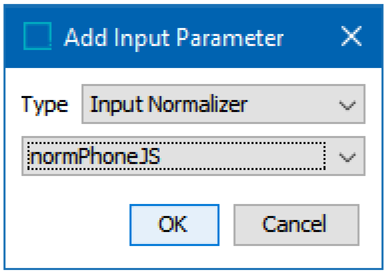


Input Parameters:

Type: Use Attribute On Object

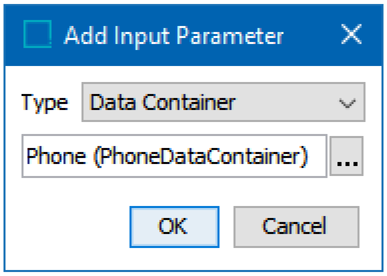
Hide
Show All Rows
Rotate Table
Add Input Parameter Ctrl+Plus
Remove Input Parameter Ctrl+Minus

Click the **Add Input Parameter** link to add other input parameters.



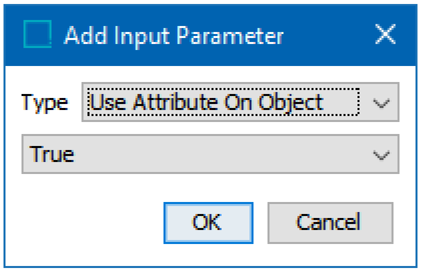
Type: Input Normalizer

normPhoneJS



Type: Data Container

Phone (PhoneDataContainer)

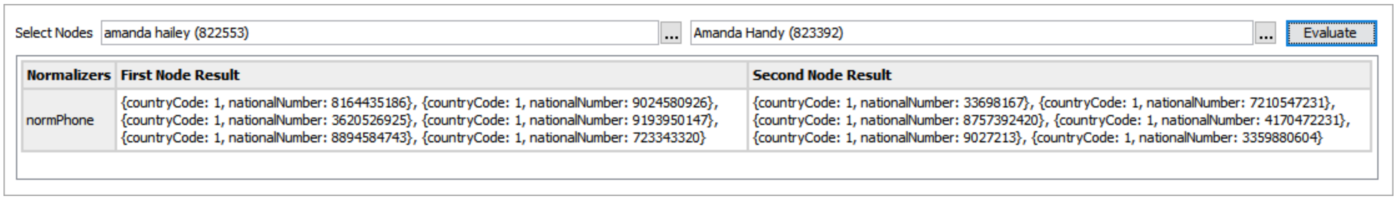


Type: Use Attribute On Object

True

- For the **Replacement Lookup Table**, click the ellipsis button (...) and select the transformation lookup table asset.
- For the **Default Country ISO Code**, enter a two-letter ISO code string to be used when no country code is included in the phone number being normalized.
- For the **Main Address Input**, optionally enter the ID of an Address Normalizer written in JavaScript that outputs a Country ISO Code. This output value is used in place of the Default ISO Code, if one exists.

3. To test the configuration, for the Select Nodes parameters:



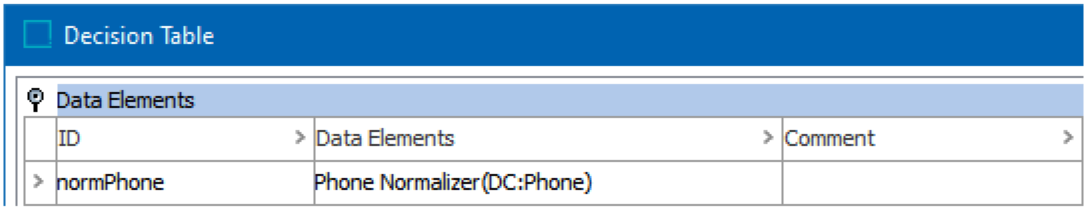
Select Nodes: amanda hailey (822553) ... Amanda Handy (823392) ...

Normalizers	First Node Result	Second Node Result
normPhone	{countryCode: 1, nationalNumber: 8164435186}, {countryCode: 1, nationalNumber: 9024580926}, {countryCode: 1, nationalNumber: 3620526925}, {countryCode: 1, nationalNumber: 9193950147}, {countryCode: 1, nationalNumber: 8894584743}, {countryCode: 1, nationalNumber: 723343320}	{countryCode: 1, nationalNumber: 33698167}, {countryCode: 1, nationalNumber: 7210547231}, {countryCode: 1, nationalNumber: 8757392420}, {countryCode: 1, nationalNumber: 4170472231}, {countryCode: 1, nationalNumber: 9027213}, {countryCode: 1, nationalNumber: 3359880604}

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

4. Click **OK** to save and display the configuration in the Data Elements flipper.



ID	Data Elements	Comment
normPhone	Phone Normalizer(DC:Phone)	

Data Element: Words Normalizer

A words normalizer can normalize attribute data for use in the corresponding words matcher. The words normalizer is often used as the first normalizer in a chain.

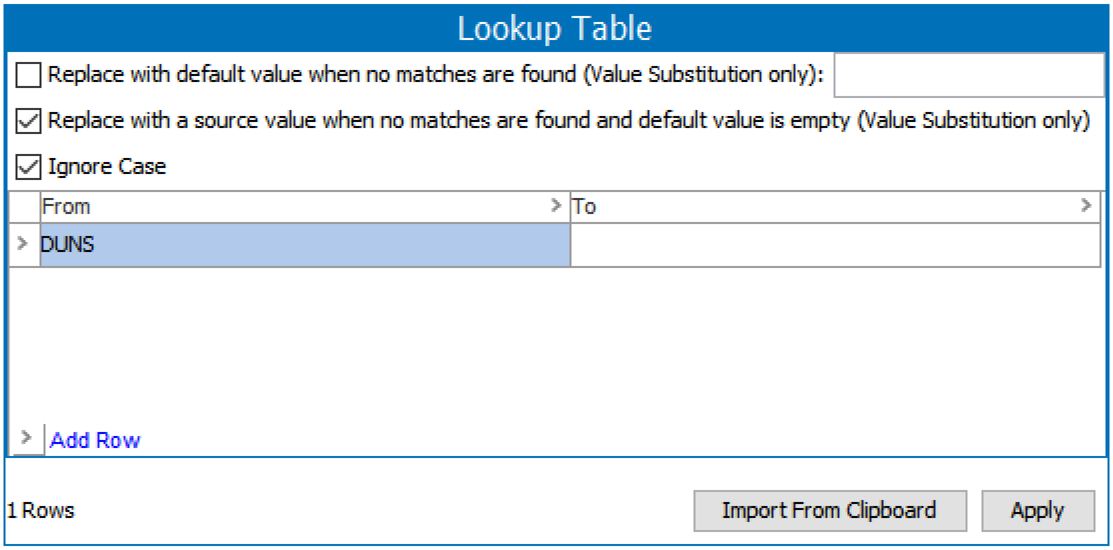
An example could be matching on IDs, like DUNS number, tax ID, social security number, insurance ID, etc. To illustrate:

- Matching on DUNS numbers would start with the words normalizer replacement table removing unwanted characters and substrings like the 'DUNS' prefix. It might be necessary to chain a business function after the words normalizer to remove prefixed zeros.
- Matching on Insurance ID, the words normalizer replacement table could remove # or 'ID' prefixes. The words normalizer can also replace '-' and tab separations in the number with simple spaces.

Considerations

As needed, create the following:

- **Replacement Word Lookup Table** - This lookup table can remove '#' or 'ID' prefixes. When the normalizer runs, it replaces entire word occurrences of a 'From' entry to the 'To' entry sequentially from the first row to the last row. Refer to the **Transformation Lookup Tables** topic in the **Resource Materials** section of online help. For example:



☐ Replace with default value when no matches are found (Value Substitution only):

☑ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

☑ Ignore Case

From	To
DUNS	

1 Rows

- **Name Split Regex** - The default (\s+) will split the name on any “white space” character like space, tab or line change. This can be changed to split on comma, semicolons or even “<multisep/>”, depending on the source data. For more information, refer to the **Regular Expression** topic in the **Resource Materials** section of online help.

Input

When configuring the data element:

1. The **Input Attributes** field defines all attributes to be used as inputs.
2. The **Input Parameters** field allows selection of:
   - 'Use Attribute on Object' – by default, this option is set to ‘True’ and indicates to read attributes on the object itself. Click the Value dropdown to manually set it to 'False' when using information from a Data Container or an Input Normalizer.
   - 'Data Container' – read attributes from the data container.
   - ‘Input Normalizer’ – read outputs from the selected Match Expression, as defined in the **Matching Algorithms and Match Expressions** topic.

Output

The output of a words normalizer is a java.util.List<java.lang.String>

Functionality

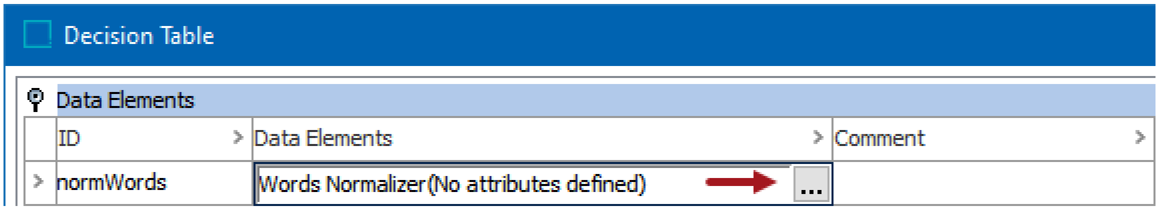
The words normalizer normalizes the output of the selected attributes in the order listed:

1. Apply the selected Replacement Word Lookup Table without using the selected 'Word Splitting Regex For Replacement Word'
2. Run the Word Splitting Regex For Replacement Word to split each input value into individual word-strings, trim leading and trailing spaces, and run the Replacement Word Lookup Table for each word-string. The word-strings are lower-cased, then appended together, separated by space characters, which results in one output string for every input string.

Configuring a Words Normalizer Data Element

After adding words normalizer in the Data Elements flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Data Elements column and click the ellipsis button (...) to access the configuration dialog.

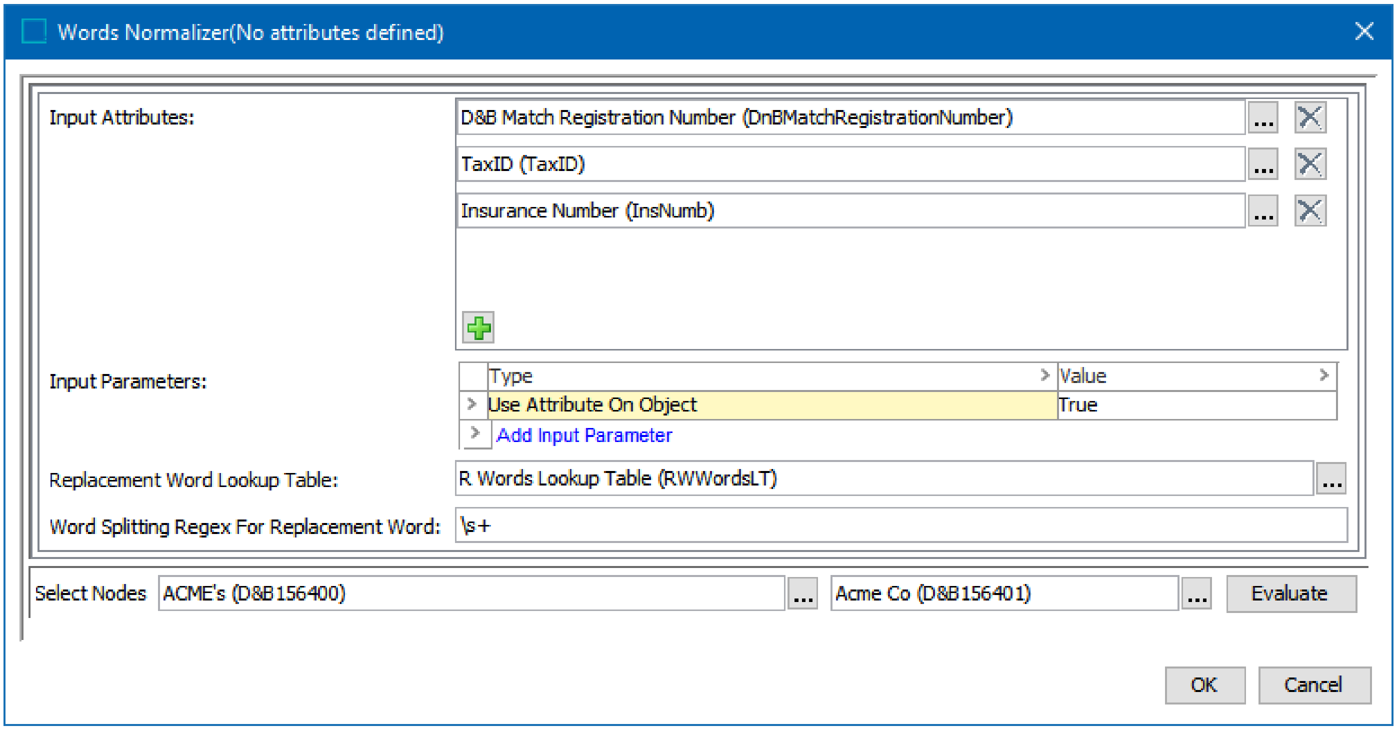


Decision Table

Data Elements

ID	Data Elements	Comment
normWords	Words Normalizer(No attributes defined)	

2. On the Words Normalizer dialog:



Words Normalizer(No attributes defined)

Input Attributes:
- D&B Match Registration Number (DnBMatchRegistrationNumber)
- TaxID (TaxID)
- Insurance Number (InsNumb)

Input Parameters:

Type	Value
Use Attribute On Object	True
Add Input Parameter	

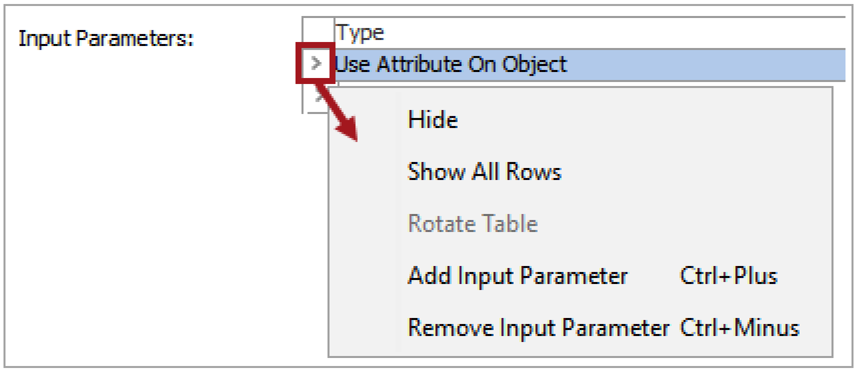
Replacement Word Lookup Table: R Words Lookup Table (RWWordsLT)

Word Splitting Regex For Replacement Word: \s+

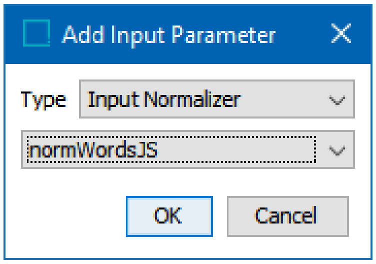
Select Nodes ACME's (D&B156400) ... Acme Co (D&B156401) ...

- For the **Input Attributes**, click the plus button (+) to add a row, then click the ellipsis button (...) to select the all the attributes to be normalized.
- For the **Input Parameters**, define the source of the data to be normalized. Refer to the **Input** section above for details.

  Right-click the arrow in the first column of the Input Parameters table for additional display and edit options. Although it appears that the default 'Use Attribute On Object' parameter can be removed, after closing the dialog it will continue to display. Instead, if a different input parameter is used, click the Value dropdown and manually set 'Use Attribute On Object' option to 'False.'

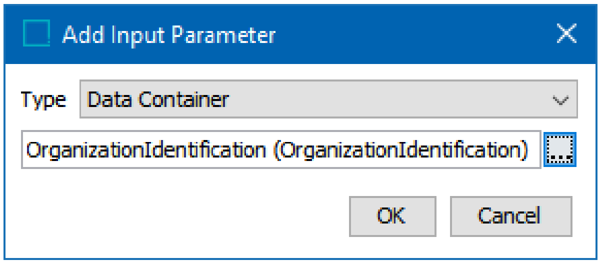


Click the **Add Input Parameter** link to add other input parameters.



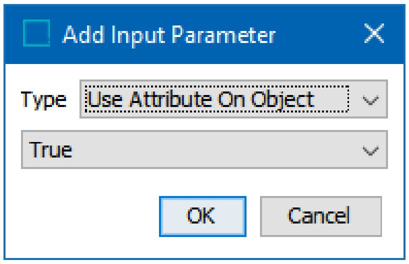
Type: Input Normalizer

normWordsJS



Type: Data Container

OrganizationIdentification (OrganizationIdentification)

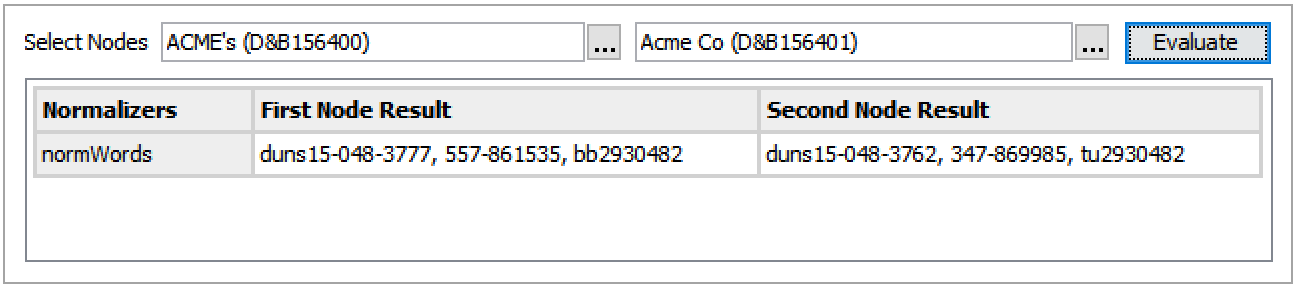


Type: Use Attribute On Object

True

- For the **Replacement Word Lookup Table**, click the ellipsis button (**...**) and select the transformation lookup table asset created as defined in the **Considerations** section above.
- For the **Word Splitting Regex for Replacement Word**, leave the default (removes any whitespace character zero or more times, such as spaces, tabs, and new lines) or add your own RegEx as defined in the **Considerations** section above.

3. To test the configuration, for the **Select Nodes** parameters:



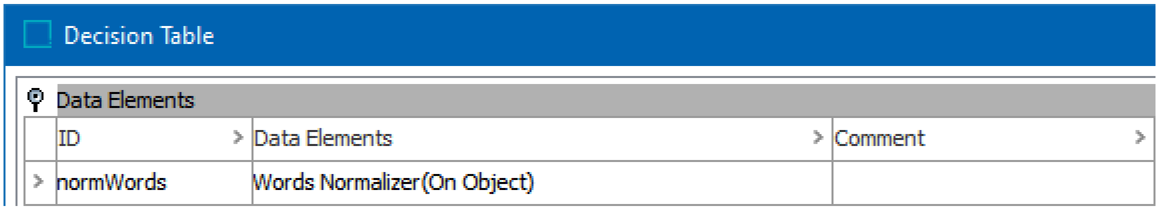
Select Nodes: ACME's (D&B156400) | Acme Co (D&B156401)

Normalizers	First Node Result	Second Node Result
normWords	duns15-048-3777, 557-861535, bb2930482	duns15-048-3762, 347-869985, tu2930482

- Click the ellipsis button (**...**) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

4. Click **OK** to save and display the configuration in the Data Elements flipper.



ID	Data Elements	Comment
normWords	Words Normalizer(On Object)	

Match Criteria Matchers

The Matchers flipper of a decision table holds matchers that compare values on two objects and produces a match score. The input values are generated by bind variables or data elements (as defined in the **Match Criteria Data Elements** topic). The match score is used in the Rules result formula to calculate the final match score (as defined in the **Match Criteria Rules** topic).

To review examples of how matching algorithm uses match scores to compare and link objects in a Golden Record Clerical Review Task List, refer to the **Potential Duplicate Match Score Examples** topic.

Matchers can be used to enable or disable Rules. For example, a rule may specify that it is only relevant if the address match score is above 70, or if address match is 'True' (depending on the inner score threshold definition of the matcher). Many matchers allow a default threshold setting to determine what is considered 'True' or 'False' in a match rule condition.

Many matchers include default weights and metrics, allowing detailed calibration of the algorithm to specific datasets. Calibration is handled during match tuning (as defined in the **Match Tuning** topic).

Matcher	Matcher Type	Object Type Allowed
Business Function	General Purpose	Entities Products
Function	General Purpose	Entities Products
JavaScript Function	General Purpose	Entities Products
Address	Preconfigured	Entities
Email	Preconfigured	Entities
Machine Learning Matcher	Preconfigured	Entities
Organization Name	Preconfigured	Entities
Person Name	Preconfigured	Entities

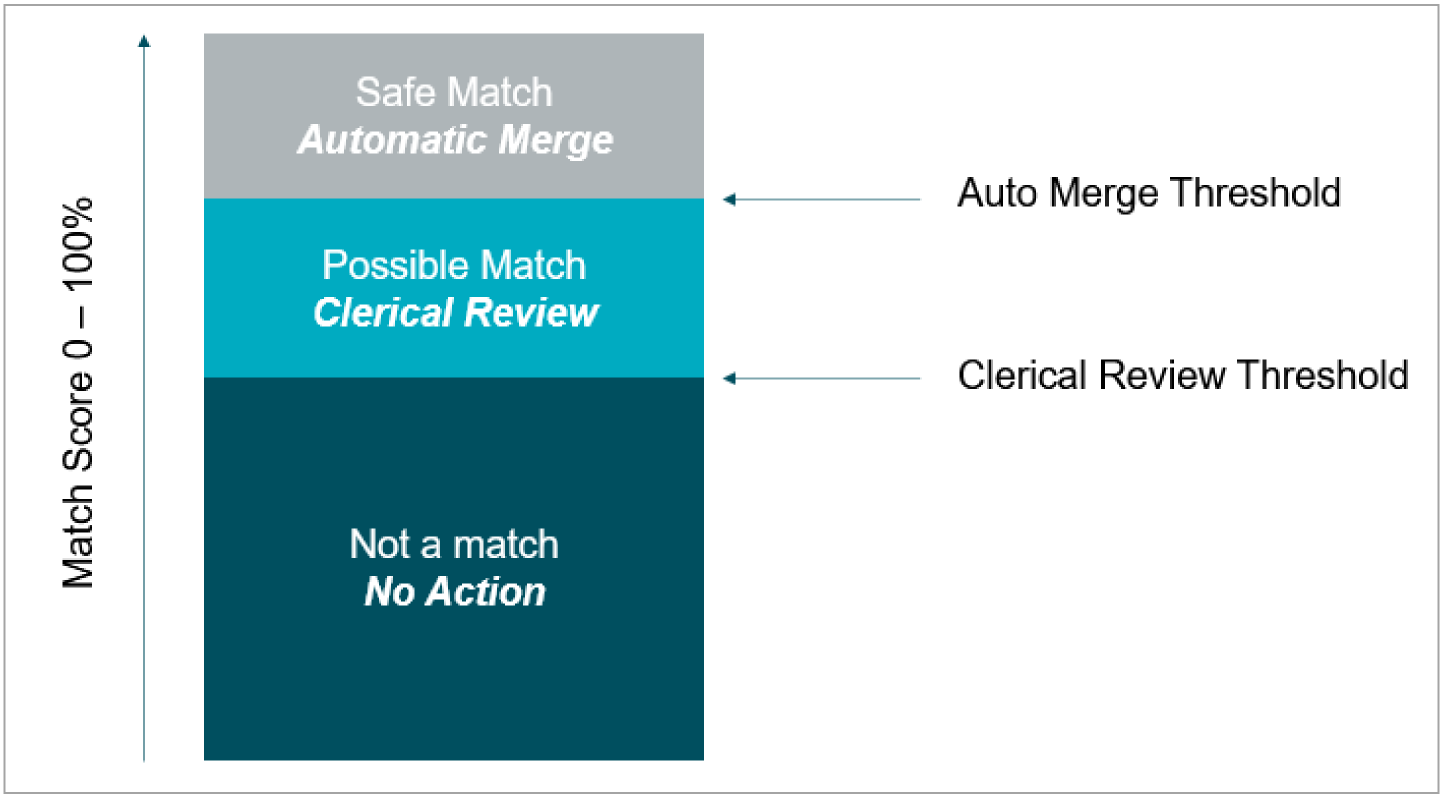
Matcher	Matcher Type	Object Type Allowed
Phone	Preconfigured	Entities
Words	Preconfigured	Entities

Match Scores

A match score is a numerical value between 0 and 100 between two golden records. Match scores are relevant in a Clerical Review Task List, where low numbers indicate a low match between a potential duplicates and the workflow node (where a workflow node is the top golden record), and 100 indicates a perfect match.

The scores determine how a record is placed in relation to the clerical review thresholds of the matching algorithm. If the value is above the upper limit of the clerical review threshold, the records are merged automatically and are removed from the task list. If the value falls below the lower limit of the clerical review threshold, no action is taken. If the value falls between the Clerical Review thresholds, the records are considered potential duplicates and appear as a task in the Clerical Review Task List for a data steward to analyze.

Note: Match scores are sometimes referred to as rank scores in documentation and within the STEP database.



Match scores are further defined in the following topics:

- **Match and Link**
- **Match and Merge**

For an illustration of how match scores affect two potential duplicates in a match and merge solution, refer to the **Potential Duplicate Match Scores Examples** topic.

Matcher: Address

The Address Matcher compares the normalized address data of two objects and generates a match score (sometimes referred to as a 'rank score') based on the weighted sum of relevant data elements and match factors.

The Address Matcher defines weights and factors that will be applied to the match score based on different criteria. As described in this documentation, these allow you to adjust the address matching score to fit specific use cases.

Input

The Address Matcher takes input from the selected address data element and retrieves data for the two objects under comparison.

The intent is to use an Address Normalizer data element (as defined in the **Data Element: Address Normalizer** topic) to normalize address data and use that data element as input to the address matcher.

Functionality

The Address Matcher considers every address in the set of input addresses of the first object and compares each of those with the set of addresses of the second object. The final score of the address matcher will be the highest score of the comparisons.

The comparison of each set of two addresses includes:

1. Using the Street Word Splitter Regex to split the Street attribute value to create street-tokens
2. Separating street-tokens into number-street-tokens and text-street-tokens
3. Defining internal temporary scores for the comparison:
   - Text Score – All text-street-tokens from the first address are paired up with all text-street-tokens on the second address. These pairings try to find exact matches, and if that is not possible, attempts to match within an edit distance. After these pairings, unmatched text-street-tokens and paired text-street-tokens where the order is different receive further penalties to the score. (Edit distance adjusts for a few different characters due to typographical errors and is only applied when the text-street-tokens are at least three (3) characters long.)
   - Number Score – Calculated exactly like the text score but can be assigned different multiplier factors. By default, the number score weighs heavier in the final score, but this can be modified on the Advanced tab.

   > **Note:** If both comparable street addresses do not contain any numbers, a Number Score will not be evaluated. Instead, a Text Score only will result in a final Street Score.

- Street Score – Compiles the text and number scores. By default, numbers in an address are assigned a heavier weight than the text, but this can be modified on the Advanced tab.
- Postcode / City Score – Determines the score based on the following:
    - If both addresses have postal codes, the City / Postcode score is 100 if the normalized post codes are an exact match, or 0 if there is not an exact match.
    - If either address has an ISO country code of 'US', the comparison of postal codes considers only the first 5 digits.
    - If at least one address lacks a postal code, then the cities are compared. An exact city match results in City / Postcode score 100.
    - If a single insertion, deletion, or substitution of a character could make the cities equal, the City / Postcode score is 0.9—unless the city name is shorter than 5 characters, in which case the City / Postcode score is heavily penalized according to the actual length of the name.

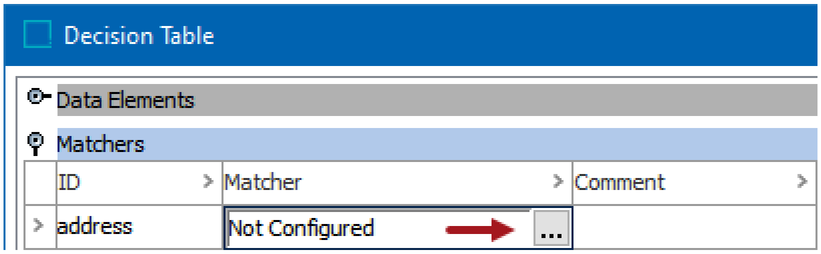
Refer to the **Token and Scoring Examples** section at the end of this topic for a detailed explanation.

> **Important:** The Address Matcher performs best when each customer has fewer than 100 addresses. For example, comparing organization customers with many addresses results in a lot of comparisons and can degrade performance.

Configuring an Address Matcher

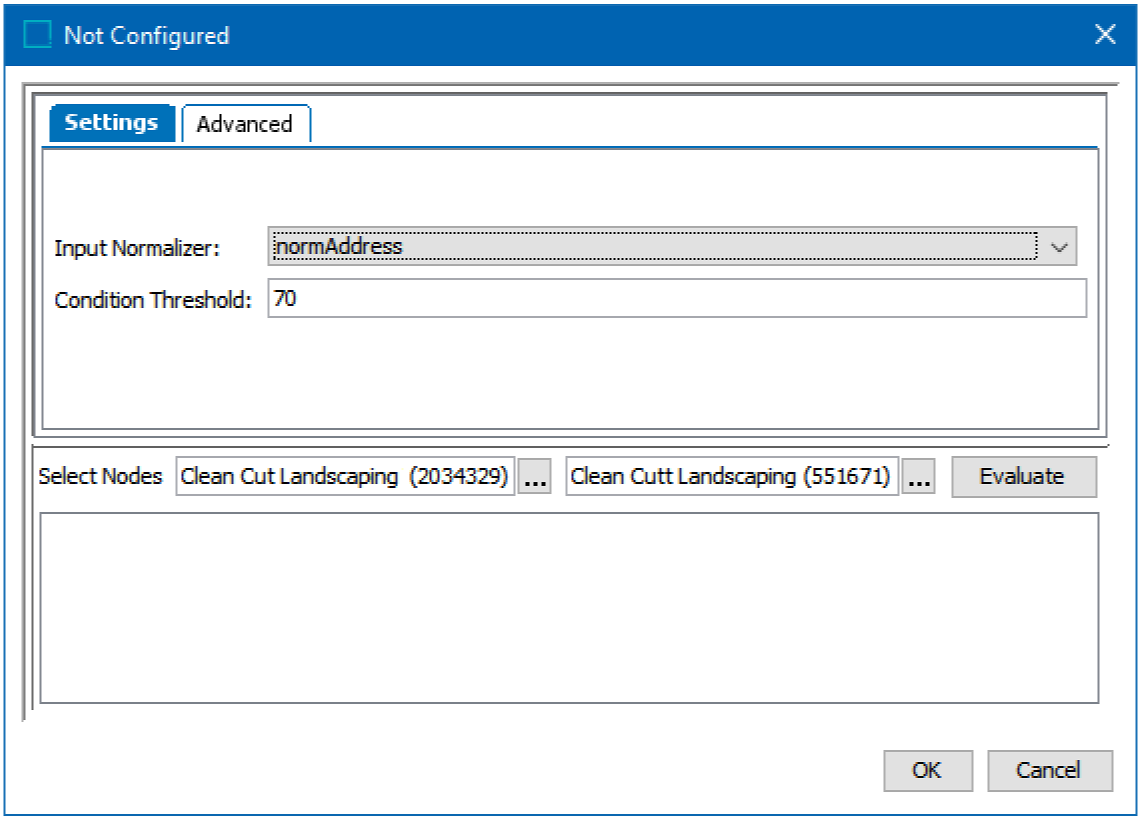
After adding the Address Matcher in the Matchers flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Matcher column and click the ellipsis button (...) to access the configuration dialog.



ID	Matcher	Comment
address	Not Configured ...	

2. On the Not Configured dialog, the **Settings** tab is displayed.



Input Normalizer: normAddress

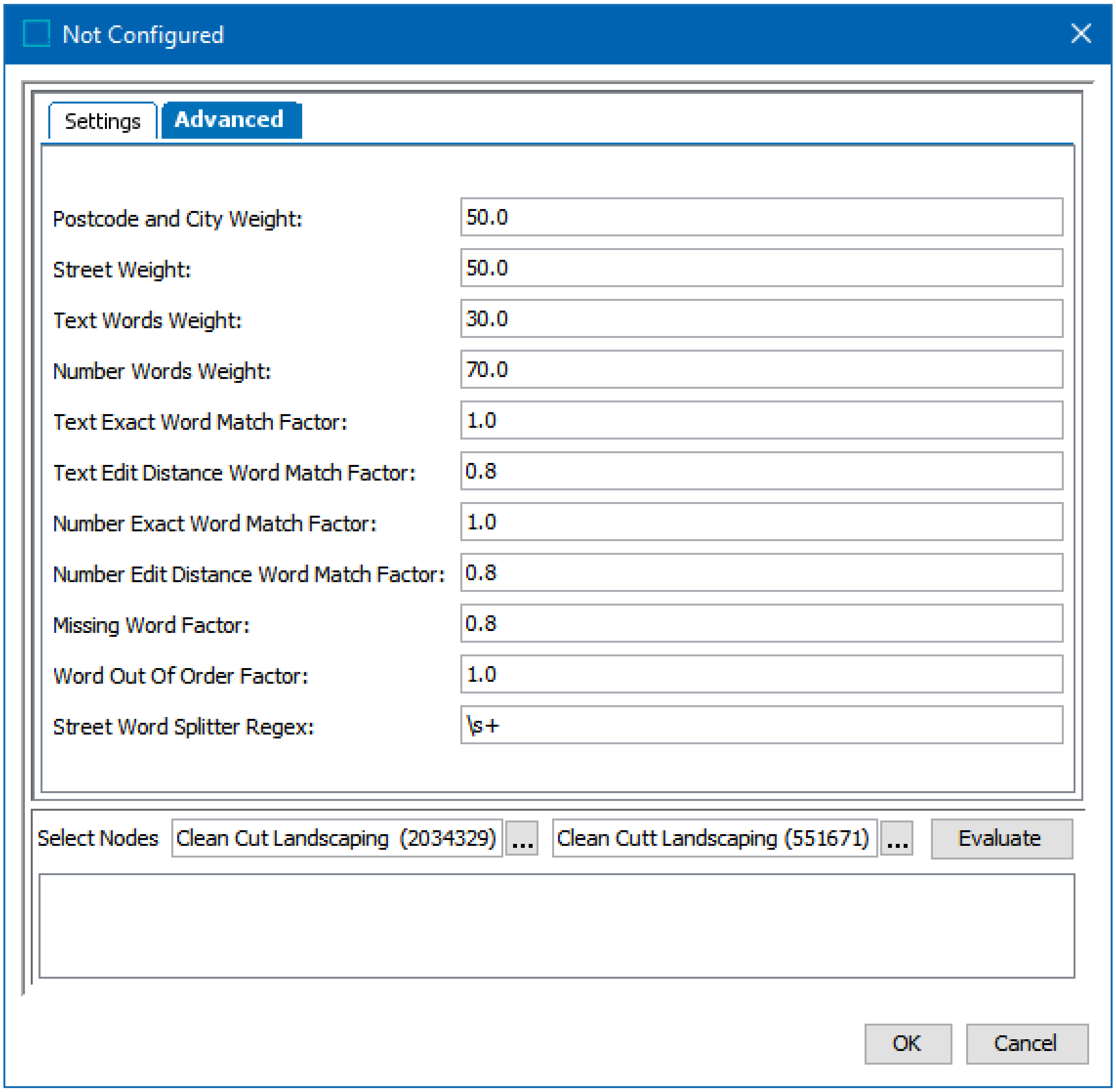
Condition Threshold: 70

Select Nodes Clean Cut Landscaping (2034329) ... Clean Cutt Landscaping (551671) ...

- For the required **Input Normalizer**, use the dropdown to select the associated Address Normalizer or enter a case-sensitive ID for the normalizer.
- For the optional **Condition Threshold**, enter the default minimum score required for the matcher to return 'True' on a rule.

> **Note:** Leave the Condition Threshold parameter empty when this matcher is used in more than one rule and the threshold varies based on the rule. For example, if one rule requires a match score of 70 while another rule requires 75, a default condition threshold can be confusing while troubleshooting. In that case, it is better to add the thresholds in the rules.

3. Click the **Advanced** tab and update the default weights and factors as needed.



Postcode and City Weight:	50.0
Street Weight:	50.0
Text Words Weight:	30.0
Number Words Weight:	70.0
Text Exact Word Match Factor:	1.0
Text Edit Distance Word Match Factor:	0.8
Number Exact Word Match Factor:	1.0
Number Edit Distance Word Match Factor:	0.8
Missing Word Factor:	0.8
Word Out Of Order Factor:	1.0
Street Word Splitter Regex:	\s+

Select Nodes: Clean Cut Landscaping (2034329) ... Clean Cutt Landscaping (551671) ...

- For the required **Postcode and City Weight**, enter the relative weight of the Postcode / City score versus the Street score.

- For the required **Street Weight**, enter the relative weight of the Street score versus the Postcode / City.

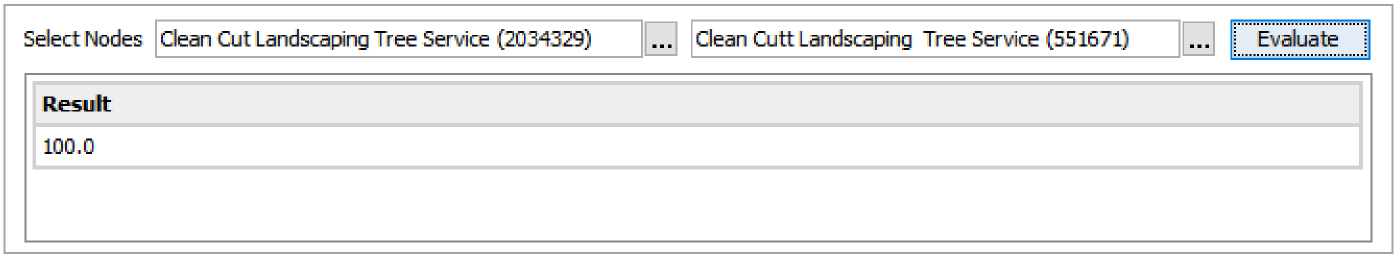
> **Note:** The Street score is a weighted sum of the Number Words score and the Text Words score.

- For the required **Text Words Weight**, enter the relative weight of the Text Words score versus the Number Words score.

- For the required **Number Words Weight**, enter the relative weight of the Number Words score versus the Text Words score. By default, the number score weighs heavier in the final score than text words.

- For the required **Text Exact Word Match Factor**, enter how greatly exact matches influence the final score.
- For the required **Text Edit Distance Word Match Factor**, enter how greatly words that are paired via edit distance influence the final score.
- For the required **Number Exact Word Match Factor**, enter how greatly pairs that are exact matches influence the final score.
- For the required **Number Edit Distance Word Match Factor**, enter how greatly words that are paired via edit distance influence the final score.
- For the required **Missing Word Factor**, enter how much unpaired or missing words penalize the final result.
- For the required **Word Out of Order Factor**, enter how much words that appear out of order penalize the final result.
- For the optional **Street Word Splitter Regex**, leave the default to split on white spaces or enter a different RegEx to split the Street value into words.

4. To test the configuration, for the Select Nodes parameters:



Select Nodes: Clean Cut Landscaping Tree Service (2034329) ... Clean Cutt Landscaping Tree Service (551671) ... Evaluate

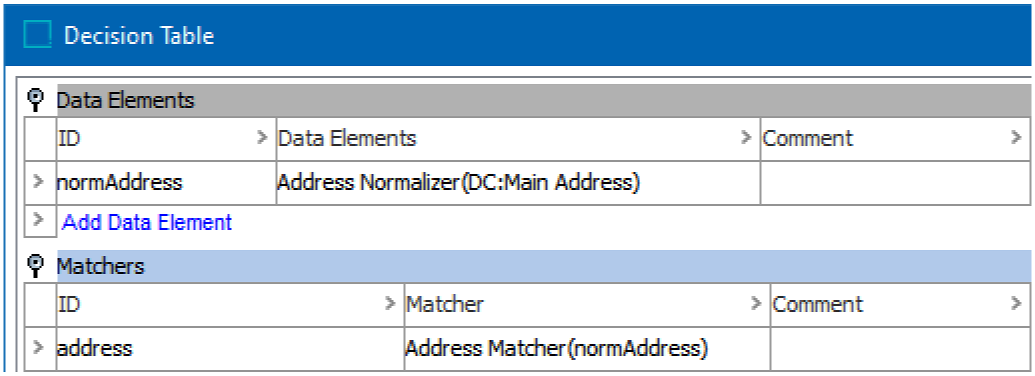
Result
100.0

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  0.0 is displayed when a value is not available in one of the selected nodes or when the addresses do not match. Adjust as indicated by the Evaluator results and repeat the evaluation.

  When red text is displayed, hover to review information about the record. For example, a record that has been deactivated, and so it produces no match code and thus no match score.

5. Click **OK** to save and display the configuration in the Matchers flipper.



Decision Table

Data Elements

ID	Data Elements	Comment
normAddress	Address Normalizer(DC:Main Address)	

Add Data Element

Matchers

ID	Matcher	Comment
address	Address Matcher(normAddress)	

Token and Scoring Examples

The following shows the process of compiling a score when comparing two entities with similar addresses in Germany.

The default weights and factors are used in this example.

1. The Street Word Splitter Regex creates number-street-tokens and text-street-tokens.

Address attribute value	Street-tokens
22 Damm Spandauer, 14059 Berlin, Germany	• 22 • Damm • Spandauer • 14059 • Berlin • Germany
Spanduer Damm 22, Berlin, Germany	• 22 • Damm • Spanduer • Berlin • Germany

Text-street-token pairing	Object 1 Value	Object 2 Value	Result
Text Exact Word Match Factor	Damm	Damm	Text-street-token is an exact match
Text Edit Distance Word Match Factor	Spandauer	Spanduer	Text-street-token has a text-edit-distance of 1

2. Text score: Determine exact matches and edit distances for text-street-tokens between two objects.

   The edit distance is only applied when the text-street-tokens are at least 3 characters long.

Text Score Elements	Setting	Object 1 Value	Object 2 Value	Result
Text Exact Word Match Factor	1.0	Damm	Damm	exact match
Text Edit Distance Word Match Factor	0.8	Spandauer	Spanduer	text-edit-distance of 1
Text score before penalties	1.0 * 0.8=			0.8

3. Text score: Consider all text-street-tokens for sequence and missing words.

Text Score Elements	Setting	Object 1 Values	Object 2 Values	Result
Word Out of Order Factor	0.8	Damm Spandauer	Spanduer Damm	Order is not the same
Missing words	0.8			All tokens are matched
Total text score				0.64

Note: If there were missing tokens ('Spandauer 10-22' compared to 'Spandauer Damm 10-22') the score would be further penalized by multiplying with the Missing Word Factor, which defaults to 0.8.

4. Number score:

Number Score Calculation	Setting	Object 1 Value	Object 2 Value	Result
Number Exact Word Match Factor	1.0	22	22	Numbers are exact matches
Total number score				1.0

5. Street score:

   (TextWordsWeight*textScore + NumberWordsWeight *numberScore) /

   (TextTokensWeight + NumberTokensWeight)

Street Score	Elements	Calculation	Result
(TextTokensWeight*textScore = 5	Text Words Weight = 30.0 Text Score = 0.64	30.0 * 0.64 =	19.2
			+
NumberTokensWeight *numberScore)	Number Words Weight = 70.0 Number Score = 1.00	70.0 * 1.00 =	70
			/
(TextTokensWeight + NumberTokensWeight)	Text Words Weight = 50.0 Number Words Weight = 50.0	50.0 + 50.0	100
Total street score			0.892

6. City / Postcode score:

   If at least one address lacks a postal code, then the cities are compared. An exact match results in City / Postcode score 1.00.

City / Postcode Score Calculation	Score	Object 1 Value	Object 2 Value	Result
Post Code	1.00	14059 Berlin	Berlin	Only one postcode
Total city / postcode score				1.00

7. Final Single Address score:

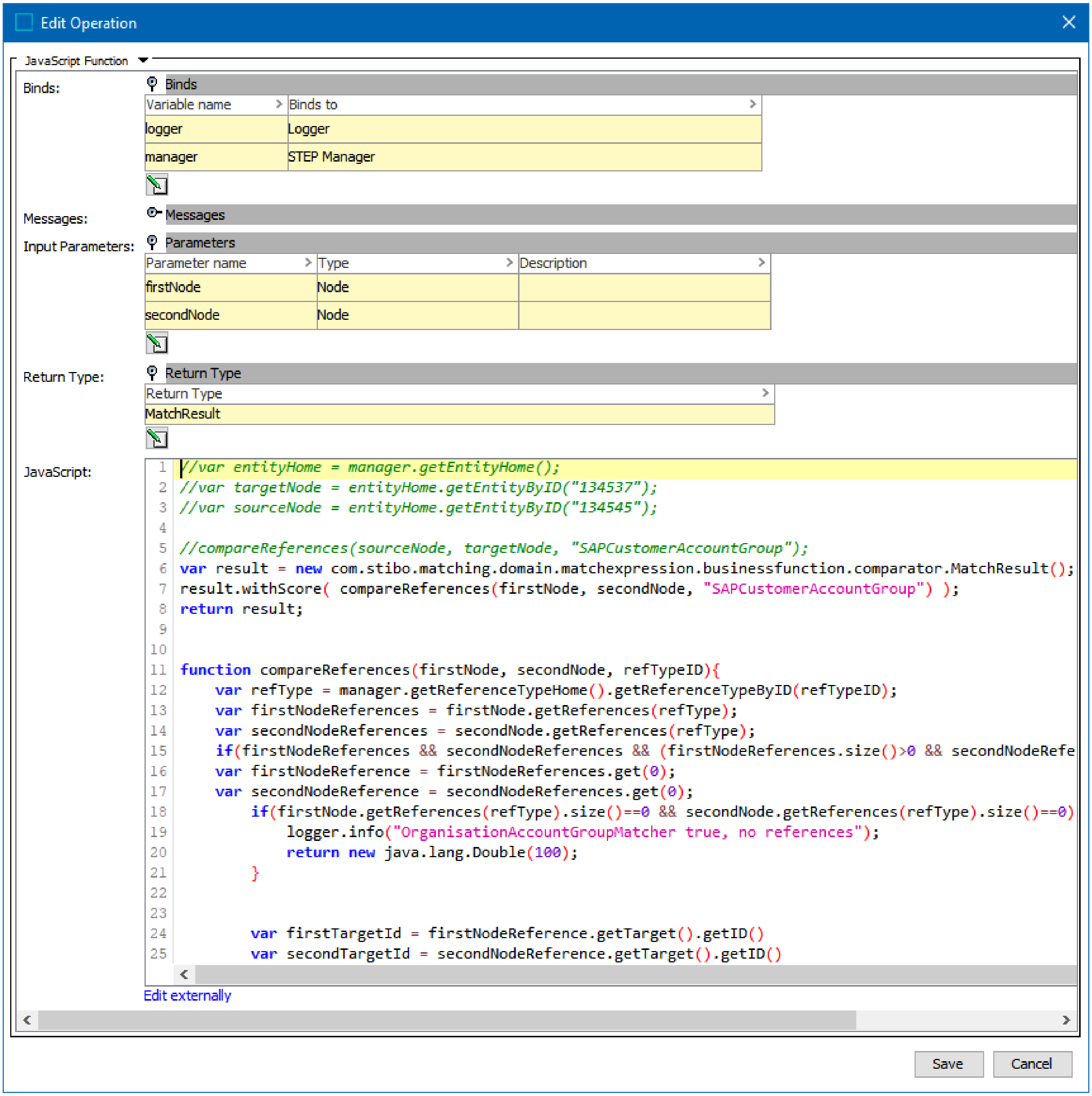
   (Postcode and City Weight* City/Postcode score + Street Weight * Street score) /

   (Postcode and City Weight + Street score)

Final Single Address score	Elements	Calculation	Result
(Postcode and City Weight* City/Postcode score	Postcode and City Weight = 50.0 Text Score = 1.00	50.0 * 1.00 =	50
			+
Street Weight * Street score)	Street Weight = 50.0 Street Score = 0.892	50.0 * 0.892 =	44.6
			/
(Postcode and City Weight + Street score)	Postcode and City Weight = 50.0 Street Weight = 50.0	50.0 + 50.0	(100)
Final Single Address score			94.6

Matcher: Business Function

The Business Function Matcher uses a business function to return a match score and is typically written in JavaScript.



Binds:

Variable name	Binds to
logger	Logger
manager	STEP Manager

Messages:

Input Parameters:

Parameter name	Type	Description
firstNode	Node	
secondNode	Node	

Return Type:

Return Type
MatchResult

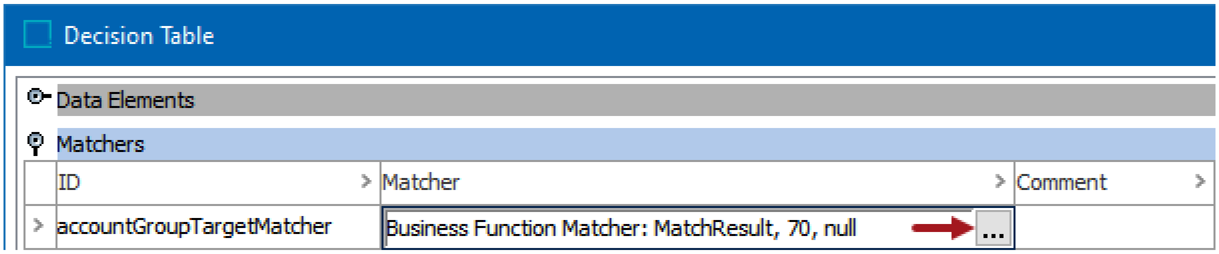
JavaScript:

```
//var entityHome = manager.getEntityHome();
//var targetNode = entityHome.getEntityByID("134537");
//var sourceNode = entityHome.getEntityByID("134545");

//compareReferences(sourceNode, targetNode, "SAPCustomerAccountGroup");
var result = new com.stibo.matching.domain.matchexpression.businessfunction.comparator.MatchResult();
result.withScore( compareReferences(firstNode, secondNode, "SAPCustomerAccountGroup") );
return result;


function compareReferences(firstNode, secondNode, refTypeID){
    var refType = manager.getReferenceTypeHome().getReferenceTypeByID(refTypeID);
    var firstNodeReferences = firstNode.getReferences(refType);
    var secondNodeReferences = secondNode.getReferences(refType);
    if(firstNodeReferences && secondNodeReferences && (firstNodeReferences.size()>0 && secondNodeRefe
    var firstNodeReference = firstNodeReferences.get(0);
    var secondNodeReference = secondNodeReferences.get(0);
        if(firstNode.getReferences(refType).size()==0 && secondNode.getReferences(refType).size()==0)
            logger.info("OrganisationAccountGroupMatcher true, no references");
            return new java.lang.Double(100);
        }


        var firstTargetId = firstNodeReference.getTarget().getID()
        var secondTargetId = secondNodeReference.getTarget().getID()
```

Configuring a Business Function Matcher

After adding the Business Function Matcher in the Matchers flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Matcher column and click the ellipsis button (...) to access the configuration dialog.



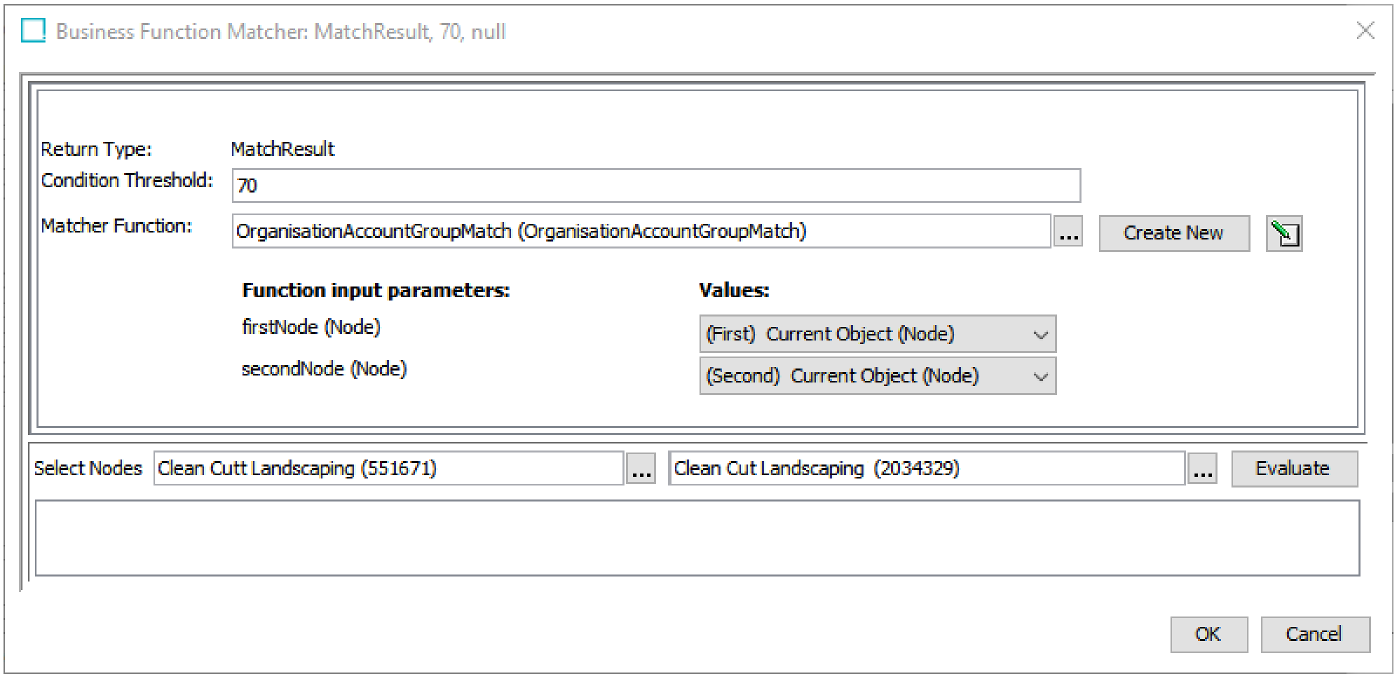
Decision Table

Data Elements

Matchers

ID	Matcher	Comment
accountGroupTargetMatcher	Business Function Matcher: MatchResult, 70, null	

2. On the Business Function Matcher dialog:



Business Function Matcher: MatchResult, 70, null

Return Type: MatchResult
Condition Threshold: 70
Matcher Function: OrganisationAccountGroupMatch (OrganisationAccountGroupMatch)

Function input parameters:	**Values:**
firstNode (Node)	(First) Current Object (Node)
secondNode (Node)	(Second) Current Object (Node)

Select Nodes: Clean Cutt Landscaping (551671) | Clean Cut Landscaping (2034329)

- The Business Function will define a number of Input Parameters. In the screenshot above, the business function declares firstNode and secondNode, both of type Node. Those input parameters are mapped to the two nodes that are matched by the matcher. In this scenario, no Data Element is required since all input is from the Business Function. If the Business Function had a third input parameter of type String, any Data Element with an output of type String could be mapped to that input.
- For **Return Type**, MatchResult is displayed. A Business Function matcher must deliver a MatchResult output.

- For the optional **Condition Threshold**, enter the minimum score required for the matcher to return 'True' on a rule.

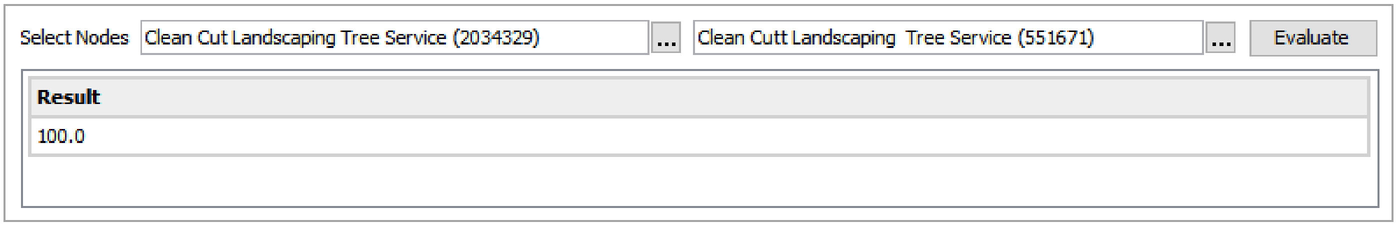
> **Note:** Leave the Condition Threshold parameter empty when this matcher is used in more than one rule and the threshold varies based on the rule. For example, if one rule requires a match score of 70 while another rule requires 75, a default condition threshold can be confusing while troubleshooting. In that case, it is better to add the thresholds in the rules.

3. For the required **Matcher Function:**

- Click the ellipsis button (**...**) to display the Select Business Function dialog. If necessary, click the ellipsis button (**...**) to modify the selected global business function.

- Click the **Create New** button to create a new business function.

4. To test the configuration, for the Select Nodes parameters:



Select Nodes: Clean Cut Landscaping Tree Service (2034329) ... Clean Cutt Landscaping Tree Service (551671) ...

Result
100.0

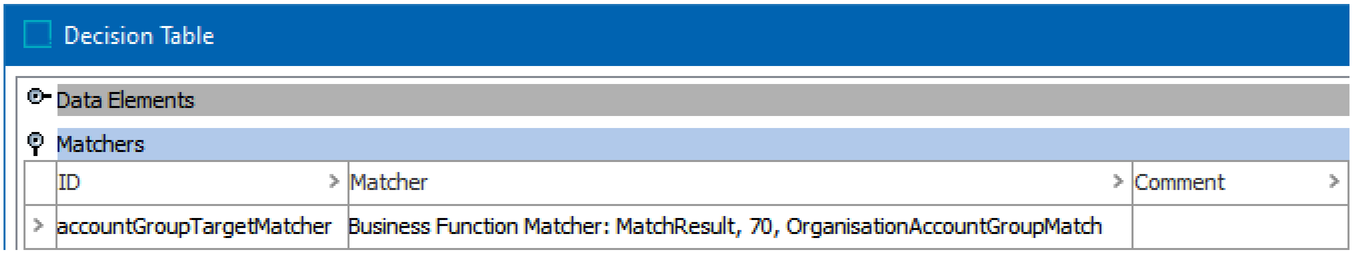
- Click the ellipsis button (**...**) for each field and select two objects for comparison.

- Click the **Evaluate** button.

  0.0 is displayed when a value is not available in one of the selected nodes or when the values do not match. Adjust as indicated by the Evaluator results and repeat the evaluation.

  When red text is displayed, hover to review information about the record. For example, a record that has been deactivated, and so it produces no match code and thus no match score.

5. Click **OK** to save and display the configuration in the Matchers flipper.



ID	Matcher	Comment
accountGroupTargetMatcher	Business Function Matcher: MatchResult, 70, OrganisationAccountGroupMatch	

Matcher: Email

The Email Normalizer data element (as defined in the **Data Element: Email Normalizer** topic) normalizes email data for two objects. The Email Matcher compares the normalizer output and generates a match score (also called the 'rank score' in Web UI).

When a match score is applied to the defined rules (refer to the **Match Criteria Rules** topic), a final match score is determined to rank the likelihood of a match between the two objects.

Input

The Email Matcher takes input from the selected email data element and retrieves all emails for the two objects under comparison.

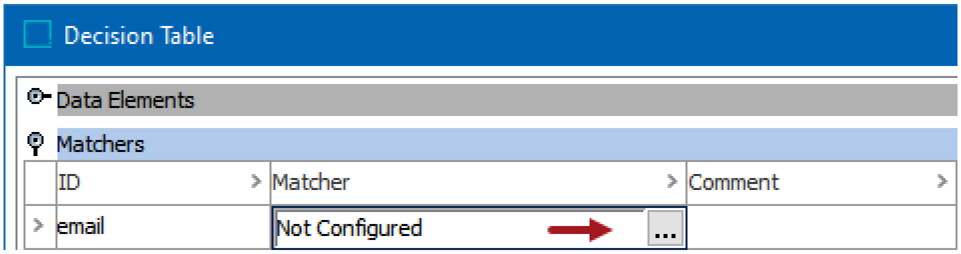
Functionality

When the email matcher compares email addresses between two objects and finds an exact match, the score is 100. If there is not an exact match, the score is 0.

Configuring an Email Matcher

After adding the Email Matcher in the Matchers flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Matcher column and click the ellipsis button (...) to access the configuration dialog.



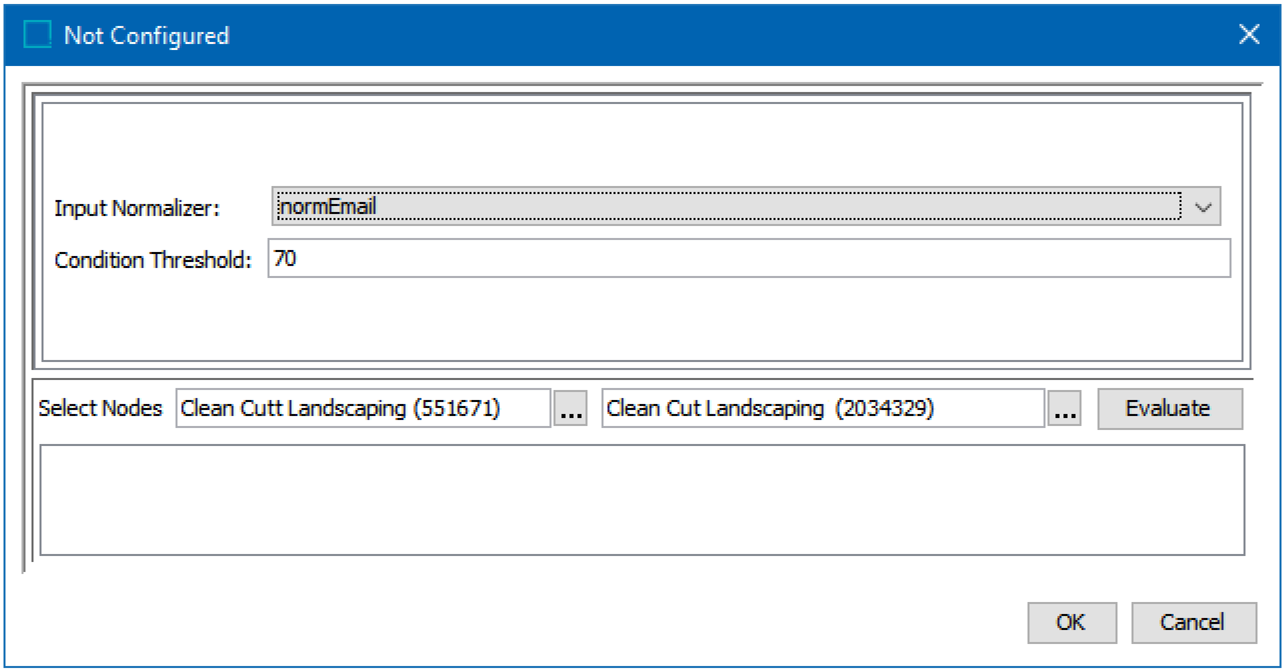
Decision Table

Data Elements

Matchers

ID	Matcher	Comment
email	Not Configured ...	

2. On the Not Configured dialog:



Input Normalizer: normEmail

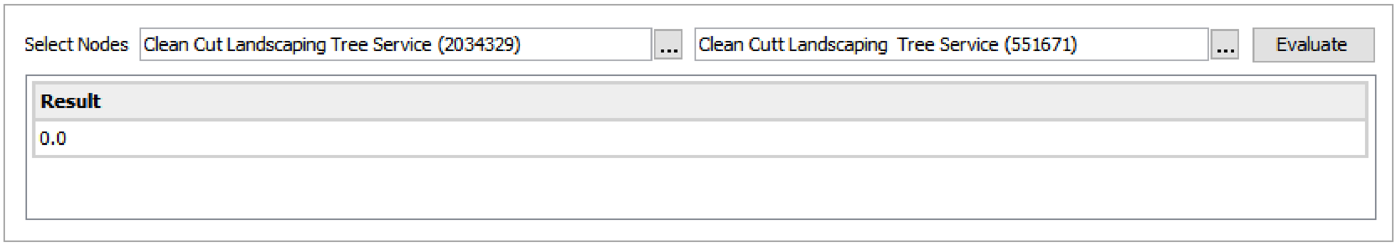
Condition Threshold: 70

Select Nodes Clean Cutt Landscaping (551671) ... Clean Cut Landscaping (2034329) ... Evaluate

- For the required **Input Normalizer**, use the dropdown to select the associated Email Normalizer or enter a case-sensitive ID for the normalizer.
- For the optional **Condition Threshold**, enter the minimum score required for the matcher to return 'True' on a rule.

> **Note:** Leave the Condition Threshold parameter empty when this matcher is used in more than one rule and the threshold varies based on the rule. For example, if one rule requires a match score of 70 while another rule requires 75, a default condition threshold can be confusing while troubleshooting. In that case, it is better to add the thresholds in the rules.

3. To test the configuration, for the Select Nodes parameters:


Select Nodes Clean Cut Landscaping Tree Service (2034329) ... Clean Cutt Landscaping Tree Service (551671) ... Evaluate

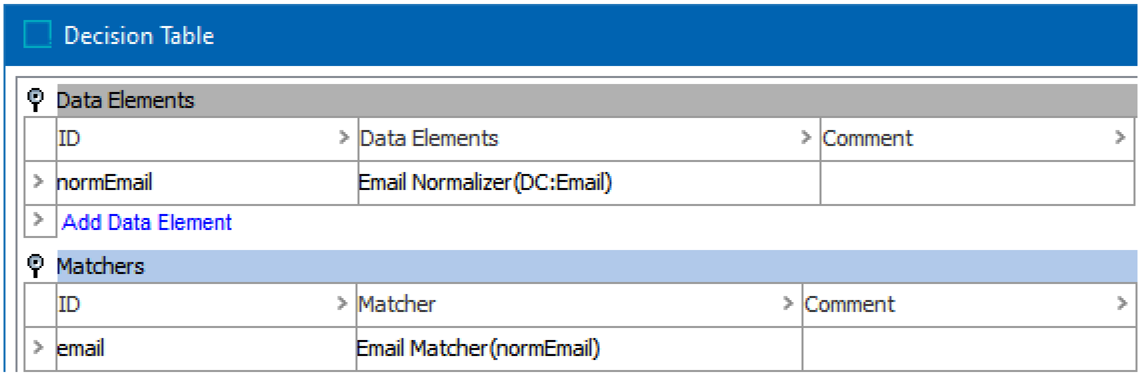
Result
0.0

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button to show the score.

0.0 is displayed when a value is not available in one of the selected nodes or when the emails do not match. Adjust as indicated by the Evaluator results and repeat the evaluation.

When red text is displayed, hover to review information about the record. For example, a record that has been deactivated, and so it produces no match code and thus no match score.

4. Click **OK** to save and display the configuration in the Matchers flipper.



Decision Table

Data Elements

ID	Data Elements	Comment
normEmail	Email Normalizer(DC:Email)	
Add Data Element		

Matchers

ID	Matcher	Comment
email	Email Matcher(normEmail)	

Matcher: Function

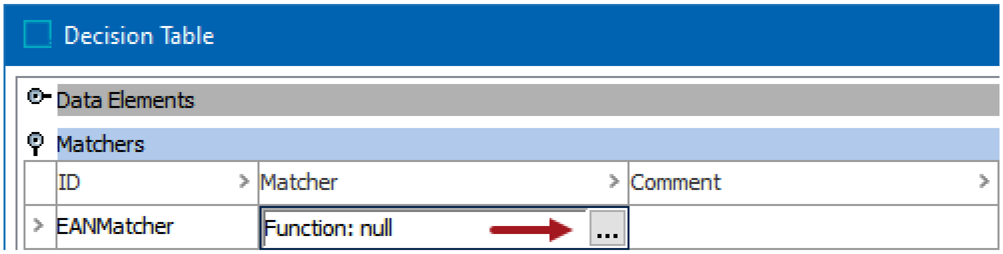
The Function Matcher uses STEP Functions to produce the match score and the function 'mcevaluate' (Match Context Evaluate) to assess elements from the Data Elements section and the Matchers section of the decision table and compare their results. For more information, refer to the **Function Editor** topic in the **Resource Materials** section of online help and the **Matching Algorithms and Match Expressions** topic.

The Function Matcher shown in the section below uses a Levenshtein distance, which is a metric for how many edits (substitution, insertion, deletion) it takes to make one string look like another. For example, the Levenshtein distance between the strings 'AXR55487' and '8XRT5487' is 2 because the first and fourth digits are different. In STEP terms, the strings would be 75 percent alike (6/8*100). The example below matches for European Article Number (EAN) and performs a comparison of the EANs in two objects. The matcher does not handle special cases, such as where the normalizer returns strings that are obviously not EANs, like empty strings, because resolving such cases is expected to be handled by the normalizer.

Configuring a Function Matcher

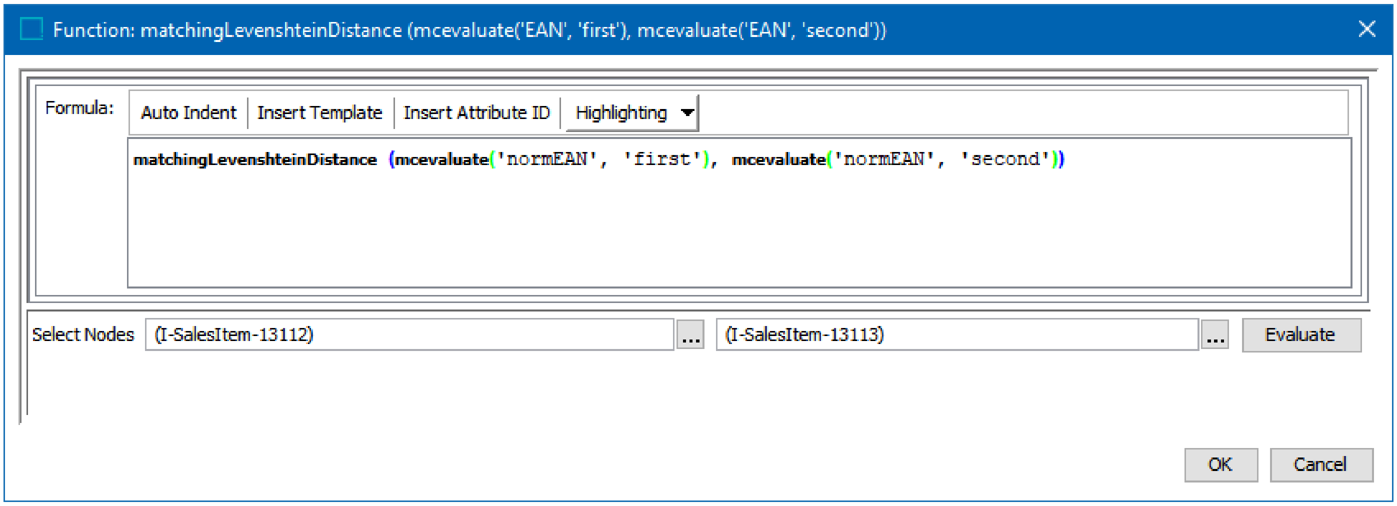
After adding the Function Matcher in the Matchers flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Matcher column and click the ellipsis button (...) to access the configuration dialog.



ID	Matcher	Comment
EANMatcher	Function: null	

2. On the Function dialog, for **Formula** add the necessary STEP function. Use the ID of the corresponding Data Element (‘normEAN’ in the image below) to identify the data being matched by the function.



Function: matchingLevenshteinDistance (mcevaluate('EAN', 'first'), mcevaluate('EAN', 'second'))

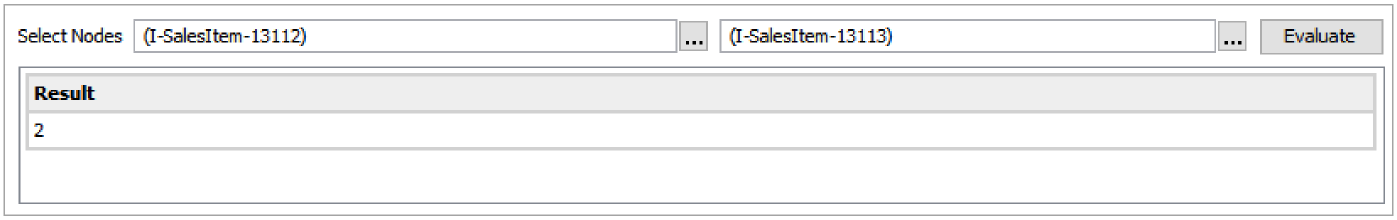
Formula:

```
matchingLevenshteinDistance (mcevaluate('normEAN', 'first'), mcevaluate('normEAN', 'second'))
```

Select Nodes (I-SalesItem-13112) (I-SalesItem-13113)

Note: The individual scores returned by the Machine Learning Matcher can be accessed using the Match Context Evaluate function and dot notation (.), for example, mcevaluate ('ml_matcher.name') to achieve the person name score. For detailed information, refer to the Matcher: Machine Learning Matcher topic in the Matching, Linking, and Merging documentation.

3. To test the configuration, for the Select Nodes parameters:



Select Nodes (I-SalesItem-13112) (I-SalesItem-13113)

Result
2

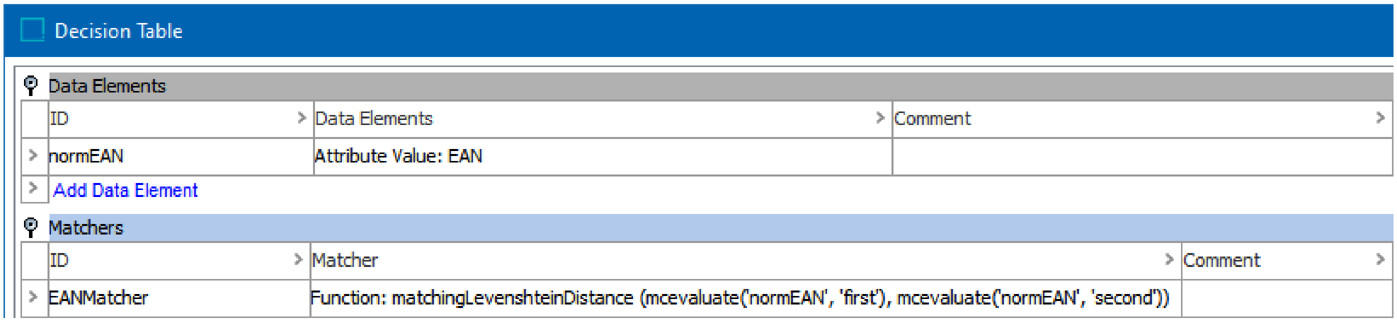
- Click the ellipsis button (...)) for each field and select two objects for comparison.

- Click the **Evaluate** button.

  0.0 is displayed when a value is not available in one of the selected nodes or when the values do not match. Adjust as indicated by the Evaluator results and repeat the evaluation.

  When red text is displayed, hover to review information about the record. For example, a record that has been deactivated, and so it produces no match code and thus no match score.

4. Click **OK** to save and display the configuration in the Matchers flipper.



Decision Table

Data Elements

ID	Data Elements	Comment
normEAN	Attribute Value: EAN	

Matchers

ID	Matcher	Comment
EANMatcher	Function: matchingLevenshteinDistance (mcevaluate('normEAN', 'first'), mcevaluate('normEAN', 'second'))	

Matcher: JavaScript Function

A JavaScript Function Matcher uses the Match Expression Context 'evaluate' function to assess elements from the Data Elements section and Matchers section of the decision table and compare their results.

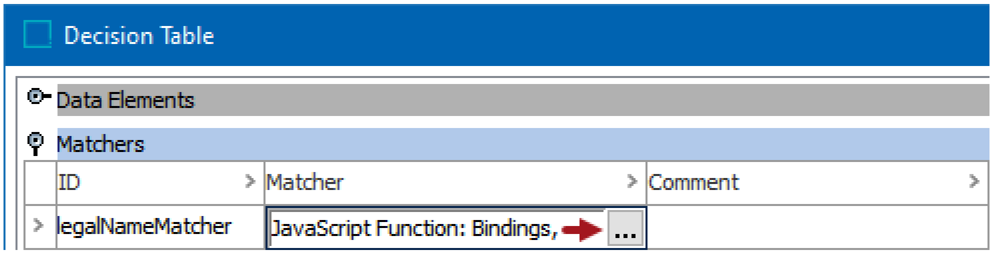
For example, in the JavaScript Matcher shown below, a basic email matcher performs a plain comparison of the emails by comparing normalized email addresses as text strings. The matcher does not handle special cases, such as where the normalizer returns strings that are obviously not emails, like empty strings, because resolving such cases is expected to be handled by the normalizer.

Refer to the **Extending Party Data Matchers With JavaScript** section below for information on expanding the JavaScript functionality.

Configuring a JavaScript Function Matcher

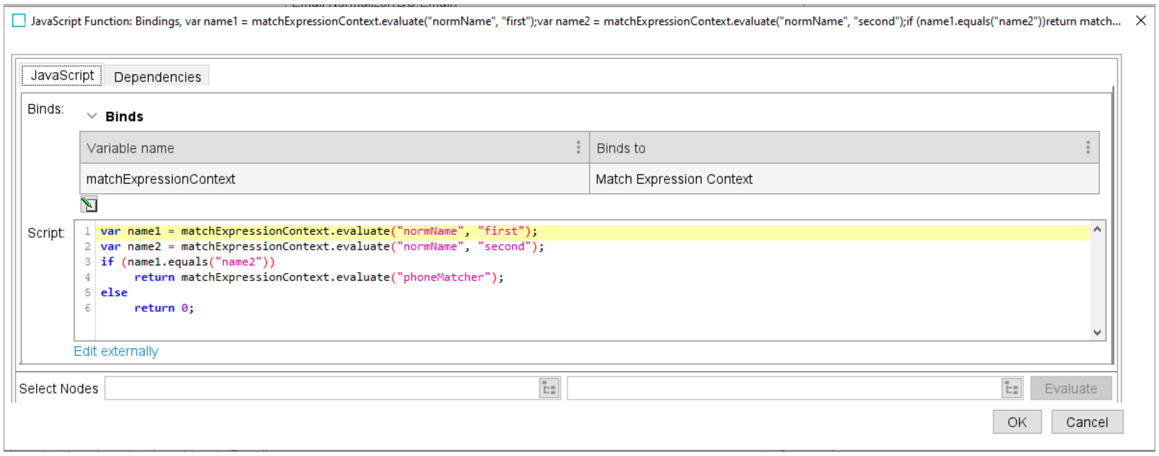
After adding the JavaScript Function Matcher in the Matchers flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Matcher column and click the ellipsis button (...) to access the configuration dialog.



ID	Matcher	Comment
legalNameMatcher	JavaScript Function: Bindings, ...	

2. On the JavaScript Function dialog, the **JavaScript** tab is displayed.


JavaScript Function: Bindings, var name1 = matchExpressionContext.evaluate("normName", "first");var name2 = matchExpressionContext.evaluate("normName", "second");if (name1.equals("name2"))return match...

Variable name	Binds to
matchExpressionContext	Match Expression Context

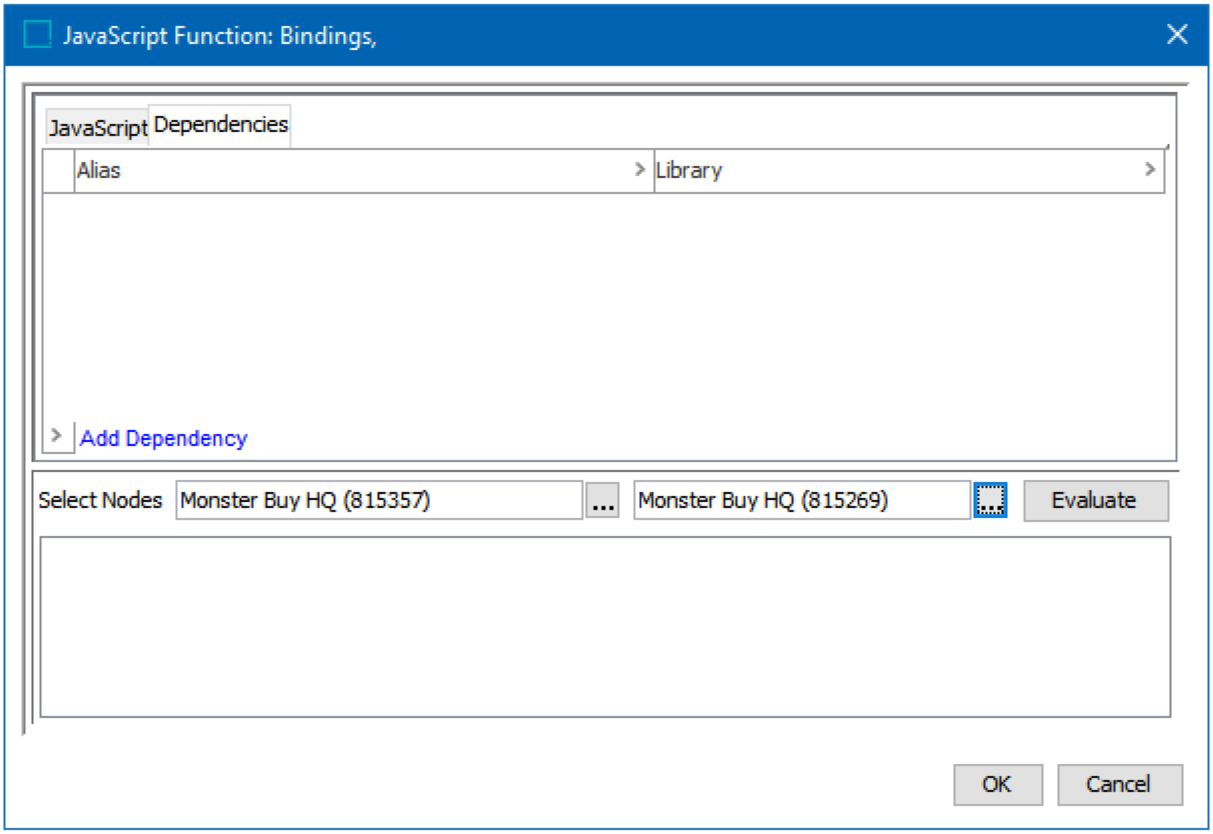
Script:

```
var name1 = matchExpressionContext.evaluate("normName", "first");
var name2 = matchExpressionContext.evaluate("normName", "second");
if (name1.equals("name2"))
     return matchExpressionContext.evaluate("phoneMatcher");
else
     return 0;
```

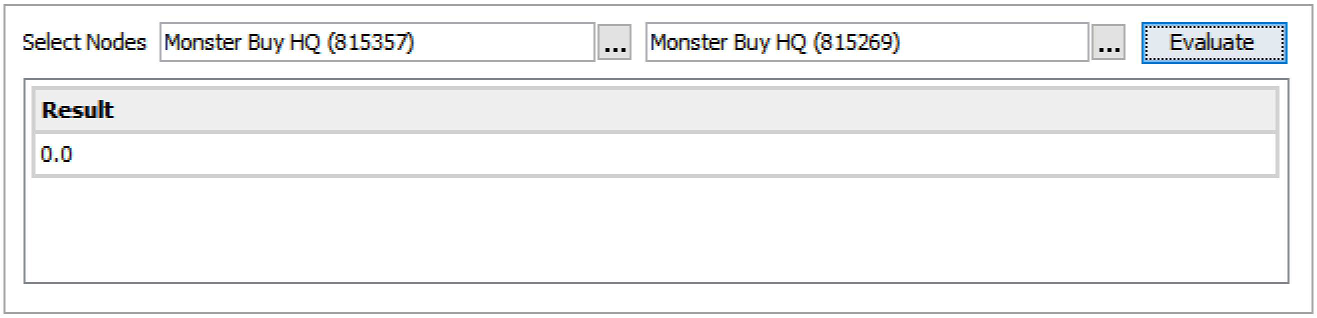
- For the required **Binds**, at a minimum, add a bind for the Match Expression Context.
- For the required **Script**, add the matcher JavaScript.

> **Note:** For the Machine Learning Matcher, the individual scores can be evaluated using dot (.) notation, such as ('ml_matcher.name'). For detailed information, refer to the Machine Learning Matcher topic in the Matching, Linking, and Merging documentation.

3. Click the **Dependencies** tab, click the **Add Dependency** link to select any libraries required for the script added on the JavaScript tab.



4. To test the configuration, for the Select Nodes parameters:



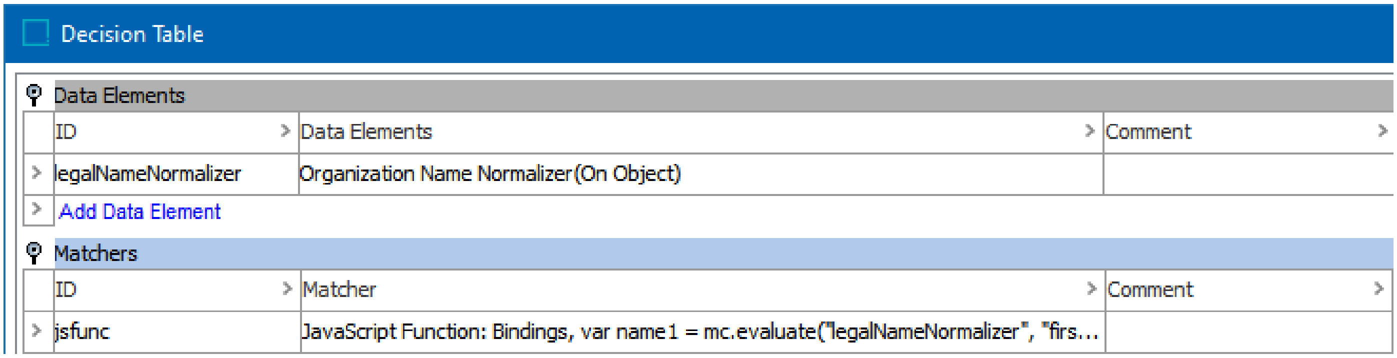
Result
0.0

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  0.0 is displayed when a value is not available in one of the selected nodes or when the values do not match. Adjust as indicated by the Evaluator results and repeat the evaluation.

  When red text is displayed, hover to review information about the record. For example, a record that has been deactivated, and so it produces no match code and thus no match score.

5. Click **OK** to save and display the configuration in the Matchers flipper.



Decision Table

Data Elements

ID	Data Elements	Comment
legalNameNormalizer	Organization Name Normalizer(On Object)	
Add Data Element		

Matchers

ID	Matcher	Comment
jsfunc	JavaScript Function: Bindings, var name1 = mc.evaluate("legalNameNormalizer", "firs...	

Expanding Party Data Matchers With JavaScript

For complicated solutions, you can extend the capabilities of a party data matcher via JavaScript to allow for more flexibility and functionality. Examples include:

- Evaluate the scores of three other matchers and perform a computation to produce a final match score.
- Evaluate the score of one matcher to establish which of two other matchers score to return.

An extended party data JavaScript matcher includes the following elements:

1. Uses the Match Expression Context **evaluate** function to retrieve the output of a desired normalizer, where 'mc' is a bind to the Match Expression Context. Refer to the **Matching Algorithms and Match Expressions** topic.
2. Uses an iterator to access the set of values / strings of both objects being matched.
3. Compares those objects to output a match score (also called the 'rank score' in Web UI).

For more information on party data JavaScript normalizers, refer to the **Data Element: JavaScript Function** topic.

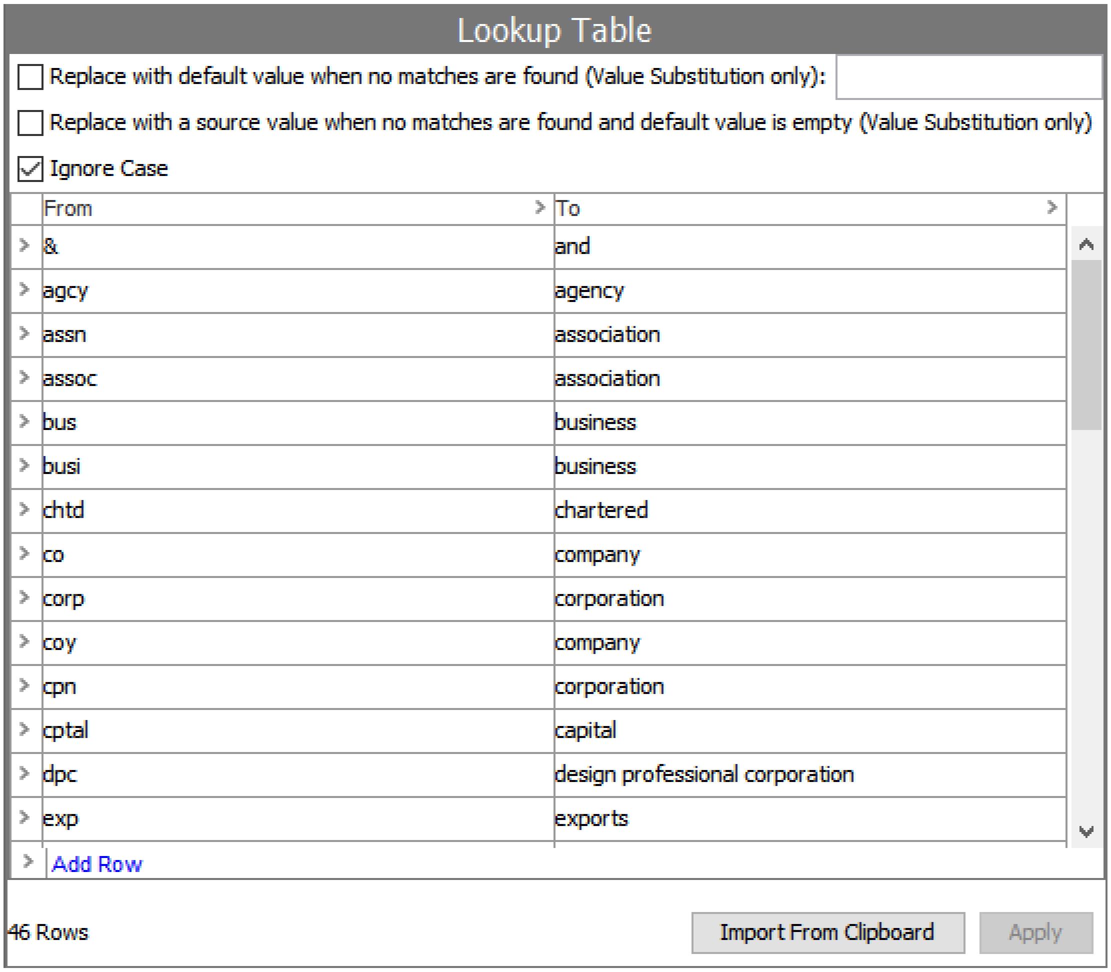
Matcher: Organization Name

The Organization Name Normalizer data element (as defined in the **Data Element: Organization Name Normalizer** topic) normalizes the organization name data for two objects. The Organization Name Matcher compares the normalizer output and generates a match score (also called the 'rank score' in Web UI). The final score calculation is based on a number of match factors available under the Advanced tab in the configuration, which allow you to tune the matcher towards your specific data set and business need.

When a match score is applied to the defined rules (refer to the **Match Criteria Rules** topic), a final match score is determined to rank the likelihood of a match between the two objects.

Considerations

A **Word Alias Table** can be used to perform case-insensitive matching by alias. The Customer & Supplier MDM Configuration Guide in the Solution Enablement documentation refers to a Word Alias Table (illustrated below) that allows the matching to handle common organization word substitutions like 'co' with 'company'. For more information, refer to the **Transformation Lookup Tables** topic in the **Resource Materials** section of the online help.



Lookup Table

□ Replace with default value when no matches are found (Value Substitution only):

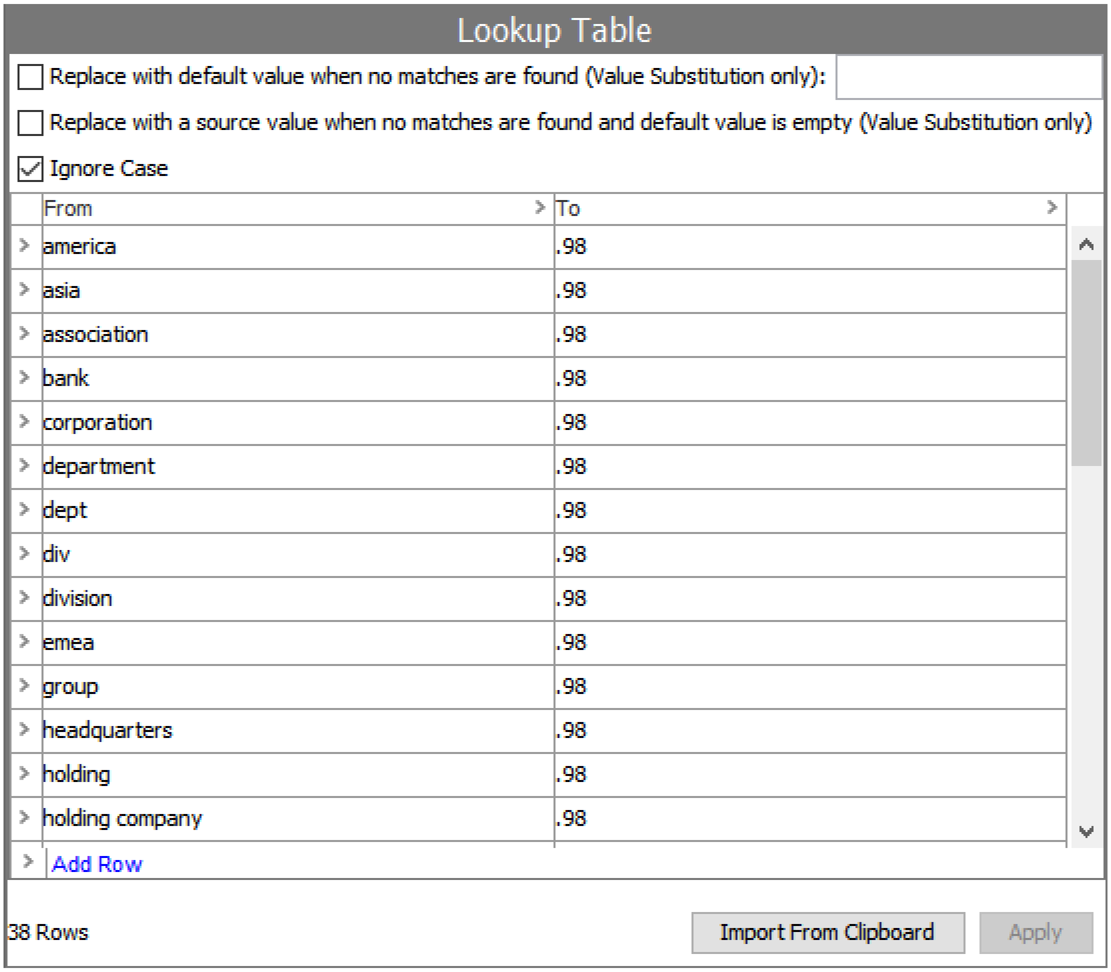
□ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

☑ Ignore Case

From	To
&	and
agcy	agency
assn	association
assoc	association
bus	business
busi	business
chtd	chartered
co	company
corp	corporation
coy	company
cpn	corporation
cptal	capital
dpc	design professional corporation
exp	exports

46 Rows

An **Unmatched Word Factor Table** assigns weights to individual words that may routinely be missing due to people writing company names in a hurry. For example, 'Ajax Company Cleaning Supplies Inc' compared to 'Ajax Cleaning Supplies'. Typically, missing words penalize the score according to the ’Missing Word Factor’ parameter. However, if the missing words ('Company' and 'Inc') are in the Unmatched Factor Word Table, the designated factor for each word will be taken from that table instead. In the unmatched word factor table below, a lot of missing words are set to a penalty of 0.98, impacting the score much less than the 0.7 that is default for the ‘Missing Word Factor’. The Customer & Supplier MDM Configuration Guide in the Solution Enablement documentation refers to an Unmatched Word Factor Table that is illustrated below. This table can also be used to assign certain words an even harsher score impact when they are missing. For more information, refer to the **Transformation Lookup Tables** topic in the **Resource Materials** section of the online help.



☐ Replace with default value when no matches are found (Value Substitution only):

☐ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

☑ Ignore Case

From	To
america	.98
asia	.98
association	.98
bank	.98
corporation	.98
department	.98
dept	.98
div	.98
division	.98
emea	.98
group	.98
headquarters	.98
holding	.98
holding company	.98

38 Rows

Input

The Organization Name Matcher takes input from the selected Input Normalizer. This is usually an Organization Name Data Element. The matcher retrieves all organization names for the two objects under comparison.

Functionality

The Organization Name Matcher considers every organization name of the first object in the match context and compares each of those with every organization name of the second object in the match context. The final score of the Organization Name Matcher is the highest score of any two organization names. Refer to the **Matching Algorithms and Match Expressions** topic.

The comparison of each set of two organization names includes:

1. Using the Name Word Splitter Regex to split the organization name value to create name-tokens
2. Defining pairs of name-tokens between the two objects

   Possible pairings for organization name tokens are:

- Exact match – Scores 100.
- Word Alias Table – If configured, performs case-insensitive matching by alias. Any name-tokens that match based on the Word Alias Table is scored a multiplier equal to the Alias Word Match Factor. For example, 'Ajax Cleaning Supplies Co' compared to 'Ajax Cleaning Supplies Company' scores the first three (3) tokens at 100 as exact matches. The final token 'co matches 'Company' by the alias table and scores a multiplier according to Alias Word Match Factor.
- Concatenation matching – If two name-tokens in one organization name can be concatenated to match one name-token in the other organization name, it receives a score multiplier equal to the Concatenation Word Match Factor. For example, 'Ajax Cleaning Supplies Co' compared to 'Ajax Cleaningsupplies Co' scores the first and last words as 100. The middle name-tokens of the first object can be concatenated to match the name-token of the second object. Concatenated name-tokens must match exactly, so a good normalization is important for this comparator to work.
- Edit distance matching – (adjusting for a few wrong characters due to typographical errors) - If both name-tokens are at least three (3) characters long, and one can be made identical with the other by adding, deleting, or changing a single character, the Edit Distance Word Match Factor is applied.
- Acronym matching – If a name-token in one organization name is an acronym of the list of name-tokens in the other organization name, the Acronym Word Match Factor is applied. For example, 'Ajax Cleaning Supplies' compared to 'ACS' is a match, 'Ajax C S' compared to 'Ajax CS' is a match. Ordering of the acronym letters is important, so 'Ajax Cleaning Supplies' compared to 'ASC' is not a match.

3. Determine score penalties

   - Sequence matching – If tokens are out of order, a further penalty multiplier is determined by the Word Out Of Order Factor. For example, 'Ajax Cleaning Supplies' compared to 'Cleaning Supplies Ajax'.
   - Unmatched / Missing matching – If there are missing tokens, the score is penalized by multiplying with the Missing Word Factor. If more than half the name-tokens in any organization name are unpaired they are considered not matching. Specific words can be assigned a higher or lower missing word penalty score by using the Unmatched Word Factor Table, described in the **Considerations** section above.

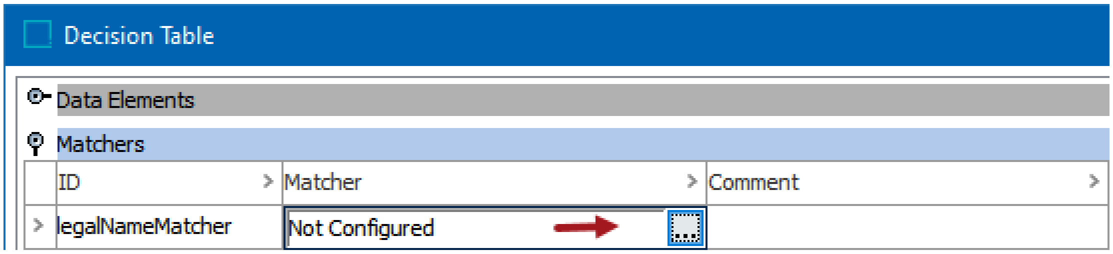
4. Determine the final score by comparing an organization name from the first object to an organization name from the second object

   The final score of an Organization Name Matcher is the best score of matching any organization name on the first object to any organization name on the second object.

Configuring an Organization Name Matcher

After adding the Organization Name Matcher in the Matchers flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Matcher column and click the ellipsis button (...) to access the configuration dialog.



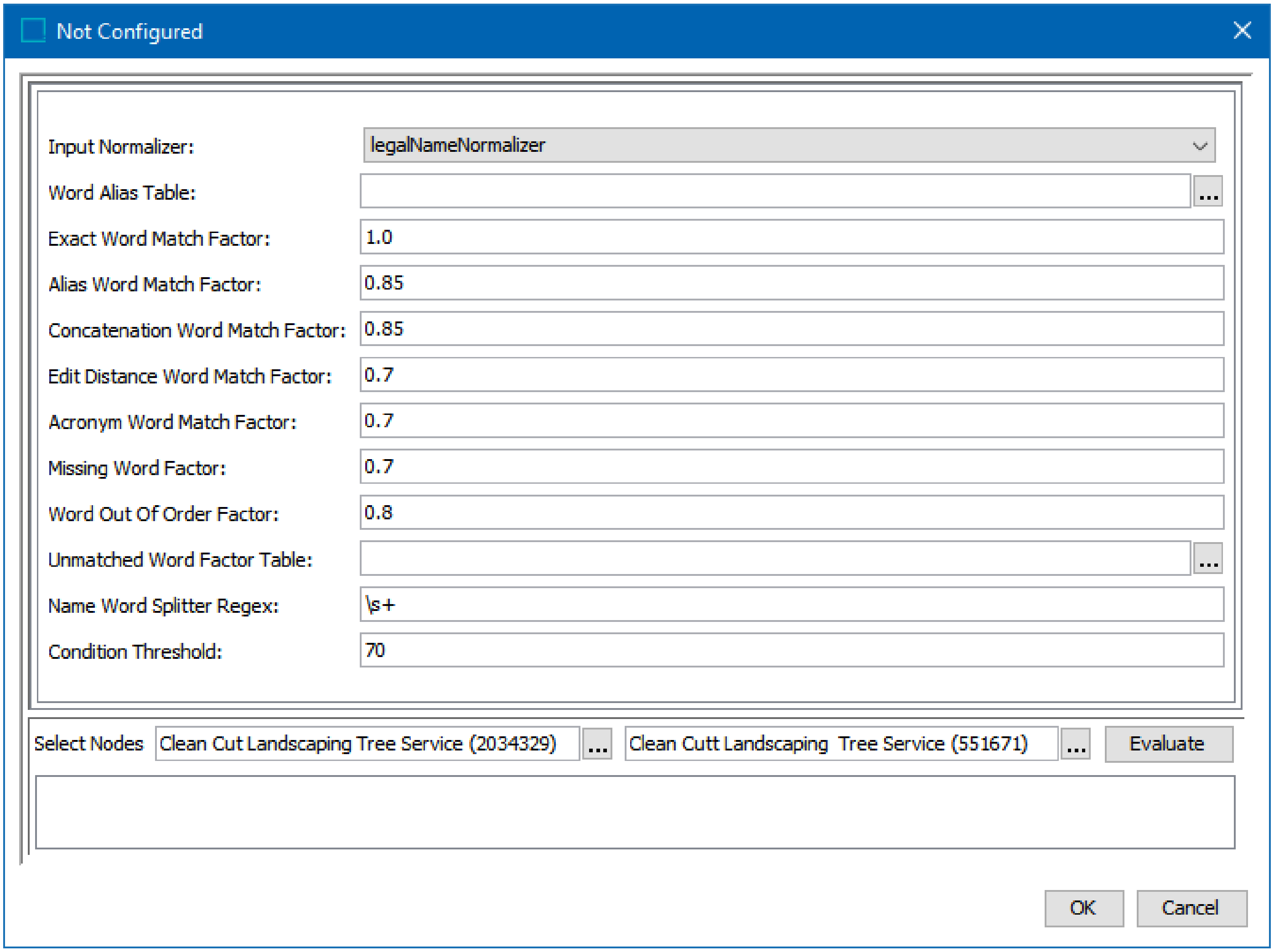
Decision Table

Data Elements

Matchers

ID	Matcher	Comment
legalNameMatcher	Not Configured	

2. On the Not Configured dialog:



Not Configured

Input Normalizer:	legalNameNormalizer
Word Alias Table:	
Exact Word Match Factor:	1.0
Alias Word Match Factor:	0.85
Concatenation Word Match Factor:	0.85
Edit Distance Word Match Factor:	0.7
Acronym Word Match Factor:	0.7
Missing Word Factor:	0.7
Word Out Of Order Factor:	0.8
Unmatched Word Factor Table:	
Name Word Splitter Regex:	\s+
Condition Threshold:	70

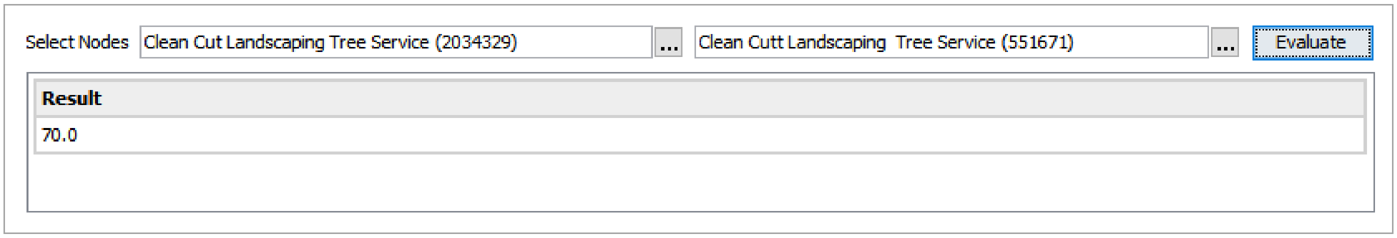
Select Nodes: Clean Cut Landscaping Tree Service (2034329) | Clean Cutt Landscaping Tree Service (551671)

- For the required **Input Normalizer**, use the dropdown to select the associated Organization Name Normalizer or enter a case-sensitive ID for the normalizer.

- For the optional **Word Alias Table**, click the ellipsis button (**...**) and select a Transformation Lookup Table to substitute words with the same or similar meaning. Refer to the **Considerations** section above.
- For the required **Exact Word Match Factor**, enter how greatly exact matches influence the final score.
- For the required **Alias Word Match Factor**, enter how greatly words that are paired via aliases influence the final score.
- For the required **Concatenation Word Match Factor**, enter how greatly pairs where one is concatenated and the other is not concatenated influence the final score.
- For the required **Edit Distance Word Match Factor**, enter how greatly pairs via edit distance influence the final score.
- For the required **Acronym Word Match Factor**, enter how greatly pairs where one is an acronym and the other is not an acronym influence the final score.
- For the required **Missing Word Factor**, enter how much unpaired or missing words penalize the final result.
- For the required **Word Out of Order Factor**, enter how much words that appear out of order penalize the final result.
- For the optional **Unmatched Word Factor Table**, click the ellipsis button (**...**) and select a Transformation Lookup Table to assign factors to certain words. Refer to the **Considerations** section above.
- For the optional **Name Word Splitter Regex**, leave the default to remove space characters or enter a different RegEx to split the value into words.
- For the optional **Condition Threshold**, enter the minimum score required for the matcher to return 'True' on a rule.

> **Note:** Leave the Condition Threshold parameter empty when this matcher is used in more than one rule and the threshold varies based on the rule. For example, if one rule requires a match score of 70 while another rule requires 75, a default condition threshold can be confusing while troubleshooting. In that case, it is better to add the thresholds in the rules.

3. To test the configuration, for the Select Nodes parameters:



Select Nodes: Clean Cut Landscaping Tree Service (2034329) | Clean Cutt Landscaping Tree Service (551671)

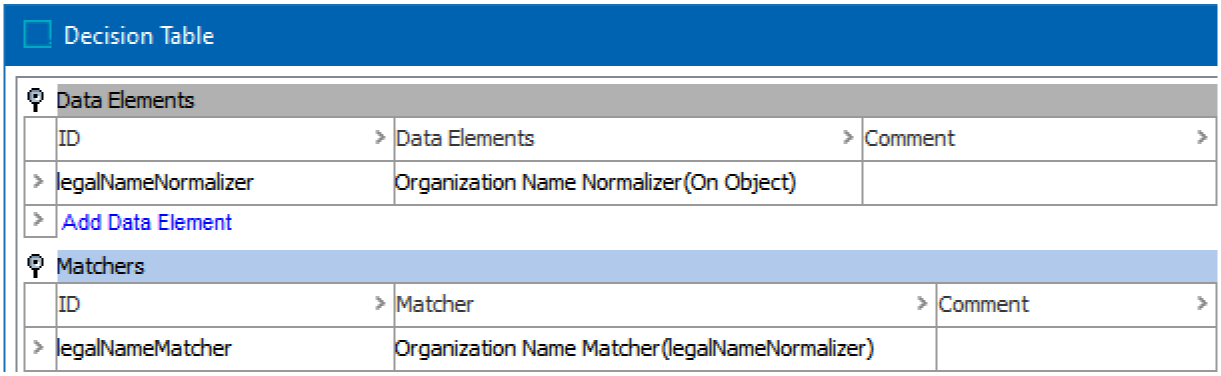
Result
70.0

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  0.0 is displayed when a value is not available in one of the selected nodes or when the organization names do not match. Adjust as indicated by the Evaluator results and repeat the evaluation.

  When red text is displayed, hover to review information about the record. For example, a record that has been deactivated, and so it produces no match code and thus no match score.

4. Click **OK** to save and display the configuration in the Matchers flipper.



Decision Table

Data Elements

ID	Data Elements	Comment
legalNameNormalizer	Organization Name Normalizer(On Object)	
Add Data Element		

Matchers

ID	Matcher	Comment
legalNameMatcher	Organization Name Matcher(legalNameNormalizer)	

Matcher: Person Name

The Person Name Normalizer data element (as defined in the **Data Element: Person Name Normalizer** topic) normalizes person name data for two objects. The Person Name Matcher compares the normalizer output and generates a match score (also called the 'rank score' in Web UI) based on the weighted sum of relevant data elements and match factors. This allows you to define which elements are more important during matching. The final score is a weighted sum of the combined first name and middle name, and the combined middle name and last name. Middle name is optional.

When a match score is applied to the defined rules (refer to the **Match Criteria Rules** topic), a final match score is determined to rank the likelihood of a match between the two objects.

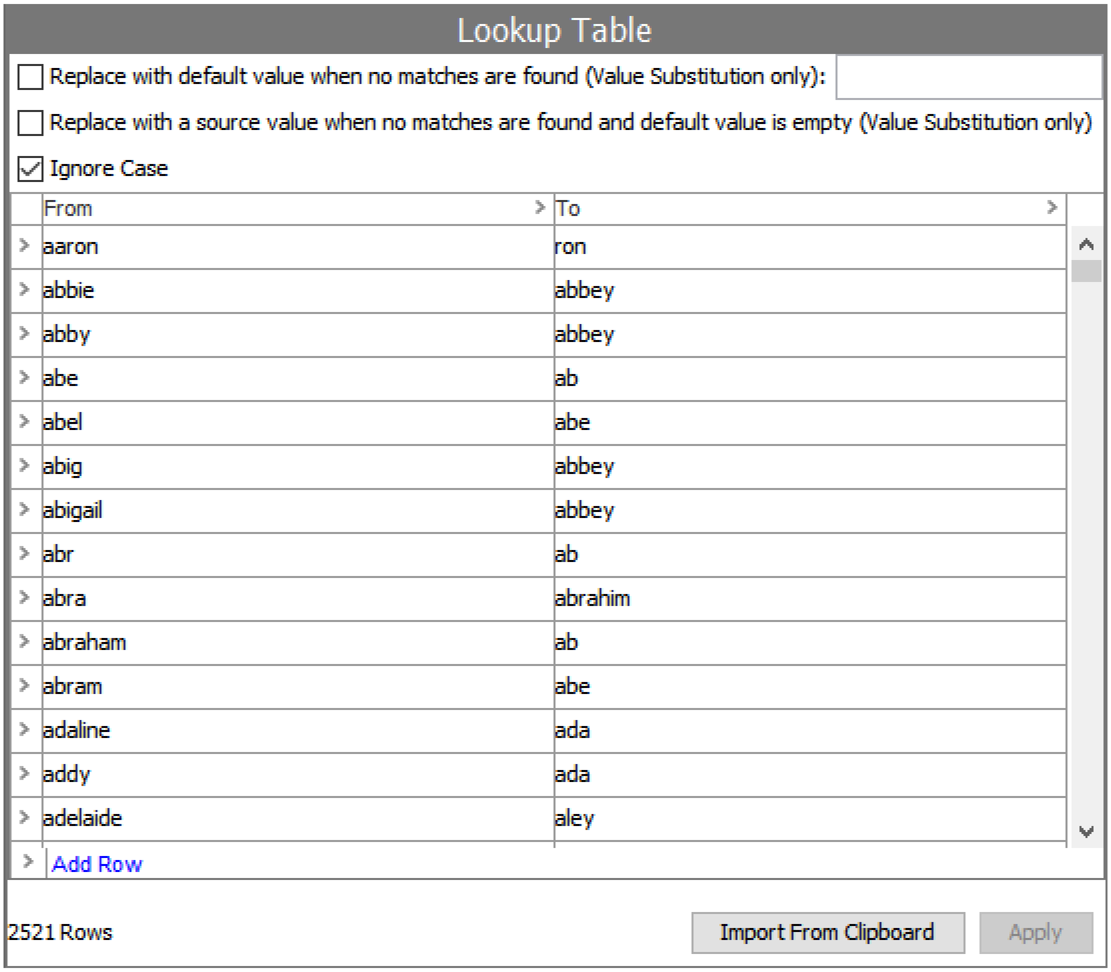
Note: If names are represented in a single field rather than split into first name and last name, use the Words Normalizer and Matcher instead of the Person Name Normalizer and Matcher.

Important: Stibo Systems recommends using the Machine Learning Matcher released with update 2024.1 as an alternative to the Person Name Matcher. The Machine Learning Matcher employs a pre-trained machine learning model to match person names and provides substantially more accurate scores. For details, refer to the Matcher: Machine Learning Matcher topic in the Matching, Linking, and Merging documentation.

Considerations

An **Unmatched Word Factor Table** assigns weights to individual words that may routinely be missing.

A **Word Alias Table** can be used to perform case-insensitive matching by alias. The Customer & Supplier MDM Configuration Guide in the Solution Enablement documentation refers to a Word Alias Table illustrated below that allows the matching to handle common name substitutions like Jasmine with Jasme or Jefferson with Jeff.



□ Replace with default value when no matches are found (Value Substitution only):

□ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

☑ Ignore Case

From	To
aaron	ron
abbie	abbey
abby	abbey
abe	ab
abel	abe
abig	abbey
abigail	abbey
abr	ab
abra	abrahim
abraham	ab
abram	abe
adaline	ada
addy	ada
adelaide	aley

2521 Rows

Input

The Person Name Matcher takes input from the selected Person Name Data Element and retrieves all person names for the two objects under comparison.

Functionality

The Person Name Matcher processes first names and last names separately, and optionally considers the middle name.

- No first name - scores 0 (unless the First Name Weight is also 0)
- No last name - scores 0 (unless the Last Name Weight is also 0)

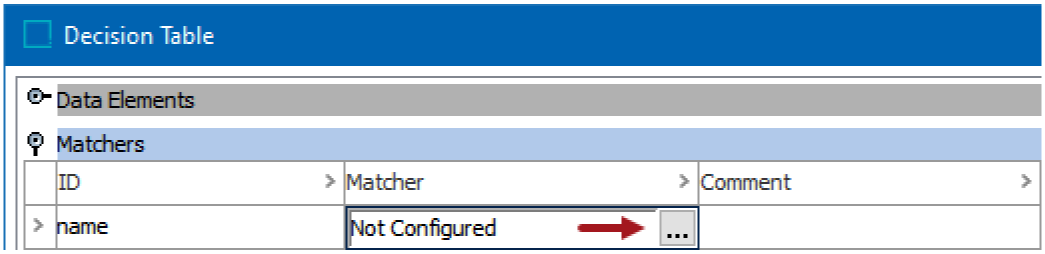
The comparison of each set of two person names includes:

1. Using the Name Word Splitter Regex to split the person name attribute value to create first, middle, and last name-tokens. If the Name Word Splitter Regex parameter is blank, the three names create a single name-token, and the names are compared as a whole.
2. Defining pairs based on 'first-name-tokens and middle-name-tokens' and 'last-name-tokens and middle-name-tokens' using the following methods:
    - Exact match – Name-tokens with at least two (2) characters that match exactly receive a score multiplied by the Exact Word Match Factor. Name-tokens with only one character are not considered exact matches but are instead treated as an initial.
    - Initials – Two name-tokens that are both a single character and are equal are matched as initials. The Initials Match Factor multiplier is applied.
    - Word Alias Table, if configured, performs case-insensitive matching by alias – Each word is scored individually. Any name-tokens that match based on the Word Alias Table is scored a multiplier equal to the Alias Word Match Factor. If the name-tokens do not match but have similar alias names, then the name-tokens are matched via the Alias Word Match Factor score multiplier.
    - Metaphone 3 matching – The algorithm (which expands on Soundex) compares names based on their pronunciation. It works well on English words, non-English words familiar to Americans, first names, and family names commonly found in the United States. The Metaphone 3 Word Match Factor multiplier is applied to a match by Metaphone 3. For more information on Metaphone 3, search the web.
    - Edit distance (adjusting for a few wrong characters due to typographical errors) – If both name-tokens are at least 3 characters long, and one can be made identical with the other by adding, deleting, or changing a single character, the score multiplier is equal to the Edit Distance Word Match Factor.
3. Finding no pairing between the first object and the second object on either first-and-middle-name-tokens or on last-and-middle-name-tokens - scores 0
4. First-and-middle-name-tokens have been paired, and last-and-middle-name-tokens have also been paired between the first object and second object under comparison, scores are calculated as follows:
    - First-and-middle-name-score and last-and-middle-name-score – Find the highest score from the first object to the second object and from the second object to the first object for each of first-and-middle-name-tokens and last-and-middle-name-tokens. Multiply these scores with the Word Out Of Order Factor, counting how many name-tokens are matched but out of sequence, and multiply the Word Out Of Order Factor with the score one time for each such sequence-mismatch. With the default Word Out Of Order Factor of 1.0, no penalties are applied for swapping the order of the name tokens.
    - Apply missing-token-multiplier – Count any name tokens in either the first object or the second object that is not matched in some way with any token in the other object. Every unmatched token causes another multiplication with the Missing Word Factor. Any token mentioned in the Unmatched Word Factor Table is exempt from this rule. If more than half the tokens in either first-and-middle-name-tokens or last-and-middle-name-tokens are unmatched, the Person Names are not considered a match, and receive a score of zero.
5. Determine the final score by comparing an person name from the first object to a person name from the second object, and also from the second object to the first object. The final score of an Person Name Matcher is the best score of matching any person name on the First object to any person name on the Second object.

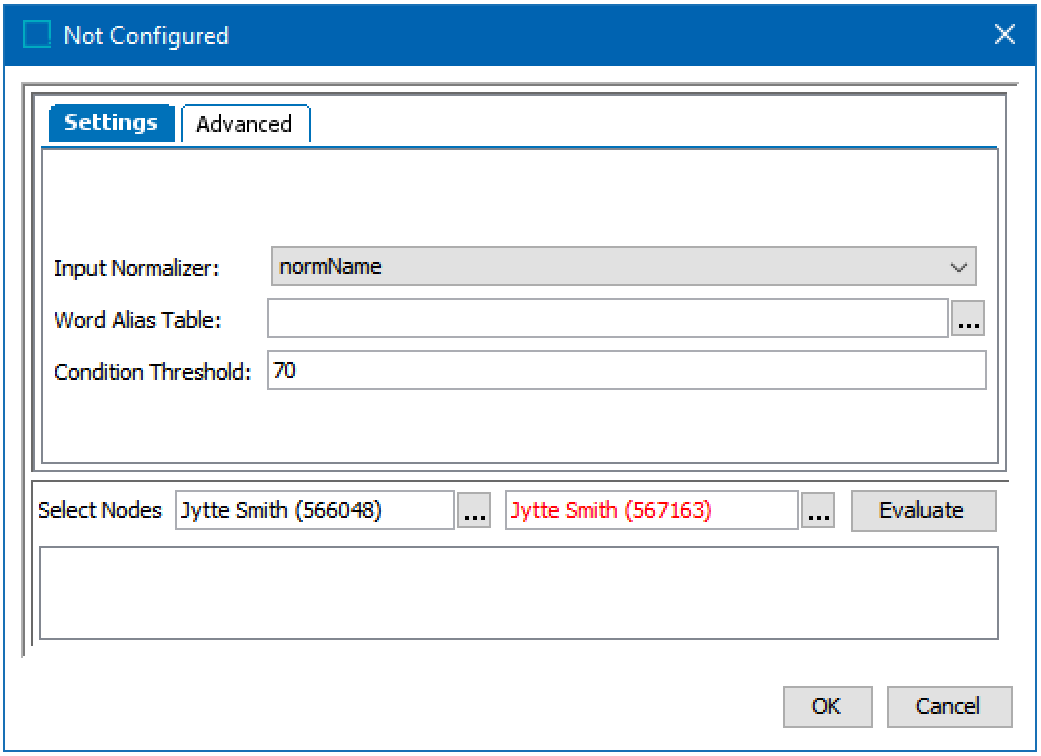
Configuring a Person Name Matcher

After adding the Person Name Matcher in the Matchers flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Matcher column and click the ellipsis button (...) to access the configuration dialog.



2. On the Not Configured dialog, the **Settings** tab is displayed.



Input Normalizer: normName

Word Alias Table:

Condition Threshold: 70

Select Nodes: Jytte Smith (566048) | Jytte Smith (567163)

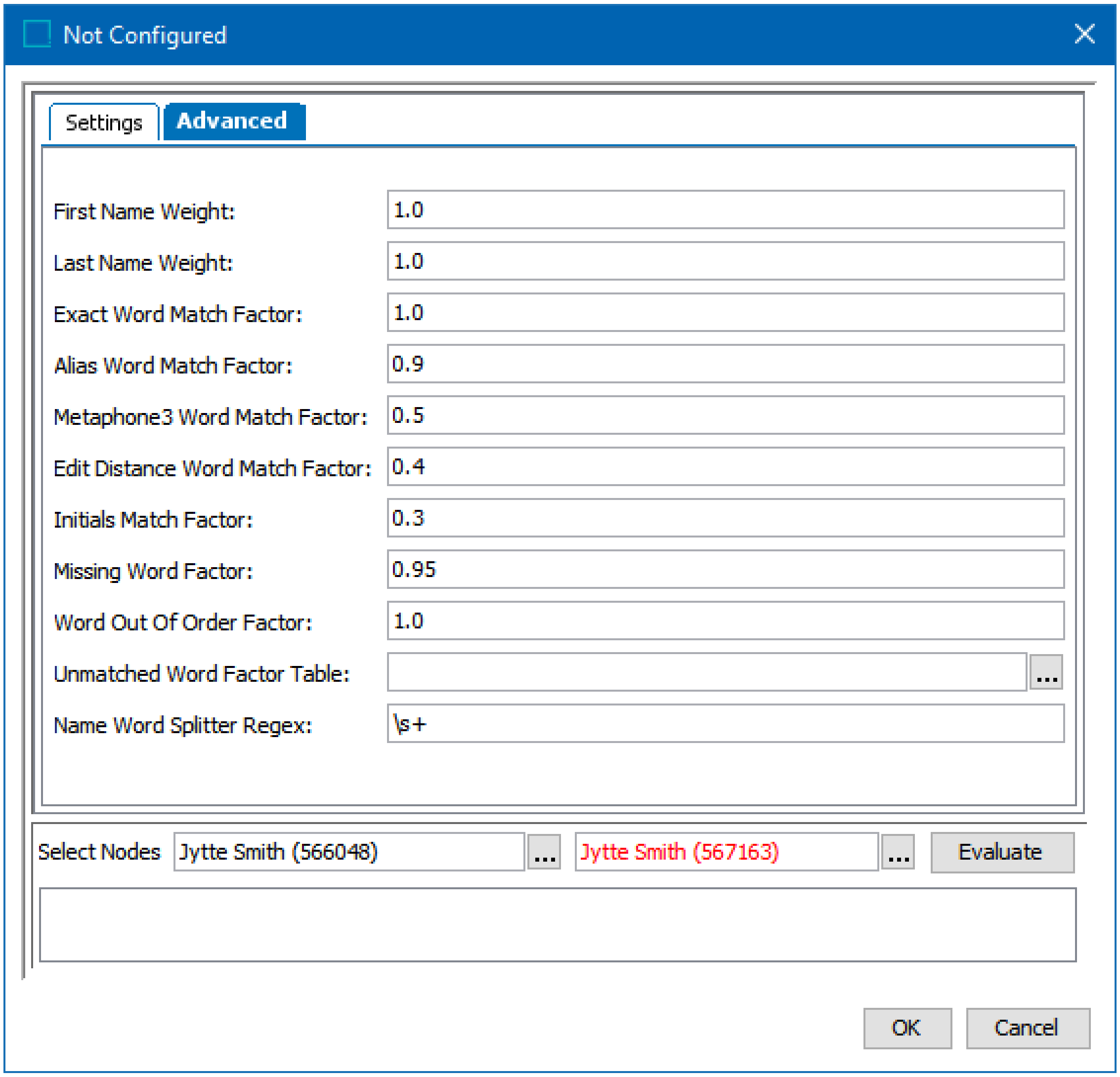
- For the required **Input Normalizer**, use the dropdown to select the associated Person Name Normalizer or enter a case-sensitive ID for the normalizer.
- For the optional **Word Alias Table**, click the ellipsis button (...) and select a Transformation Lookup Table to substitute words with the same or similar meaning.

- The optional **Name Word Splitter Regex** runs before applying the Word Alias Table. Refer to the **Considerations** section above.

- For the optional **Condition Threshold**, enter the minimum score required for the matcher to return 'True' on a rule.

> **Note:** Leave the Condition Threshold parameter empty when this matcher is used in more than one rule and the threshold varies based on the rule. For example, if one rule requires a match score of 70 while another rule requires 75, a default condition threshold can be confusing while troubleshooting. In that case, it is better to add the thresholds in the rules.

3. Click the **Advanced** tab and update the default weights and factors as needed.



First Name Weight: 1.0

Last Name Weight: 1.0

Exact Word Match Factor: 1.0

Alias Word Match Factor: 0.9

Metaphone3 Word Match Factor: 0.5

Edit Distance Word Match Factor: 0.4

Initials Match Factor: 0.3

Missing Word Factor: 0.95

Word Out Of Order Factor: 1.0

Unmatched Word Factor Table:

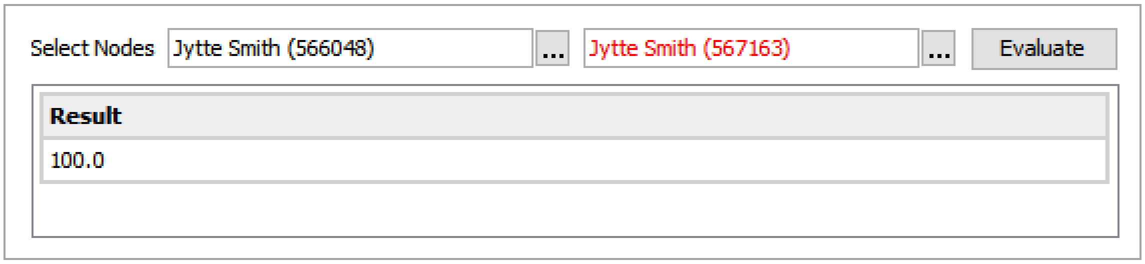
Name Word Splitter Regex: \s+

Select Nodes Jytte Smith (566048) Jytte Smith (567163)

- For the required **First Name Weight**, enter the relative weight of the combined 'first name and middle name' score versus and the combined 'middle name and last name' score.

- For the required **Last Name Weight**, enter the relative weight of the combined 'middle name and last name' score versus and the combined 'first name and middle name' score.
- For the required **Exact Word Match Factor**, enter how greatly exact matches influence the final score.
- For the required **Alias Word Match Factor**, enter how greatly words that are paired via aliases influence the final score.
- For the required **Metaphone3 Word Match Factor**, enter how greatly pairs via Metaphone 3 influence the final score.
- For the required **Edit Distance Word Match Factor**, enter how greatly pairs via edit distance influence the final score.
- For the required **Initials Word Match Factor**, enter how greatly pairs via initials influence the final score.
- For the required **Missing Word Factor**, enter how much unpaired or missing words penalize the final result. To modify the factor for specific words, select an Unmatched Word Factor Table in the parameter below.
- For the required **Word Out of Order Factor**, enter how much words that appear out of order penalize the final result.
- For the optional **Unmatched Word Factor Table**, click the ellipsis button (...) and select a Transformation Lookup Table to assign factors to certain words and increase or decrease the significance of the unmatched word. Unmatched words that are included in this lookup table use the factor in the table instead of the Missing Word Factor from the parameter above. Refer to the **Considerations** section above.
- For the optional **Name Word Splitter Regex**, leave the default to split names on space characters or enter a different RegEx to split the First Name, Middle Name, and Last Name values into individual words

4. To test the configuration, for the Select Nodes parameters:



Select Nodes: Jytte Smith (566048) ... Jytte Smith (567163) ...

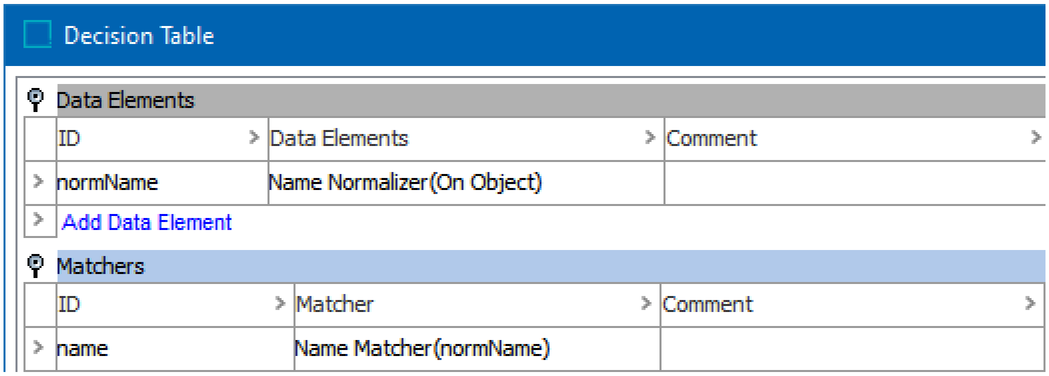
Result
100.0

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  0.0 is displayed when a value is not available in one of the selected nodes or when the phone numbers do not match. Adjust as indicated by the Evaluator results and repeat the evaluation.

Hover over the red text to review information about the record. In this example, the record has been deactivated, and so it produces no match code and thus no match score.

5. Click **OK** to save and display the configuration in the Matchers flipper.



Decision Table

Data Elements

ID	Data Elements	Comment
normName	Name Normalizer(On Object)	

Matchers

ID	Matcher	Comment
name	Name Matcher(normName)	

Matcher: Machine Learning Matcher

The Machine Learning Matcher employs a pre-trained machine learning model to match individual party data elements such as person names, emails, phone numbers, and addresses. The first pre-trained model version released with update 2024.1 supports person name matching exclusively.

The Machine Learning Matcher simplifies the matching process by improving the ability to create accurate and efficient matching algorithms for comparing person names. Additionally, the Machine Learning Matcher supports groups of nicknames, further simplifying the process of defining the nickname aliases.

> **Important:** The option to utilize the Machine Learning Matcher is exclusive to STEP SaaS systems. On-premises systems are not supported for implementing this matcher. Additionally, the use of the Machine Learning Matcher is exclusive to matching algorithms using embedded match codes. For more information, refer to the Match Codes topic in the Matching, Linking, and Merging documentation

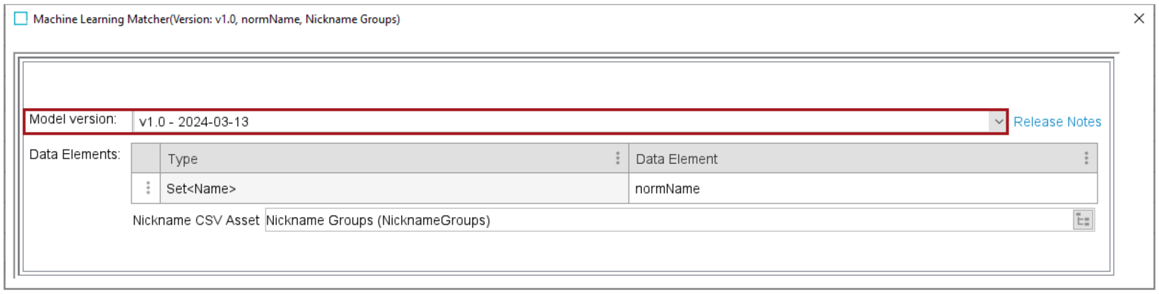
Model version

The Machine Learning Matcher has a model version concept that allows versioning of the matcher. New model versions are released outside of the normal STEP update cycle. In the Machine Learning Matcher editor dialog, there is a link to open 'Release Notes' that explain the changes done in each model version.

Every version of the model has a different underlying pre-trained machine learning model and will therefore produce different scores. Additionally, each model version can have different capabilities, supporting different data elements and producing different score output elements.

The versioning system consists of a major version number (first digit), a minor version number (second digit), and the date it was released. The rules governing these are:

- Major versions: Involves a change in supported input data elements and / or a change in output score elements.
- Minor versions: Involves a change to the scores, but the supported input data elements and output score elements remain the same.



Machine Learning Matcher(Version: v1.0, normName, Nickname Groups)

Model version: v1.0 - 2024-03-13

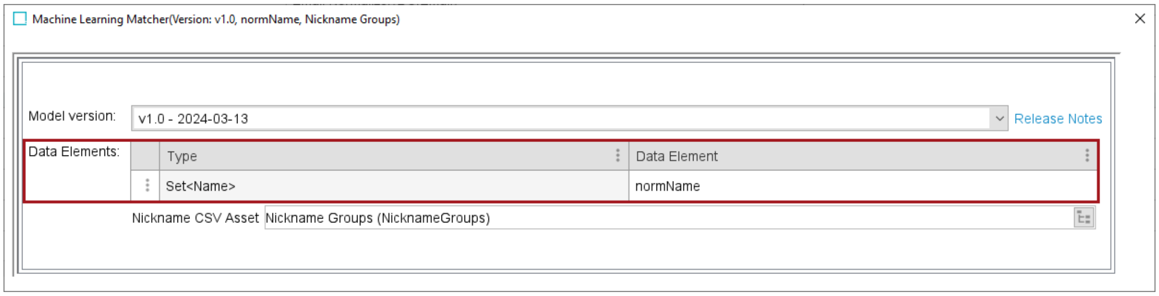
Data Elements:

Type	Data Element
Set<Name>	normName

Nickname CSV Asset Nickname Groups (NicknameGroups)

Data Elements

The Machine Learning Matcher takes input from the Data Elements that are selected in the matcher.



Machine Learning Matcher(Version: v1.0, normName, Nickname Groups)

Model version: v1.0 - 2024-03-13

Data Elements:

Type	Data Element
Set<Name>	normName

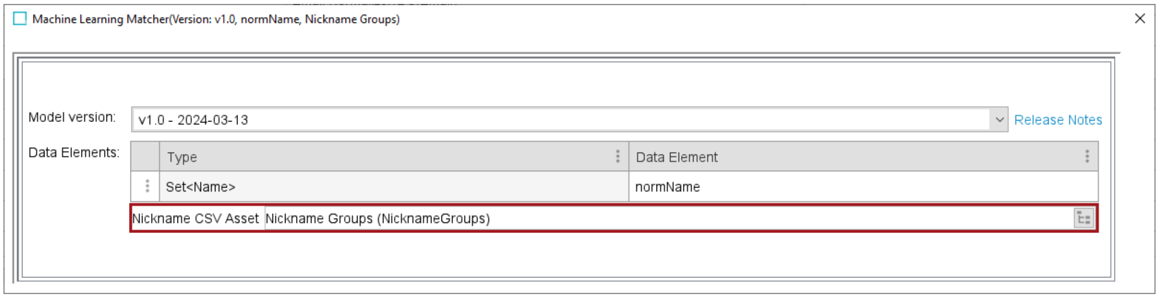
Nickname CSV Asset Nickname Groups (NicknameGroups)

It is possible to configure only part of the data elements. The output scores corresponding to unconfigured data elements will always be 0.

> **Note:** Subsequent model versions will be released outside of the normal STEP update cycle to support additional party data object types.

Nickname CSV Asset

Some model versions support a ‘Nickname CSV Asset’ which is a CSV file containing nickname alias groups that will be used in the person name matching. The file enables STEP to provide additional information to the matching process, facilitating the identification of names that are nicknames or shorthand versions of longer names.



Machine Learning Matcher(Version: v1.0, normName, Nickname Groups)

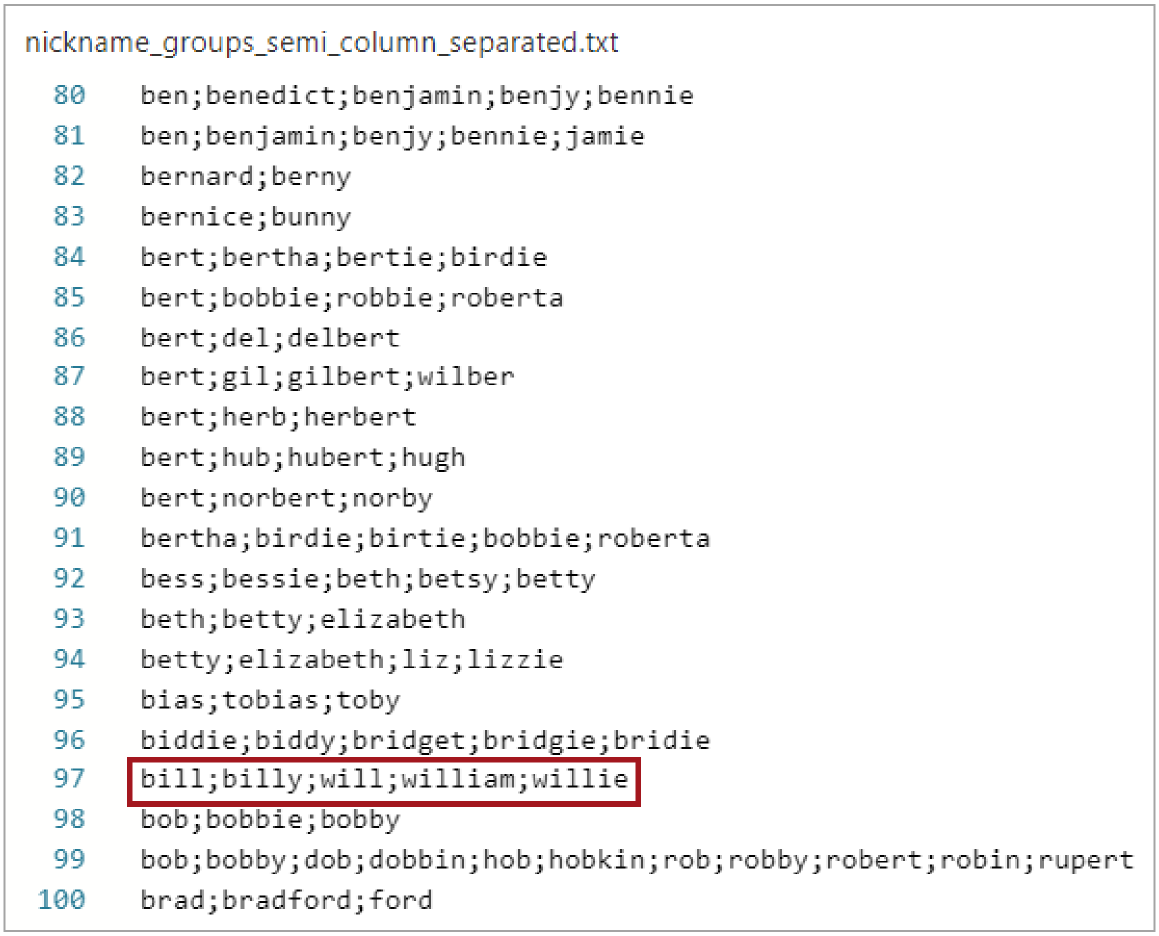
Model version: v1.0 - 2024-03-13

Data Elements:

Type	Data Element
Set<Name>	normName

Nickname CSV Asset Nickname Groups (NicknameGroups)

For example, if a person is registered under different names like ‘Bill’ or ‘William’, the matcher might return a low name match score. By providing the Nickname CSV Asset, the matching service can recognize the match between the two names, resulting in a higher score. Data stewards can maintain and adjust the Nickname CSV Asset to suit their company’s unique data requirements.



nickname_groups_semi_column_separated.txt

```
ben;benedict;benjamin;benjy;bennie
ben;benjamin;benjy;bennie;jamie
bernard;berny
bernice;bunny
bert;bertha;bertie;birdie
bert;bobbie;robbie;roberta
bert;del;delbert
bert;gil;gilbert;wilber
bert;herb;herbert
bert;hub;hubert;hugh
bert;norbert;norby
bertha;birdie;birtie;bobbie;roberta
bess;bessie;beth;betsy;betty
beth;betty;elizabeth
betty;elizabeth;liz;lizzie
bias;tobias;toby
biddie;biddy;bridget;bridgie;bridie
bill;billy;will;william;willie
bob;bobbie;bobby
bob;bobby;dob;dobbin;hob;hobkin;rob;robby;robert;robin;rupert
brad;bradford;ford
```

All names on each row of the file are considered part of a nickname group, and all names are handled equally, meaning that the order of the names has no significance.

Additionally, the nickname groups can be utilized when generating match codes. For more information, refer to the Match Code Generator: Person Name and Address topic in the Matching, Linking, and Merging documentation.

When creating the CSV file, Stibo Systems recommends that users adhere to the following guidance to prevent errors:

- **Semicolon is a reserved character.** Avoid using semicolons within names, as they serve as separators between names.
- **Avoid line breaks in nickname values.** Nickname values should not contain line breaks, as this will be interpreted as the start of a new nickname group.
- **Support for ‘Newline’ formats.** The system supports ‘Newline’ formats, including CR LF, LF, and CR.
- **UTF-8 file format is required.** The CSV file must be in the UTF-8 format to ensure compatibility with the system.

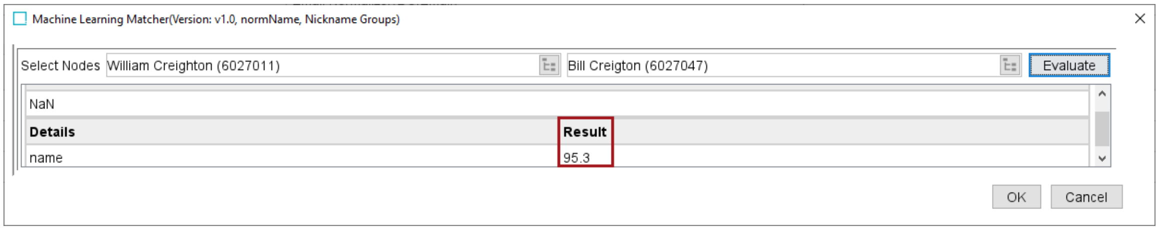
- **Lines without a semicolon are ignored.** Lines lacking a semicolon are disregarded. Ensure that a semicolon is included as a name separator to have the line included in the output.
- **Tabs and spaces are trimmed.** White-space characters at the beginning or end of a line are removed, so avoid using them.
- **Multiple tokens separated with white-space(s) will be ignored.** Nicknames should consist of a single name without internal white-space.

A default Nickname CSV Asset, containing typical US nicknames, can be acquired by contacting Stibo Systems at cmdm@stibo.com.

Output scores

The Machine Learning Matcher produces individual scores for each of the configured elements in the Model version, e.g., Names, Addresses, Emails, and Phone numbers. Some Model versions for person name matching also facilitate the generation of name subscores, such as ‘name.firstname’ and ‘name.lastname’. These subscores are derived from the overall ‘Name’ score, providing more detailed information about the first name and the last name match scores.The score hierarchy is displayed with the use of dot (.) notation.

In the example below, the matcher returns an overall match score of 95.3 when comparing the two selected names.



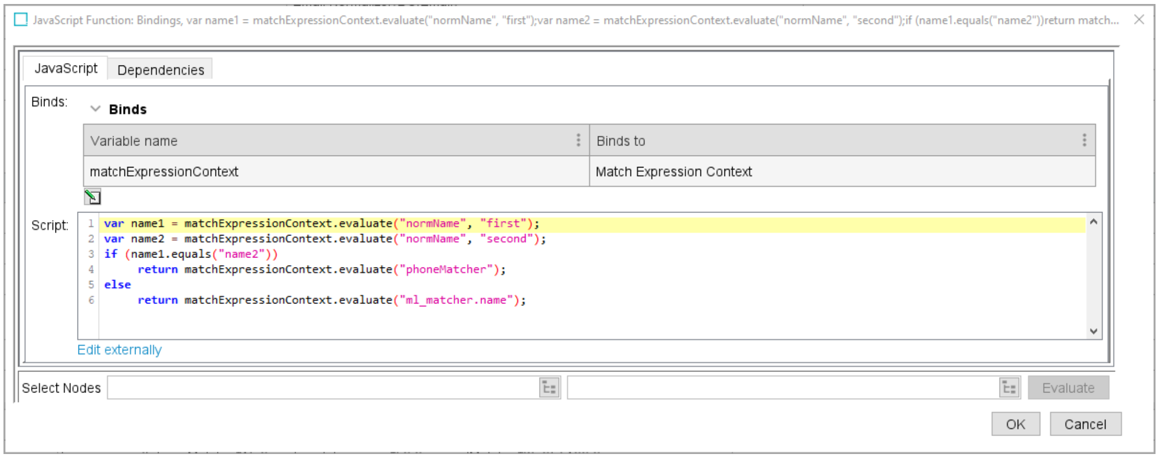
Machine Learning Matcher(Version: v1.0, normName, Nickname Groups)

Select Nodes William Creighton (6027011) | Bill Creigton (6027047)

NaN

Details	Result
name	95.3

All individual scores and subscores can be used in Rules in the Match Criteria as well as in Function and JavaScript Function matchers, using the same dot notation. For more information, refer to the Matcher: JavaScript Function topic in the Matching, Linking, and Merging documentation.



JavaScript Function: Bindings, var name1 = matchExpressionContext.evaluate("normName", "first");var name2 = matchExpressionContext.evaluate("normName", "second");if (name1.equals("name2"))return match...

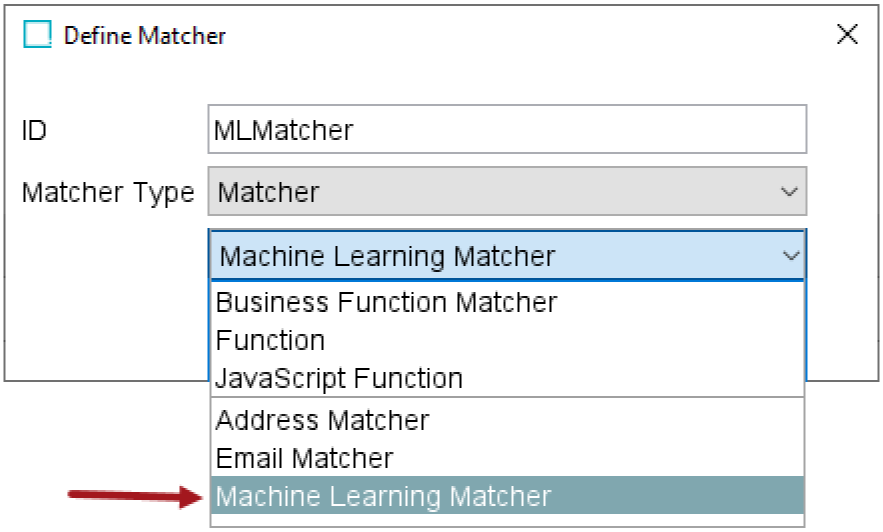
Variable name	Binds to
matchExpressionContext	Match Expression Context

Script:

```
var name1 = matchExpressionContext.evaluate("normName", "first");
var name2 = matchExpressionContext.evaluate("normName", "second");
if (name1.equals("name2"))
      return matchExpressionContext.evaluate("phoneMatcher");
else
      return matchExpressionContext.evaluate("ml_matcher.name");
```

Configuring a Machine Learning Matcher

The Machine Learning Matcher can be added in the ‘Matchers’ flipper of the Decision Table dialog by clicking the ‘Add Matcher’ link (as defined in the Match Criteria topic of the Matching, Linking, and Merging documentation).



Define Matcher

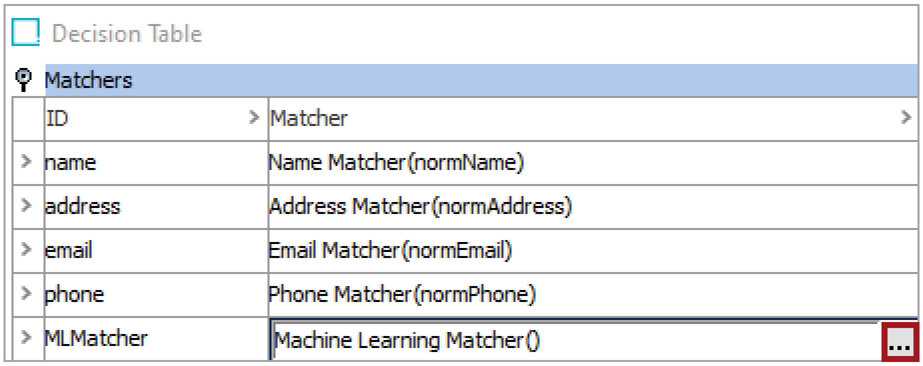
ID: MLMatcher

Matcher Type: Matcher

Machine Learning Matcher

After adding the Machine Learning Matcher, configure it as follows:

1. Click into the ‘Matcher’ column and click the click the ellipsis button (...) to access the configuration dialog.

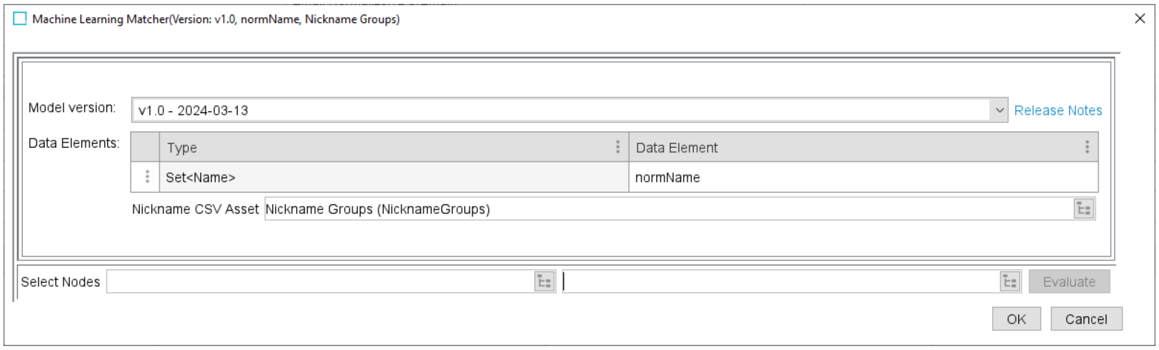


Decision Table

Matchers

ID	Matcher
name	Name Matcher(normName)
address	Address Matcher(normAddress)
email	Email Matcher(normEmail)
phone	Phone Matcher(normPhone)
MLMatcher	Machine Learning Matcher() ...

2. The configuration dialog for the Machine Learning Matcher opens.



Machine Learning Matcher(Version: v1.0, normName, Nickname Groups)

Model version: v1.0 - 2024-03-13

Data Elements:

Type	Data Element
Set<Name>	normName

Nickname CSV Asset: Nickname Groups (NicknameGroups)

Select Nodes

- To choose the **Model version**, click the dropdown to select the desired pre-trained model. The dropdown provides a list of available model versions. By default, the latest version is selected.
- Clicking the **Release Notes** link will display a table showing the release notes for all available model versions. The table includes information such as the version number, the release date, and the release note information itself.
- In **Data Elements**, a table is available with the ‘Type’ and ‘Data Element’ fields. The ‘Type’ field is pre-populated with the supported types for the model version selected. To specify the data elements from which the Machine Learning Matcher should obtain input, click into the ‘Data Element’ field and make a selection.
- Model versions that support person name matching often also support nickname groups. To provide a **Nickname CSV Asset** containing nickname alias groups, click the ellipsis button (...) and browse to select the file. Before supplying the CSV asset, a new object type for Person Name Alias Groups must be added.

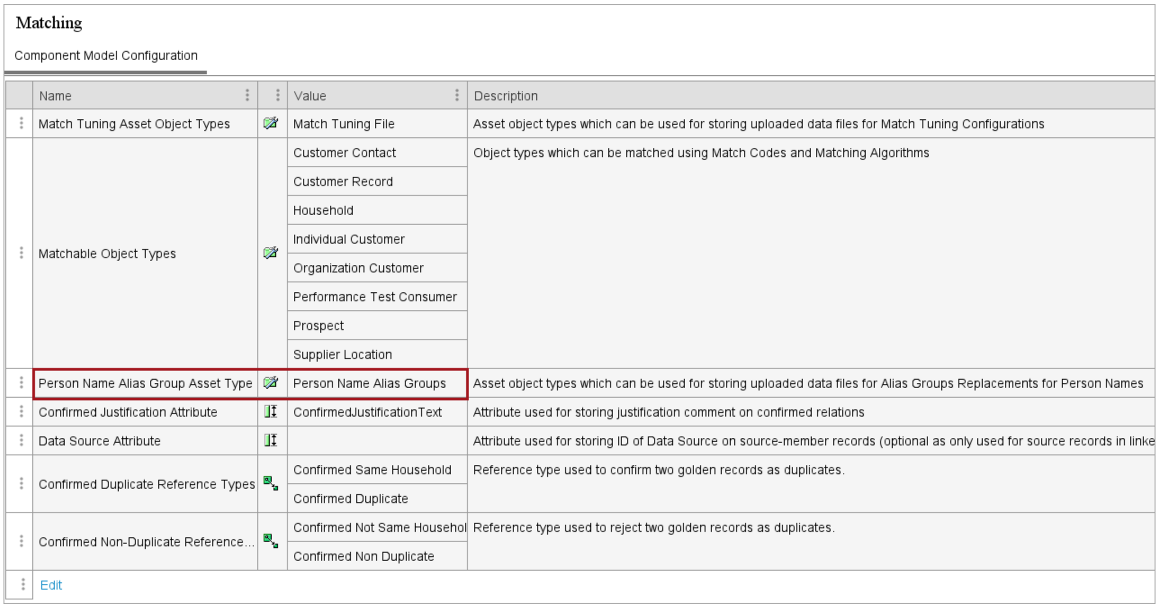
3. To evaluate the configuration of the data model for the **Select Nodes** parameter:
   - Click the ellipsis button (...)for each field and select two objects for comparison.
   - Click the **Evaluate** button.

     When evaluating the two nodes, the Machine Learning Matcher produces individual scores for each of the configured elements in the Data Element field, e.g., Names, Addresses, Emails, and Phone numbers. Additional subscores are also displayed if the selected model version supports it.
4. Click **OK** to save and display the configuration in the 'Matchers' flipper.

Configuring new object type for Person Name Alias Groups

Before providing a Nickname CSV Asset, it is necessary to add a new object type for Person Name Alias Groups. Subsequently, configure the Matching Component Model with this newly added object type.

For details, refer to the Configuring Matching Component Model topic in the Matching, Linking, and Merging documentation.

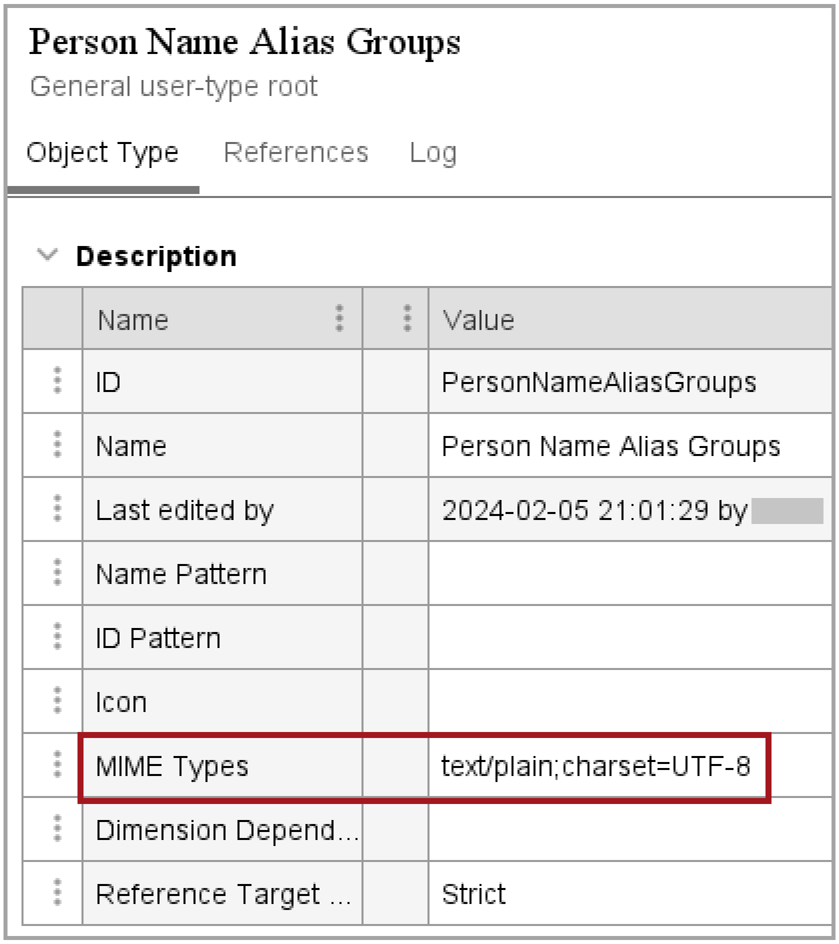


Matching

Component Model Configuration

Name		Value	Description
Match Tuning Asset Object Types		Match Tuning File	Asset object types which can be used for storing uploaded data files for Match Tuning Configurations
Matchable Object Types		Customer Contact Customer Record Household Individual Customer Organization Customer Performance Test Consumer Prospect Supplier Location	Object types which can be matched using Match Codes and Matching Algorithms
Person Name Alias Group Asset Type		Person Name Alias Groups	Asset object types which can be used for storing uploaded data files for Alias Groups Replacements for Person Names
Confirmed Justification Attribute		ConfirmedJustificationText	Attribute used for storing justification comment on confirmed relations
Data Source Attribute			Attribute used for storing ID of Data Source on source-member records (optional as only used for source records in linke
Confirmed Duplicate Reference Types		Confirmed Same Household Confirmed Duplicate	Reference type used to confirm two golden records as duplicates.
Confirmed Non-Duplicate Reference…		Confirmed Not Same Househol Confirmed Non Duplicate	Reference type used to reject two golden records as duplicates.
Edit			

Ensure that the object type is configured with the **MIME Type** 'Text/plain; charset=UTF-8' to allow the matching service to recognize and read the content of the file. For details, refer to the ‘Object Types and Structures’ section of the System Setup documentation.



Person Name Alias Groups

General user-type root

Name		Value
ID		PersonNameAliasGroups
Name		Person Name Alias Groups
Last edited by		2024-02-05 21:01:29 by [redacted]
Name Pattern		
ID Pattern		
Icon		
MIME Types		text/plain;charset=UTF-8
Dimension Depend...		
Reference Target ...		Strict

Now, it is possible to add the Nickname CSV file to a new asset using the newly created object type and select the asset when configuring the Machine Learning Matcher.

Support Guidelines for the Machine Learning Matcher

The scores provided by the pre-trained Machine Learning Matcher often depend on subjective assessment. Nevertheless, Stibo Systems aims to continuously improve the quality and accuracy of these scores. A dedicated team is ready to engage in a collaborative process to help improve the customer's understanding of the produced scores and to improve the accuracy of the scores through the release of new versions of our pre-trained models.

For detailed information about the Machine Learning Matcher, refer to the Matcher: Machine Learning Matcher topic in the Matching, Linking, and Merging documentation.

If the matching scores you receive result in questions for you or your team, in the Support Portal, create a ticket with the following details:

- Summary: Preface your summary content with 'ML Matcher' so it is clear to the support team that the issue relates to the pre-trained Machine Learning Matcher. The format will look like this: 'ML Matcher - <summary of the issue>'.
- Description: Specify the Model version in use. Provide examples of pairs with current scores and expected scores, and present clear arguments regarding the discrepancies. The examples must include the output of the data elements (normalizers) that are used as input for the Machine Learning Matcher.
- Issue Type: 'Matching, Linking & Merging'.

Matcher: Phone

The Phone Normalizer data element (as defined in the **Data Element: Phone Normalizer** topic) normalizes phone data for two objects. The Phone Matcher compares the normalizer output and generates a match score (also called the 'rank score' in Web UI).

When a match score is applied to the defined rules (refer to the **Match Criteria Rules** topic), a final match score is determined to rank the likelihood of a match between the two objects.

Input

The Phone Matcher takes input from the selected phone data element and retrieves all phone numbers for the two objects under comparison.

Functionality

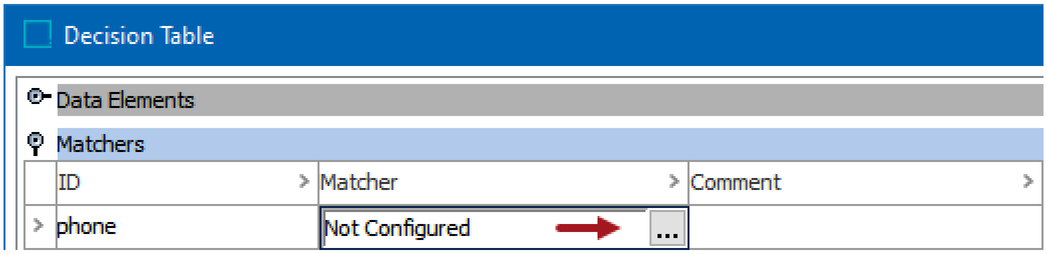
The Phone Matcher compares phone numbers between two objects:

- If there is an exact match, the score is 100.
- If the phone numbers match but have different country codes, the score is 80.
- If there are no matching numbers, the score is 0.

Configuring a Phone Matcher

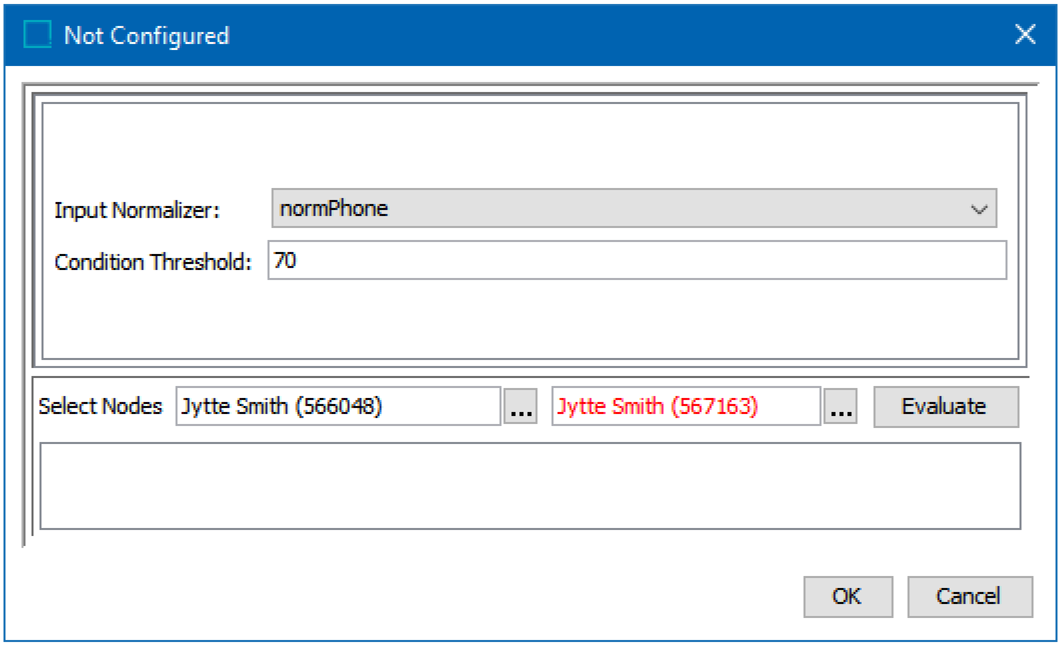
After adding the Phone Matcher in the Matchers flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Matcher column and click the ellipsis button (...) to access the configuration dialog.



ID	Matcher	Comment
phone	Not Configured ...	

2. On the Not Configured dialog:



Input Normalizer: normPhone

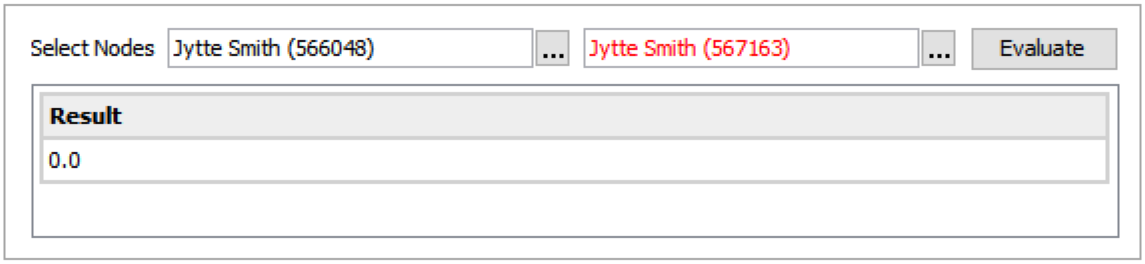
Condition Threshold: 70

Select Nodes: Jytte Smith (566048) | Jytte Smith (567163)

- For the required **Input Normalizer**, use the dropdown to select the associated Phone Normalizer or enter a case-sensitive ID for the normalizer.
- For the optional **Condition Threshold**, enter the minimum score required for the matcher to return 'True' on a rule.

> **Note:** Leave the Condition Threshold parameter empty when this matcher is used in more than one rule and the threshold varies based on the rule. For example, if one rule requires a match score of 70 while another rule requires 75, a default condition threshold can be confusing while troubleshooting. In that case, it is better to add the thresholds in the rules.

3. To test the configuration, for the Select Nodes parameters:



Select Nodes: Jytte Smith (566048) | Jytte Smith (567163)

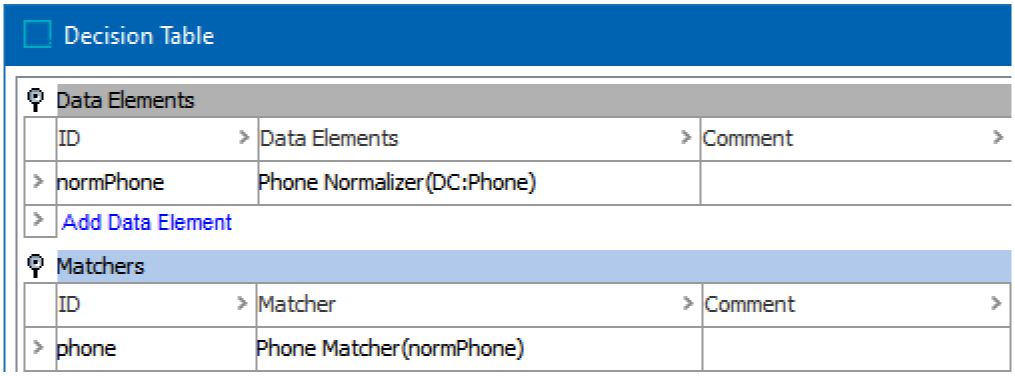
Result
0.0

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button to show the score.

0.0 is displayed when a value is not available in one of the selected nodes or when the phone numbers do not match. Adjust as indicated by the Evaluator results and repeat the evaluation.

When red text is displayed, hover to review information about the record. For example, a record that has been deactivated, and so it produces no match code and thus no match score.

4. Click **OK** to save and display the configuration in the Matchers flipper.



Decision Table

Data Elements

ID	Data Elements	Comment
normPhone	Phone Normalizer(DC:Phone)	
Add Data Element		

Matchers

ID	Matcher	Comment
phone	Phone Matcher(normPhone)	

Matcher: Words

The Words Normalizer data element (as defined in the **Data Element: Words Normalizer** topic) normalizes word data for two objects. The Words Matcher compares the normalizer output and generates a match score (also called the 'rank score' in Web UI) based on the weighted sum of relevant data elements and match factors. This allows you to define which elements are more important during matching.

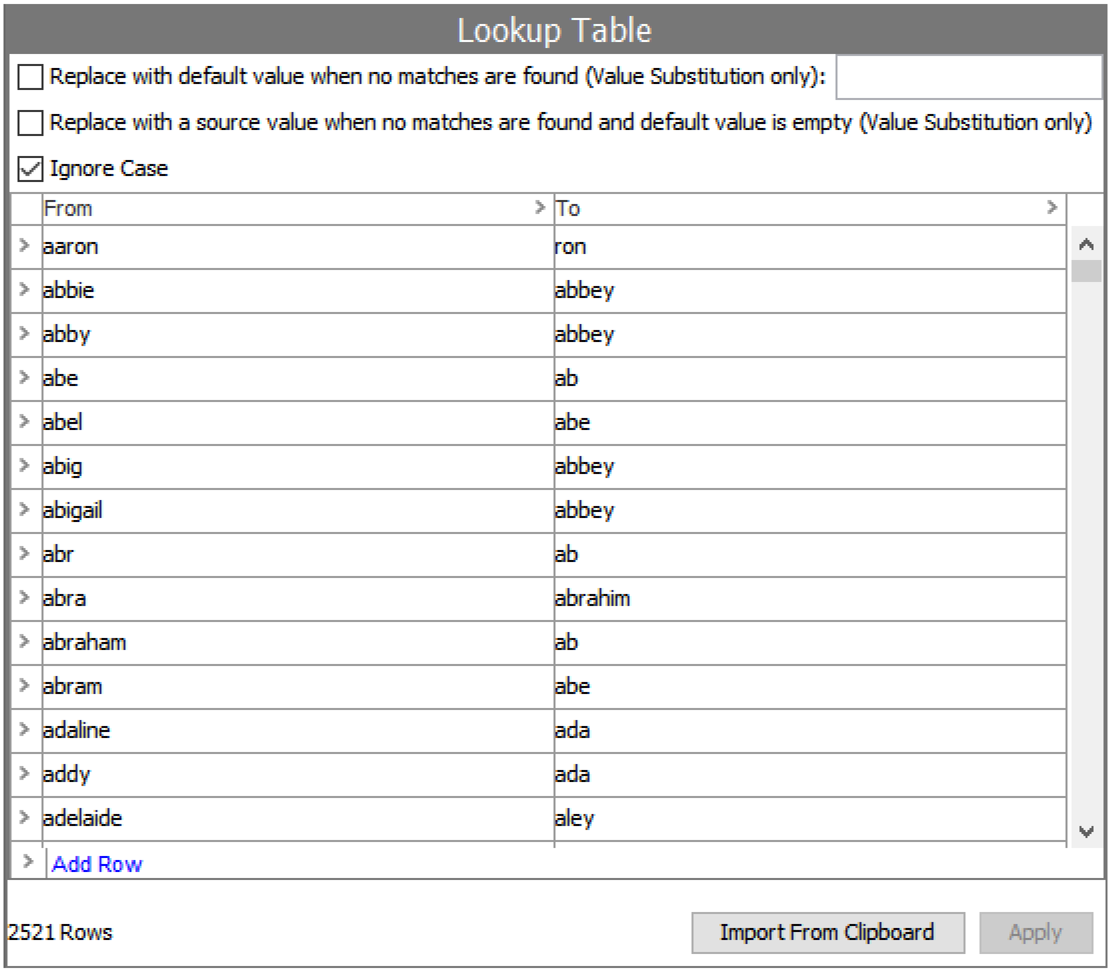
When a match score is applied to the defined rules (refer to the **Match Criteria Rules** topic), a final match score is determined to rank the likelihood of a match between the two objects.

> **Note:** The Words Normalizer and Words Matcher are generic and can handle multi-word values including a wide range of data, such as customer names and social security numbers.

Considerations

An **Unmatched Word Factor Table** assigns weights to individual words that may be routinely be missing.

A **Word Alias Table** can be used to perform case-insensitive matching by alias. The Customer & Supplier MDM Configuration Guide in the Solution Enablement documentation refers to a Word Alias Table illustrated below that allows the matching to handle common name substitutions like Jasmine with Jasme or Jefferson with Jeff. A similar lookup table can be configured for words commonly encountered by this matcher.



Lookup Table

□ Replace with default value when no matches are found (Value Substitution only):

□ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

☑ Ignore Case

From	To
aaron	ron
abbie	abbey
abby	abbey
abe	ab
abel	abe
abig	abbey
abigail	abbey
abr	ab
abra	abrahim
abraham	ab
abram	abe
adaline	ada
addy	ada
adelaide	aley

2521 Rows

Input

The Words Matcher takes input from the selected Words Data Element configured as List<String> for the two objects under comparison. The word-string provided as input to the Words Matcher may consist of several individual word-tokens.

Functionality

The Words Matcher processes a word-token from the first object with any word-token from the second object.

The comparison of each set of two word-tokens includes:

1. Using the Word Splitter Regex to split the word-string value into word-tokens for comparison or create a word-token identical to the word-string when the Word Splitter Regex parameter is blank.

2. Defining pairs based on word-token using the following methods:

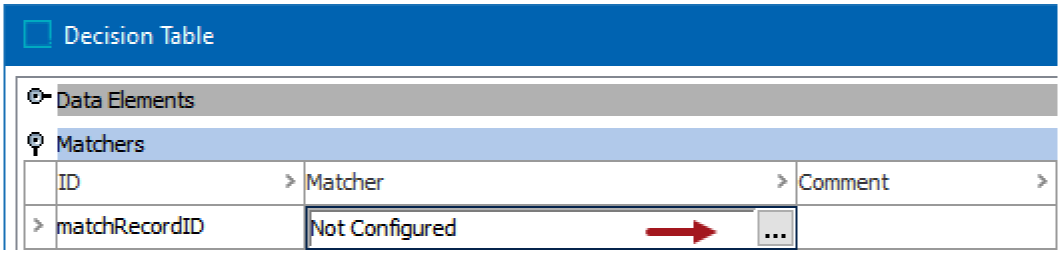
- Exact matching – applies the Exact Word Match Factor as a multiplier to the score.
- Word Alias Table, if configured, performs case-insensitive matching by alias – Each word-token is scored individually. Any word-tokens that match based on the Word Alias Table is scored a multiplier equal to the Alias Word Match Factor. For example, 'Ajax Cleaning Supplies Co' compared to 'Ajax Cleaning Supplies Company' results in three (3) exact matches, and the Word Alias Table allows 'Co' to match with 'Company' so the Alias Word Match Factor is applied once. If the name-tokens do not match but have similar alias names, then the name-tokens are matched but with the Alias Word Match Factor score multiplier.
- Metaphone 3 matching – The algorithm (which expands on Soundex) compares names based on their pronunciation. It works well on English words, non-English words familiar to Americans, first names, and family names commonly found in the United States. The Metaphone 3 Word Match Factor multiplier is applied to a match by Metaphone 3. For more information on Metaphone 3, search the web.
- Edit distance matching (adjusting for a few wrong characters due to typographical errors) – If both name-tokens are at least 3 characters long, and one can be made identical with the other by adding, deleting, or changing a single character, the score multiplier is equal to the Edit Distance Word Match Factor.
- Sequence matching – If tokens are out of order, a further penalty multiplier is determined by the Word Out Of Order Factor. For example, 'Ajax Cleaning Supplies' compared to 'Cleaning Supplies Ajax'.
- Unmatched / Missing matching – If there are missing tokens, the score is penalized by multiplying with the Missing Word Factor. For example, 'Ajax Company Cleaning Supplies Inc' compared to 'Ajax Cleaning Supplies'. Since two words are missing, the factor is applied twice. The Customer & Supplier MDM Configuration Guide in the Solution Enablement documentation includes an Unmatched Word Factor Table that assigns the word 'Company' a special weight of 0.98 if exactly that word is missing, since it is often left out by people writing company names. If more than half the word-tokens are unpaired they considered not matching.

3. Determine the final score by identifying the best score of matching any word-token on the first object to any word-token on the second object as defined by the following calculation: WordString Score= PairScore* MissingTokensMultiplier* OutOfOrderMultiplier*100

Configuring a Words Matcher

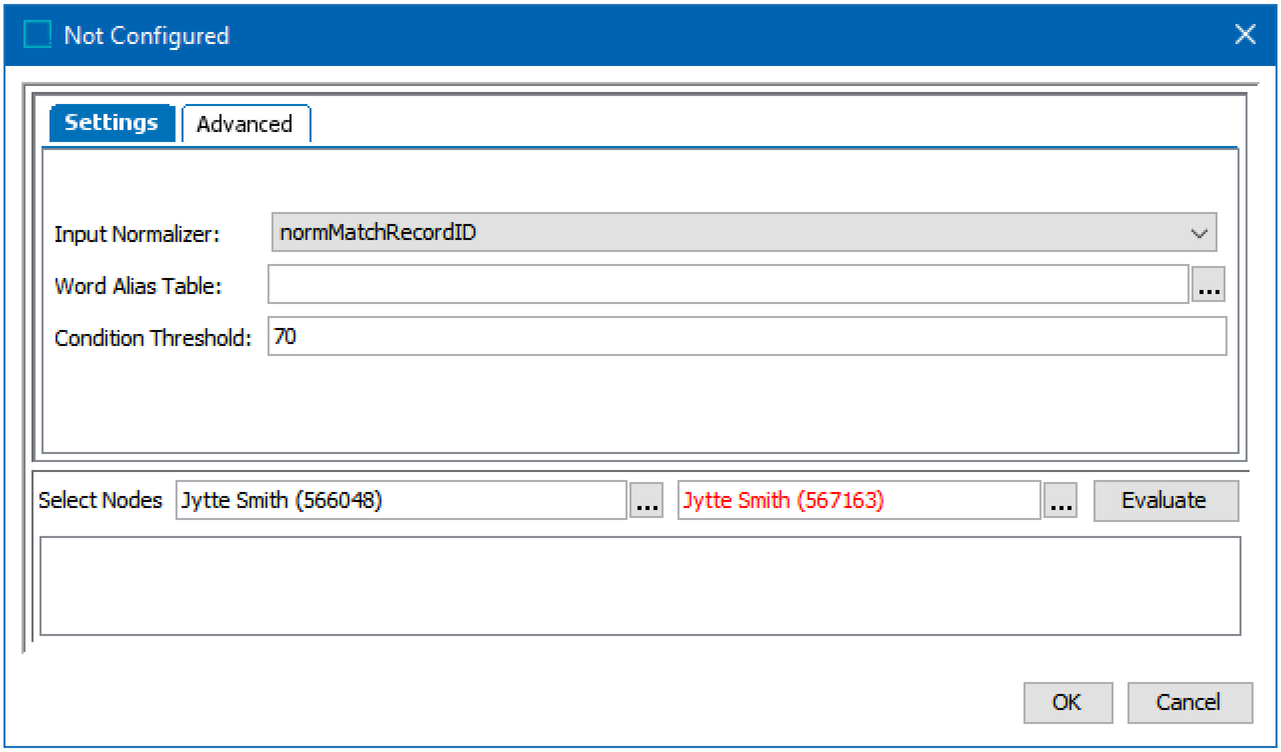
After adding the Words Matcher in the Matchers flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Matcher column and click the ellipsis button (...) to access the configuration dialog.



ID	Matcher	Comment
matchRecordID	Not Configured	

2. On the Not Configured dialog, the **Settings** tab is displayed.



Not Configured

Settings | Advanced

Input Normalizer: normMatchRecordID

Word Alias Table:

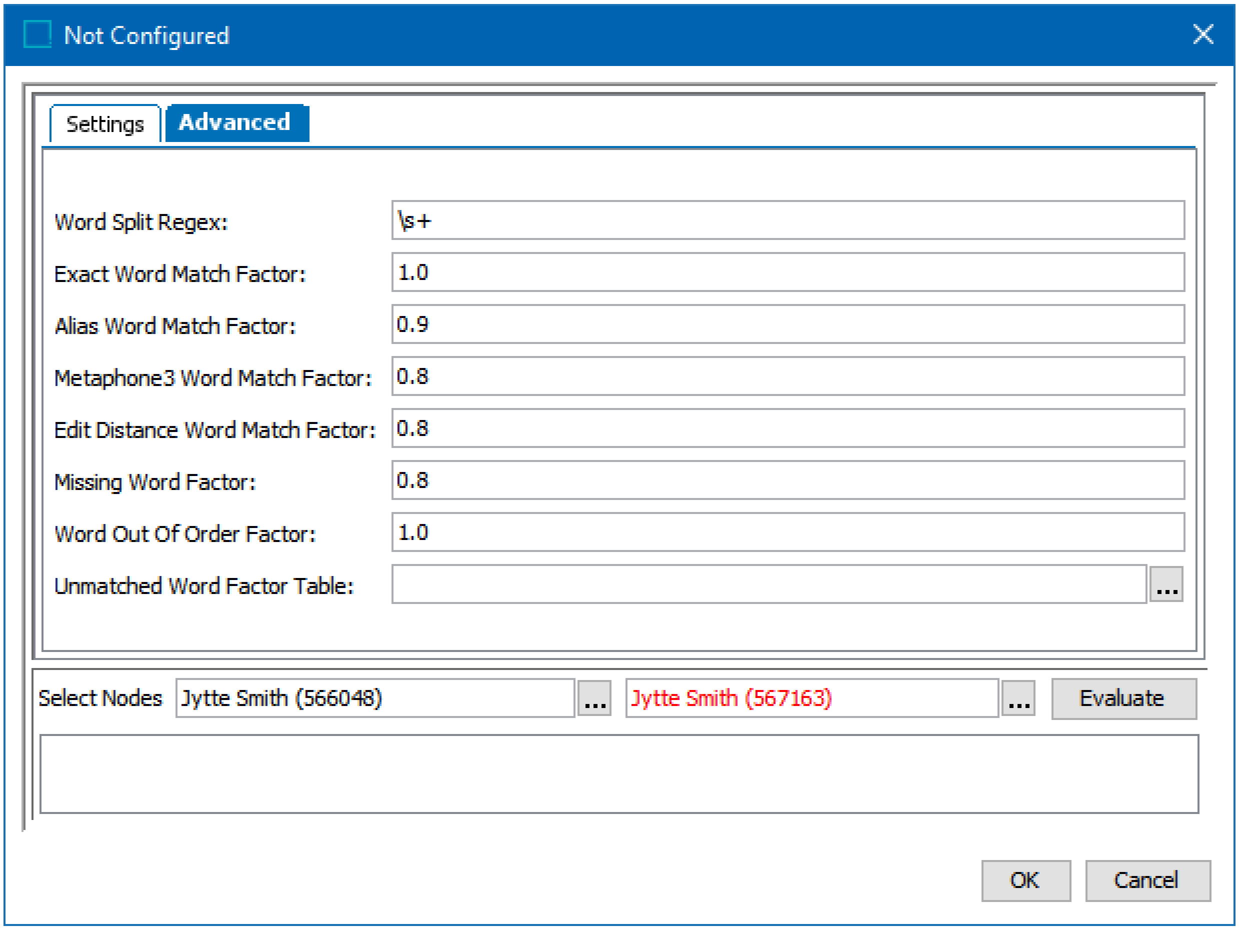
Condition Threshold: 70

Select Nodes Jytte Smith (566048) ... Jytte Smith (567163) ...

- For the required **Input Normalizer**, use the dropdown to select the associated Person Name Normalizer or enter a case-sensitive ID for the normalizer.
- For the optional **Word Alias Table**, click the ellipsis button (...) and select a Transformation Lookup Table to substitute words with the same or similar meaning. The optional **Name Word Splitter Regex** runs before applying the Word Alias Table. Refer to the **Considerations** section above.
- For the optional **Condition Threshold**, enter the minimum score required for the matcher to return 'True' on a rule.

> **Note:** Leave the Condition Threshold parameter empty when this matcher is used in more than one rule and the threshold varies based on the rule. For example, if one rule requires a match score of 70 while another rule requires 75, a default condition threshold can be confusing while troubleshooting. In that case, it is better to add the thresholds in the rules.

3. Click the **Advanced** tab and update the default weights and factors as needed.



Word Split Regex: \s+

Exact Word Match Factor: 1.0

Alias Word Match Factor: 0.9

Metaphone3 Word Match Factor: 0.8

Edit Distance Word Match Factor: 0.8

Missing Word Factor: 0.8

Word Out Of Order Factor: 1.0

Unmatched Word Factor Table:

Select Nodes Jytte Smith (566048) Jytte Smith (567163)

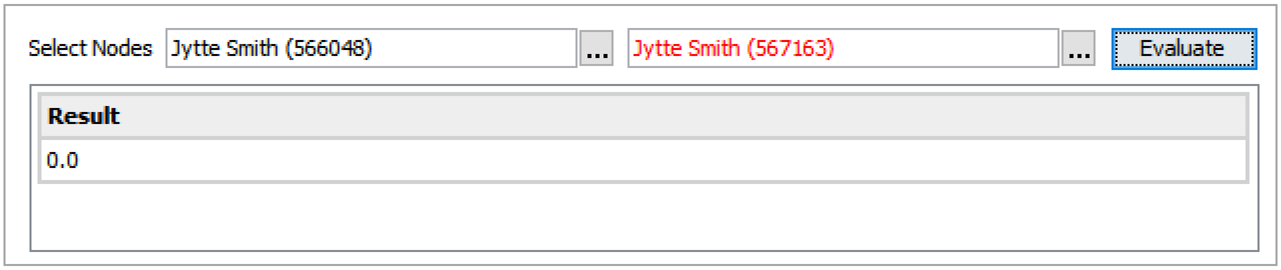
- For the optional **Word Splitter Regex**, determine the Regex based on the data being processed:

  Data such as social security numbers (SSN) or DUNS numbers that should not be split: Remove the Word Splitter Regex parameter value so no splitting is performed, and the word-strings are identical to the word-tokens.

  Data such as location name, customer names, or sentence-like constructs: Add a Word Splitter Regex to split, such as the default which splits the word-string into word-tokens based on white spaces.
- For the required **Exact Word Match Factor**, enter how greatly exact matches influence the final score.
- For the required **Alias Word Match Factor**, enter how greatly words that are paired via aliases influence the final score.
- For the required **Metaphone3 Word Match Factor**, enter how greatly pairs via Metaphone 3 influence the final score.
- For the required **Edit Distance Word Match Factor**, enter how greatly pairs via edit distance influence the final score.

- For the required **Missing Word Factor**, enter how much unpaired or missing words penalize the final result. To modify the factor for specific words, select an Unmatched Word Factor Table in the parameter below.
- For the required **Word Out of Order Factor**, enter how much words that appear out of order penalize the final result.
- For the optional **Unmatched Word Factor Table**, click the ellipsis button (...) and select a Transformation Lookup Table to assign factors to certain words and increase or decrease the significance of the unmatched word. Unmatched words that are included in this lookup table use the factor in the table instead of the Missing Word Factor from the parameter above. Refer to the **Considerations** section above.

4. To test the configuration, for the Select Nodes parameters:



Select Nodes: Jytte Smith (566048) ... Jytte Smith (567163) ...

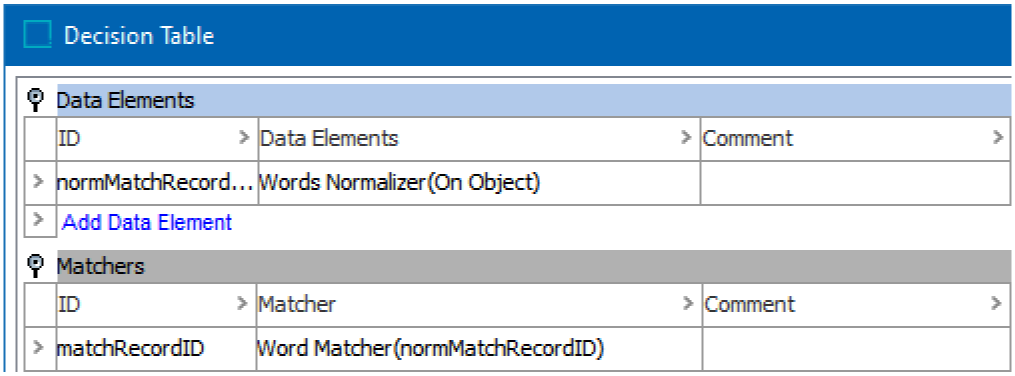
Result
0.0

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  0.0 is displayed when a value is not available in one of the selected nodes or when the words do not match. Adjust as indicated by the Evaluator results and repeat the evaluation.

  When red text is displayed, hover to review information about the record. For example, a record that has been deactivated, and so it produces no match code and thus no match score.

5. Click **OK** to save and display the configuration in the Matchers flipper.



Decision Table

Data Elements

ID	Data Elements	Comment
normMatchRecord...	Words Normalizer(On Object)	

Matchers

ID	Matcher	Comment
matchRecordID	Word Matcher(normMatchRecordID)	

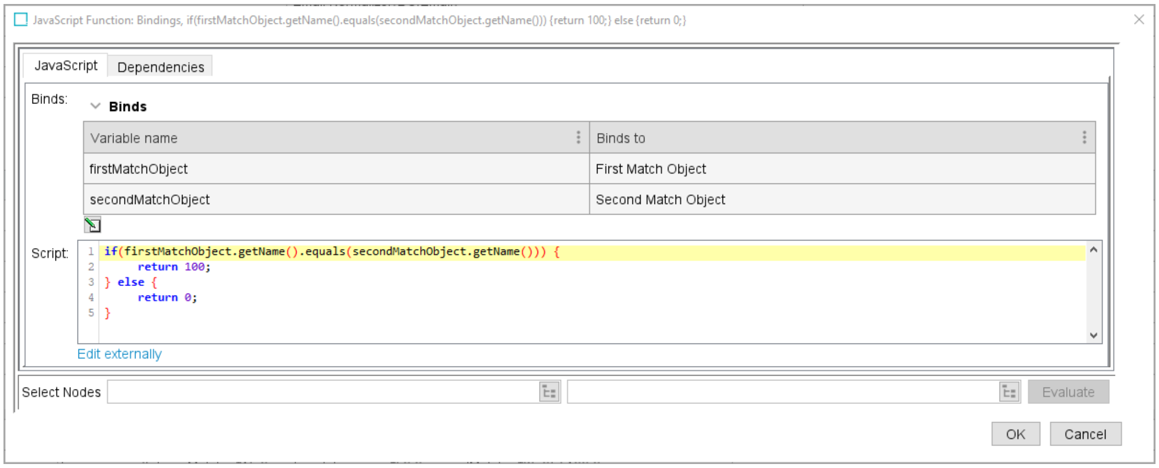
Matching Binds

JavaScript binds for matching assess elements from the Data Elements section and the Matchers section of the decision table and compare their results. For more information on Data Elements, refer to the **Match Criteria Data Elements** topic. For more information on Matchers, refer to the **Match Criteria Matchers** topic.

The same JavaScript binds are available for all business rules, using the functionality exposed in the public Java API. JavaScript in a business rule will have access to the standard Java packages. A connection into the STEP Java API can be created via binds where Java objects are bound to JavaScript variables. For more information, click the **STEP API Documentation** button on the Start Page, and refer to the **Javadoc** link under the **Extension API** section.

First Match Object and Second Match Object

The first match object and second match object binds are used to access the first and second nodes respectively. The example below compares the name of one object to that of another and returns a score of either 100 (if they are a perfect match) or 0 (if they are not a match).



JavaScript Function: Bindings, if(firstMatchObject.getName().equals(secondMatchObject.getName())) {return 100;} else {return 0;}

Variable name	Binds to
firstMatchObject	First Match Object
secondMatchObject	Second Match Object

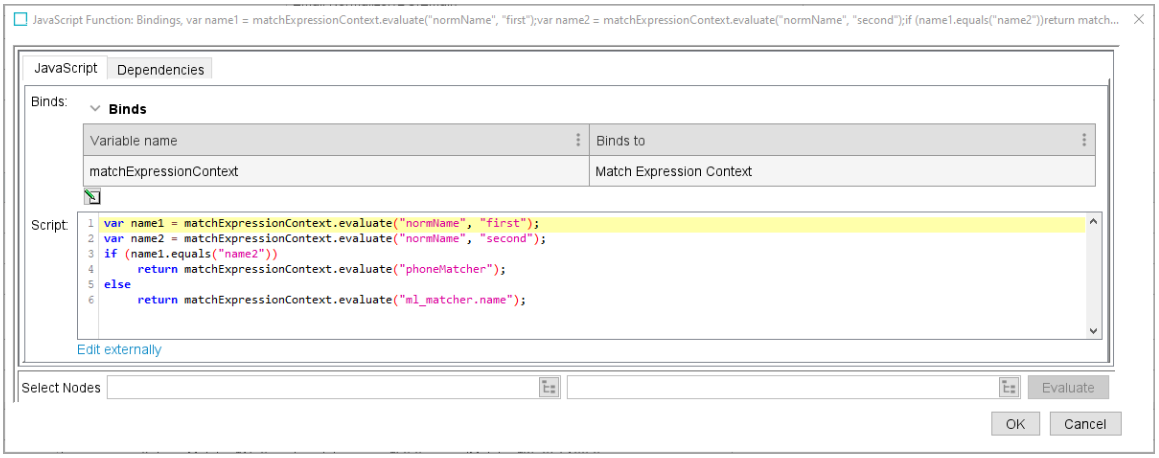
Script:

```
if(firstMatchObject.getName().equals(secondMatchObject.getName())) {
     return 100;
} else {
     return 0;
}
```

```
if(firstMatchObject.getName().equals(secondMatchObject.getName())) {
  return 100;
} else {
  return 0;
}
```

Match Expression Context

The match expression context bind is used by matchers, where two objects are in scope of the evaluation, so that the matcher can fetch data from a data element on both ‘first’ and ‘second’ objects. The example below compares the normalized legal names as text strings, and if they are a match it moves on to evaluate the phone number. If they are not a match, the name element of the Machine Learning Matcher is evaluated.



JavaScript Function: Bindings, var name1 = matchExpressionContext.evaluate("normName", "first");var name2 = matchExpressionContext.evaluate("normName", "second");if (name1.equals("name2"))return match...

Variable name	Binds to
matchExpressionContext	Match Expression Context

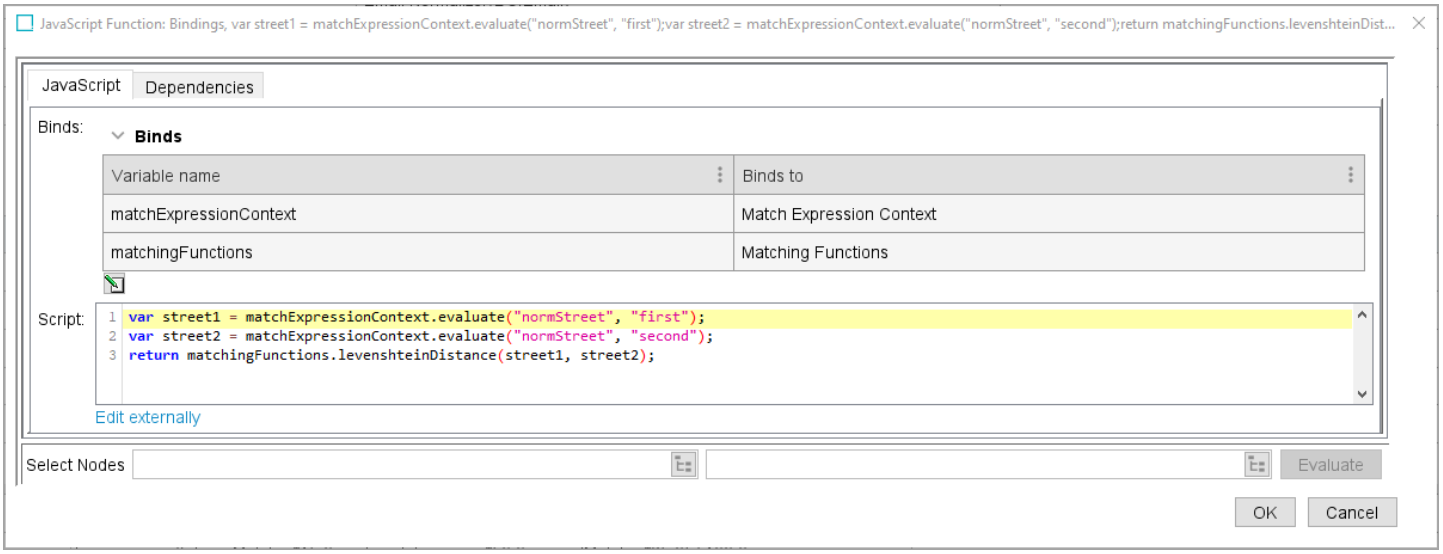
Script:

```
var name1 = matchExpressionContext.evaluate("normName", "first");
var name2 = matchExpressionContext.evaluate("normName", "second");
if (name1.equals("name2"))
     return matchExpressionContext.evaluate("phoneMatcher");
else
     return matchExpressionContext.evaluate("ml_matcher.name");
```

```
var name1 = matchExpressionContext.evaluate("normName", "first");
var name2 = matchExpressionContext.evaluate("normName", "second");
if (name1.equals("name2"))
return matchExpressionContext.evaluate("phoneMatcher");
else
return matchExpressionContext.evaluate("ml_matcher.name");
```

Matching Functions

The example below uses the built-in levenshteinDistance function to get the edit distance between normalized street values. 'Matching Functions' is bound to 'matchingFunctions.'


JavaScript Function: Bindings, var street1 = matchExpressionContext.evaluate("normStreet", "first");var street2 = matchExpressionContext.evaluate("normStreet", "second");return matchingFunctions.levenshteinDist...
JavaScript
Dependencies
Binds:
Binds
Script:
Edit externally
Select Nodes
Evaluate
OK
Cancel

Variable name	Binds to
matchExpressionContext	Match Expression Context
matchingFunctions	Matching Functions

```
var street1 = matchExpressionContext.evaluate("normStreet", "first");
var street2 = matchExpressionContext.evaluate("normStreet", "second");
return matchingFunctions.levenshteinDistance(street1, street2);
```

```
var street1 = matchExpressionContext.evaluate("normStreet", "first");
var street2 = matchExpressionContext.evaluate("normStreet", "second");
return matchingFunctions.levenshteinDistance(street1, street2);
```

Advanced Binds

There are binds available that require more advanced configuration and explanation. For more information, refer to the following topics in the **Resource Materials** documentation:

- **Match and Merge Survivorship Context Bind**
- **Pair of Attribute Values Bind**
- **Secondary Object Bind**
- **Survivorship Rule Source Objects Bind**

Match Criteria Rules

Match criteria rules dictate the final outcome of the matching evaluation. Each rule is evaluated by itself and represents a possible result of a comparison of two records. Only one rule will eventually provide the final score.

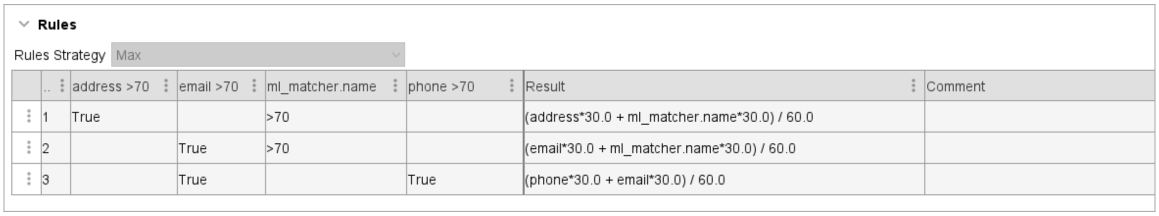
Matchers are represented as a condition column on the rules table and each row corresponds to a separate rule. The Result column calculates a score of the matched objects.

Rules Strategy and Result

The rules strategy determines which rule provides the final score based on the rule expression displayed in the Result column.

- With ‘First’ rules strategy, the first rule with no condition evaluating to false, provides the score.
- With ‘Max’ rules strategy, the rule with the highest score, with no condition evaluating to false, provides the score.

In this example, the result of the rule with the maximum score is returned via Rules Strategy: Max.



Rules

Rules Strategy: Max

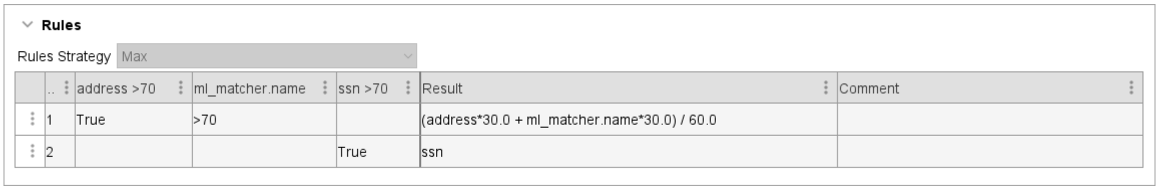
..	address >70	email >70	ml_matcher.name	phone >70	Result	Comment
1	True		>70		(address*30.0 + ml_matcher.name*30.0) / 60.0	
2		True	>70		(email*30.0 + ml_matcher.name*30.0) / 60.0	
3		True		True	(phone*30.0 + email*30.0) / 60.0	

As shown in the Result column, the individual rules combine:

- address and name
- email and name
- phone and email

This allows one of the elements to be missing and yet still return a match score of 100. For more information on match scores, refer to the Match Scores topic.

The next example uses Rules Strategy: First.



Rules

Rules Strategy: Max

..	address >70	ml_matcher.name	ssn >70	Result	Comment
1	True	>70		(address*30.0 + ml_matcher.name*30.0) / 60.0	
2			True	ssn	

The rules are:

- a combination of address and name
- social security number

If the name and address both are 'True' (that is, the threshold of 70 as set in their respective matchers is attained), the address + name rule is used, and the ssn rule is not evaluated.

If the records share a social security number (ssn), they are always matched.

Matcher Condition Columns

The conditions columns control if the rule formula Result is used to calculate a possible score. For each column, the condition threshold from the Matcher (if a condition threshold defined) is displayed next to the name of the matcher.

- With the 'Max' rules strategy, the result of all rules with true conditions are calculated, and the maximum result is reported as the match criteria score.
- With the 'First' rules strategy, the result of the top-most rule with true conditions is reported as the match criteria score.

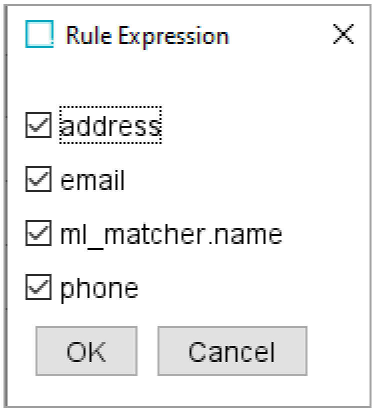
In the previous image, the '>70' value displayed in the title row of the conditions has been provided by the corresponding matcher configurations, each of which states that if it exceeds a condition threshold of '70' it will return 'True.'

In some cases, such as when using a more function-based decision table, the threshold is not established in the matcher configuration and must instead be defined in the table cell.

Configuring Match Criteria Rules

Use these options to configure decision table rules:

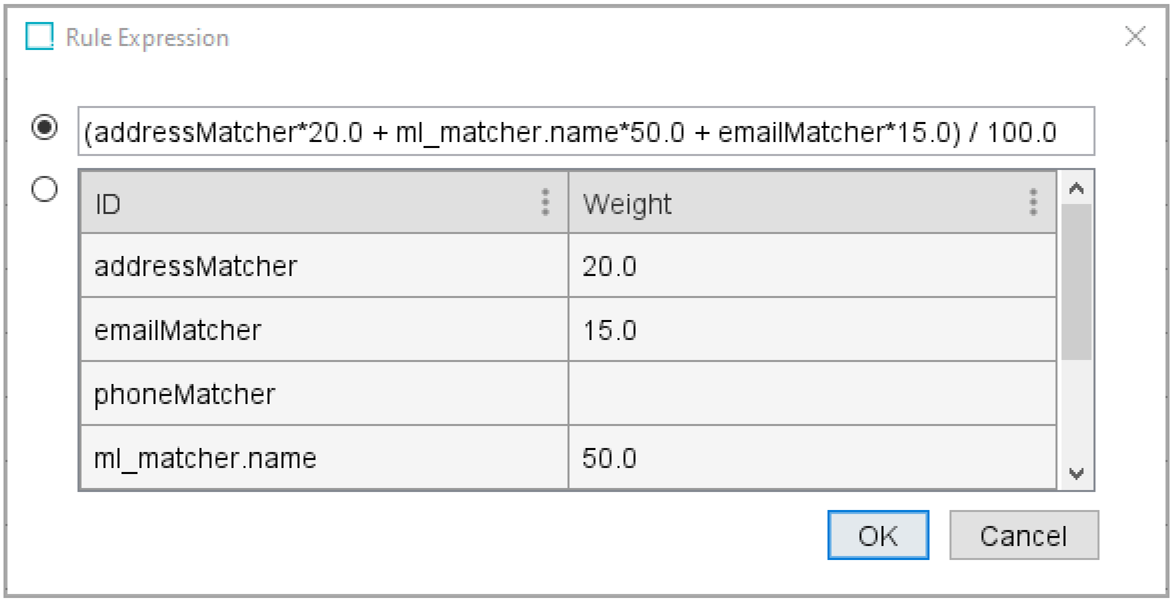
1. Click the **Edit Conditions** button to add or remove Matchers column(s) from the rules table.



Note: The Machine Learning Matcher data elements are displayed using dot notation (.) that connects the matcher ID 'ml_matcher' to the mapped data element ID, such as 'ml_matcher.name'.

2. Click the **Rules Strategy** dropdown and set the strategy:
   - 'First' runs the rules from top to bottom and bases the results on the first rule in which all conditions return 'True.'
   - 'Max' evaluates all conditions that return 'True' and combines their maximum results.
3. Click the **Add Rule** link to insert a new row in the table.
4. Click into the **Result** column and enter the expression directly into the cell or click the ellipsis button (**...**) to access the Rule Expression dialog.

   On the 'Rule Expression' dialog, click the radio button to manually define an expression or use the table to add the desired weights for the relevant conditions and generate the relevant expression. Click **OK**.

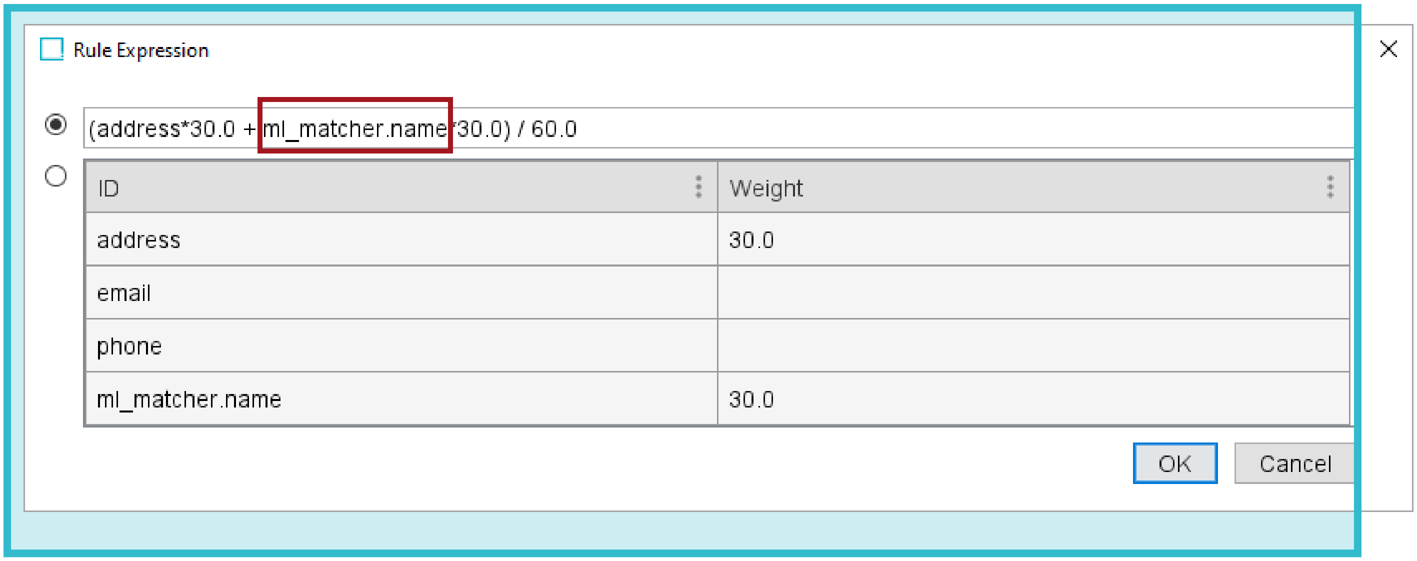


● (addressMatcher*20.0 + ml_matcher.name*50.0 + emailMatcher*15.0) / 100.0

○

ID	Weight
addressMatcher	20.0
emailMatcher	15.0
phoneMatcher	
ml_matcher.name	50.0

Note: When manually entering a condition for the Machine Learning Matcher, use dot notation (.) to access the scores. Type the ID for the matcher ('ml_matcher') followed by, for example, '.name', or '.address', as demonstrated in the below example where 'ml_matcher.name' has been manually entered into the field.

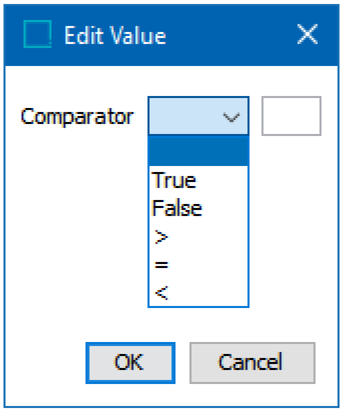


● (address*30.0 + ml_matcher.name*30.0) / 60.0

○

ID	Weight
address	30.0
email	
phone	
ml_matcher.name	30.0

5. Click into a **condition column** and manually add a comparator for the rule and condition, or click the ellipsis button (**...**) to access the 'Edit Value' dialog and select a comparator from the dropdown. If the condition threshold is defined in the matcher, select 'True', otherwise, manually enter the condition threshold.

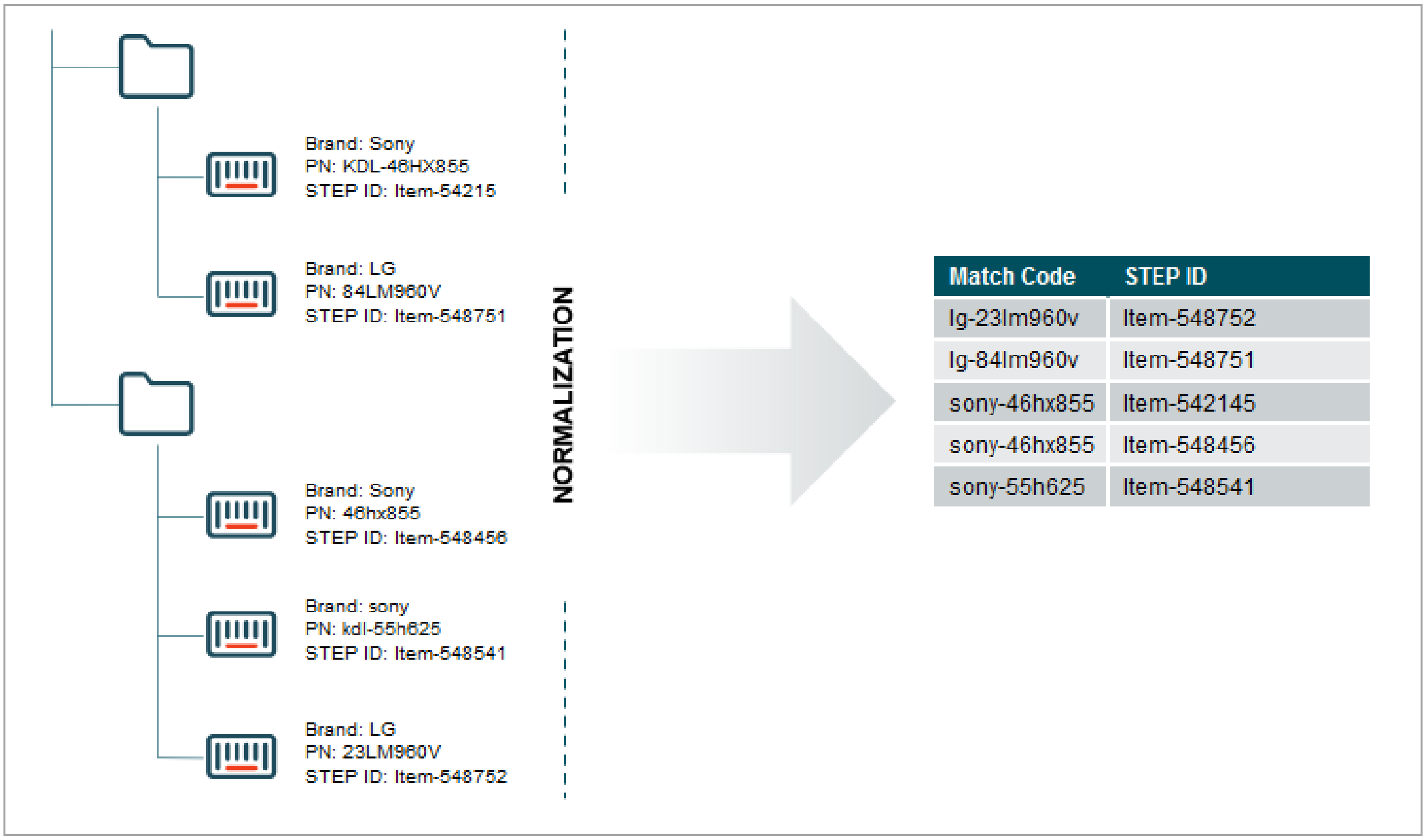


Match Codes

The matching algorithm is typically configured first, followed by the match codes.

The purpose of match criteria is to determine if the current record matches another record in the database. The purpose of match codes is to provide a fast and efficient way to find the records that are potential matches and will score above the auto merge and clerical review thresholds. Records with at least one match code in common are compared with the match score, which is defined in the **Match Scores** topic. Since the database can contain an incredible amount of data, algorithms use match codes to compare created results and process records quickly.

A match code is essentially a string (i.e., a text) that represents an object. Once generated, match codes populate a table sorted alphabetically. Rather than comparing every object with every other object in the dataset, only objects with at least one equal match code are compared.



Match Code	STEP ID
lg-23lm960v	Item-548752
lg-84lm960v	Item-548751
sony-46hx855	Item-542145
sony-46hx855	Item-548456
sony-55h625	Item-548541

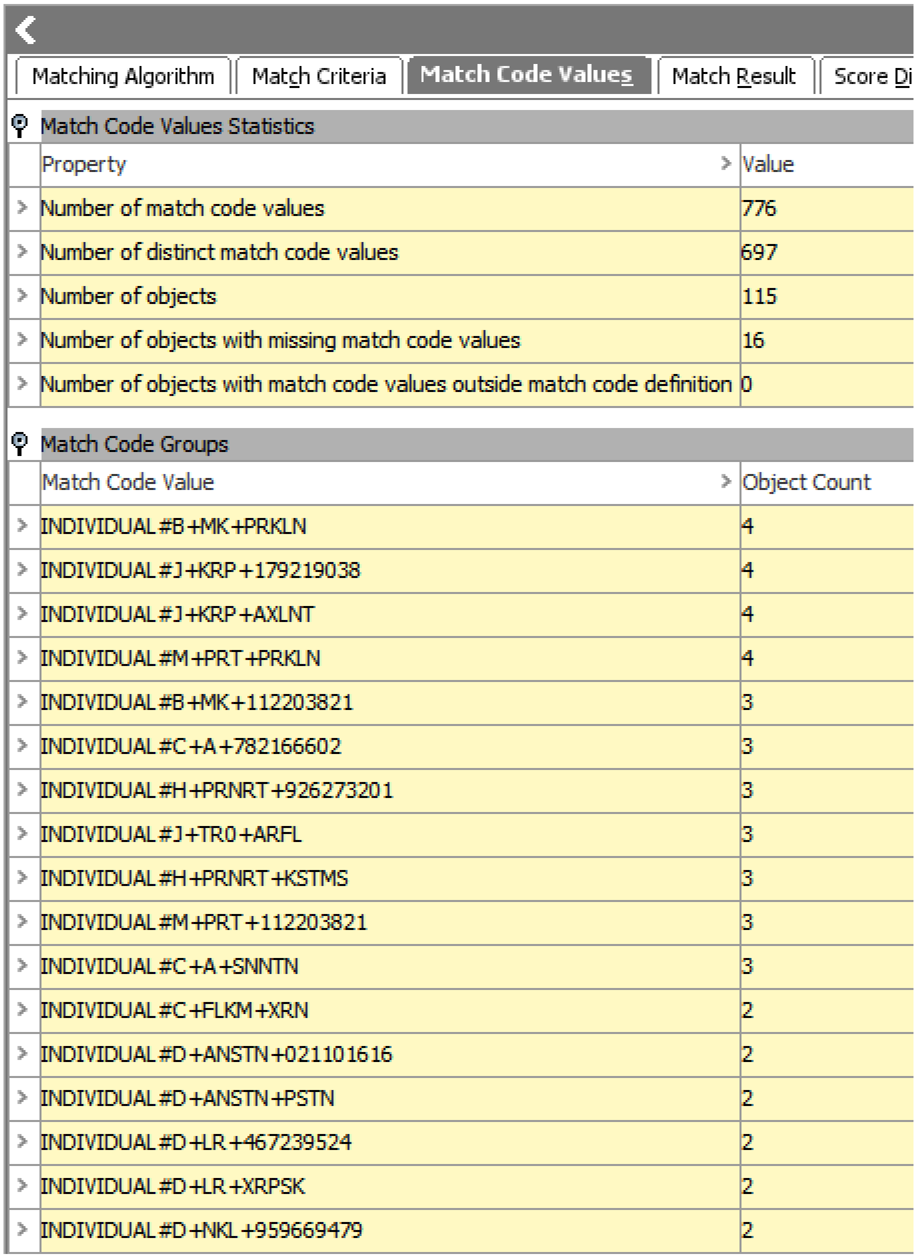
In the example above, the product with STEP ID Item-548456 is the current record. Reviewing the product in the match code table shows that one other object has an identical match code.

Typically, it is necessary to use several different match codes to ensure matching records are compared. There is a balance between determining which match codes to use and how many match codes to use. It is important that matching records share at least one match code. Non-matching records should not share match codes since running full match criteria comparisons on those records will waste system resources.

Note: A match code definition can be exported as comments and submitted to an external source control system for comparison purposes. For details, refer to the **Configuration Management** documentation.

Match Code Values

On a running system, match code values can be examined in workbench using the match code values tab on the matching algorithm. Match codes are expected to be relatively unique. A group of equal match codes is referred to as a match code group, which should be small. No match code group size should be larger than 100 and generally, most objects (95 percent) should be in a match code group with a size of 10 or smaller.



Match Code Values Statistics

Property	Value
Number of match code values	776
Number of distinct match code values	697
Number of objects	115
Number of objects with missing match code values	16
Number of objects with match code values outside match code definition	0

Match Code Groups

Match Code Value	Object Count
INDIVIDUAL#B+MK+PRKLN	4
INDIVIDUAL#J+KRP+179219038	4
INDIVIDUAL#J+KRP+AXLNT	4
INDIVIDUAL#M+PRT+PRKLN	4
INDIVIDUAL#B+MK+112203821	3
INDIVIDUAL#C+A+782166602	3
INDIVIDUAL#H+PRNRT+926273201	3
INDIVIDUAL#J+TR0+ARFL	3
INDIVIDUAL#H+PRNRT+KSTMS	3
INDIVIDUAL#M+PRT+112203821	3
INDIVIDUAL#C+A+SNNTN	3
INDIVIDUAL#C+FLKM+XRN	2
INDIVIDUAL#D+ANSTN+021101616	2
INDIVIDUAL#D+ANSTN+PSTN	2
INDIVIDUAL#D+LR+467239524	2
INDIVIDUAL#D+LR+XRPSK	2
INDIVIDUAL#D+NKL+959669479	2

Use the following points to closely examine the data before configuring a match code:

- The data profiling tool provides much valuable information. If you are planning to use a specific attribute in the match code, verify the degree to which the attribute is populated. If values are missing on a lot of objects, the attribute is likely not a good candidate or at least should not be used alone. Objects with empty values for a match codes are not compared based on that match code.
- If an attribute is sufficiently unique, like an EAN number, the match code can be based on just that single piece of data.

- If an attribute is less unique, like a name, it should be used in combination with other values in order to generate good match codes. An example is the Person Name and Address match code generator which is available for customer data.
- When working with match codes combining several pieces of data, always put the most significant data first. For example, when deduplicating address objects, put the ZIP code before street and street number, since ZIP codes are geographic, standardized, and mutually exclusive, which most effectively separates addresses into discrete objects.
- Normalize the data used in match codes. For example, if a manufacturer name is often abbreviated, the match code definition should ensure the name is represented the same way in the match codes, regardless if the source object is abbreviated or not.
- Several match codes can be generated per source object, even by the same match code generator. Use STEP functions to resolve to a list of multiple match codes, and in JavaScript return an array. In these cases, each element is a separate match code. Consider, for example, a customer with several email addresses. Each email address should result in a separate email match code.
- Sometimes an otherwise great identifier has exception cases that should be filtered out. Phone numbers are often very good match code candidates, but multiple contacts at a customer business may have provided the reception main number, resulting in a single match code group with hundreds of records. In this case, a match code filter can be applied to the phone match code to remove this exceptional case. For more information, refer to the **Match Criteria Match Code Filter** topic.

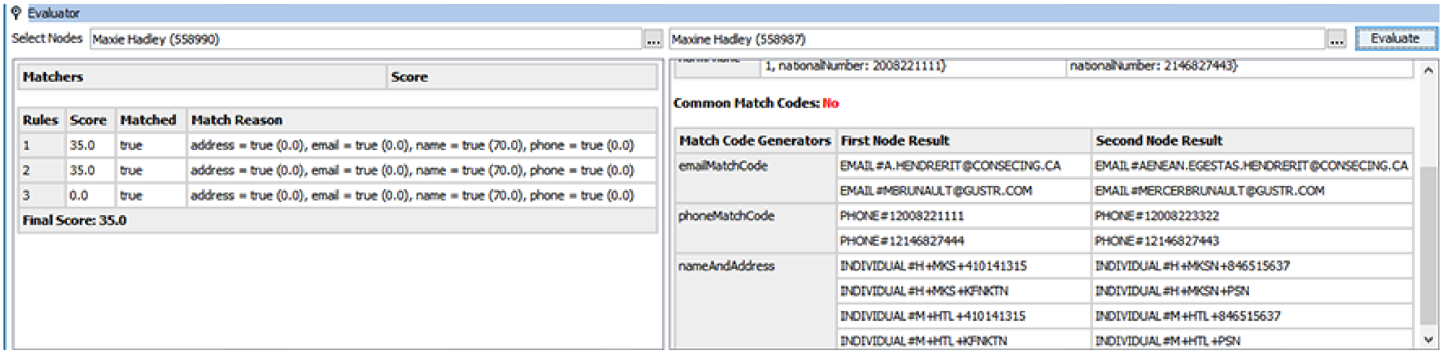
Creating Match Code Values

On the matching algorithm, the methods used to create match code values are available as defined below. For information about each, review the following topics:

- **Match Criteria Match Code Generators** on the Match Criteria tab
- **Match Criteria Match Code Filter** on the Match Criteria tab
- **Configuring a Legacy External Match Code** on a separate Match Code object

Evaluator

The matching algorithm evaluator tool verifies results and can help identify unexpected results. In the evaluator, select two objects that you want to compare and click the **Evaluate** button. Detailed information is displayed including how the result was obtained. Additionally, the evaluators on individual sub components of the algorithm can be used to expose more details.



Select Nodes: Maxie Hadley (558990) | Maxine Hadley (558987)

Matchers	Score

Rules	Score	Matched	Match Reason
1	35.0	true	address = true (0.0), email = true (0.0), name = true (70.0), phone = true (0.0)
2	35.0	true	address = true (0.0), email = true (0.0), name = true (70.0), phone = true (0.0)
3	0.0	true	address = true (0.0), email = true (0.0), name = true (70.0), phone = true (0.0)

Final Score: 35.0

	1, nationalNumber: 2008221111}	nationalNumber: 2146827443}

Common Match Codes: No

Match Code Generators	First Node Result	Second Node Result
emailMatchCode	EMAIL #A.HENDRERIT@CONSECING.CA	EMAIL #AENEAN.EGESTAS.HENDRERIT@CONSECING.CA
	EMAIL #MBRUNAULT@GUSTR.COM	EMAIL #MERCERBRUNAULT@GUSTR.COM
phoneMatchCode	PHONE#12008221111	PHONE#12008223322
	PHONE#12146827444	PHONE#12146827443
nameAndAddress	INDIVIDUAL #H+MKS+410141315	INDIVIDUAL #H+MKSN+846515637
	INDIVIDUAL #H+MKS+KFNKTN	INDIVIDUAL #H+MKSN+PSN
	INDIVIDUAL #M+HTL+410141315	INDIVIDUAL #M+HTL+846515637
	INDIVIDUAL #M+HTL+KFNKTN	INDIVIDUAL #M+HTL+PSN

Using Data Profiling to Select Match Codes

Designing match criteria for a deduplication strategy requires an intimate understanding of the data and STEP Data Profiles can be of great assistance. Data profiles show the extent to which relevant attributes are populated and highlight the most frequent and rare values and patterns. For more information, refer to the **Data Profiling** documentation.

The example below shows how data profiling is an indispensable tool in determining the right matching algorithm configuration.

Prerequisites

Configure the Matching component model as defined in **Configuring Matching Component Model** topic.

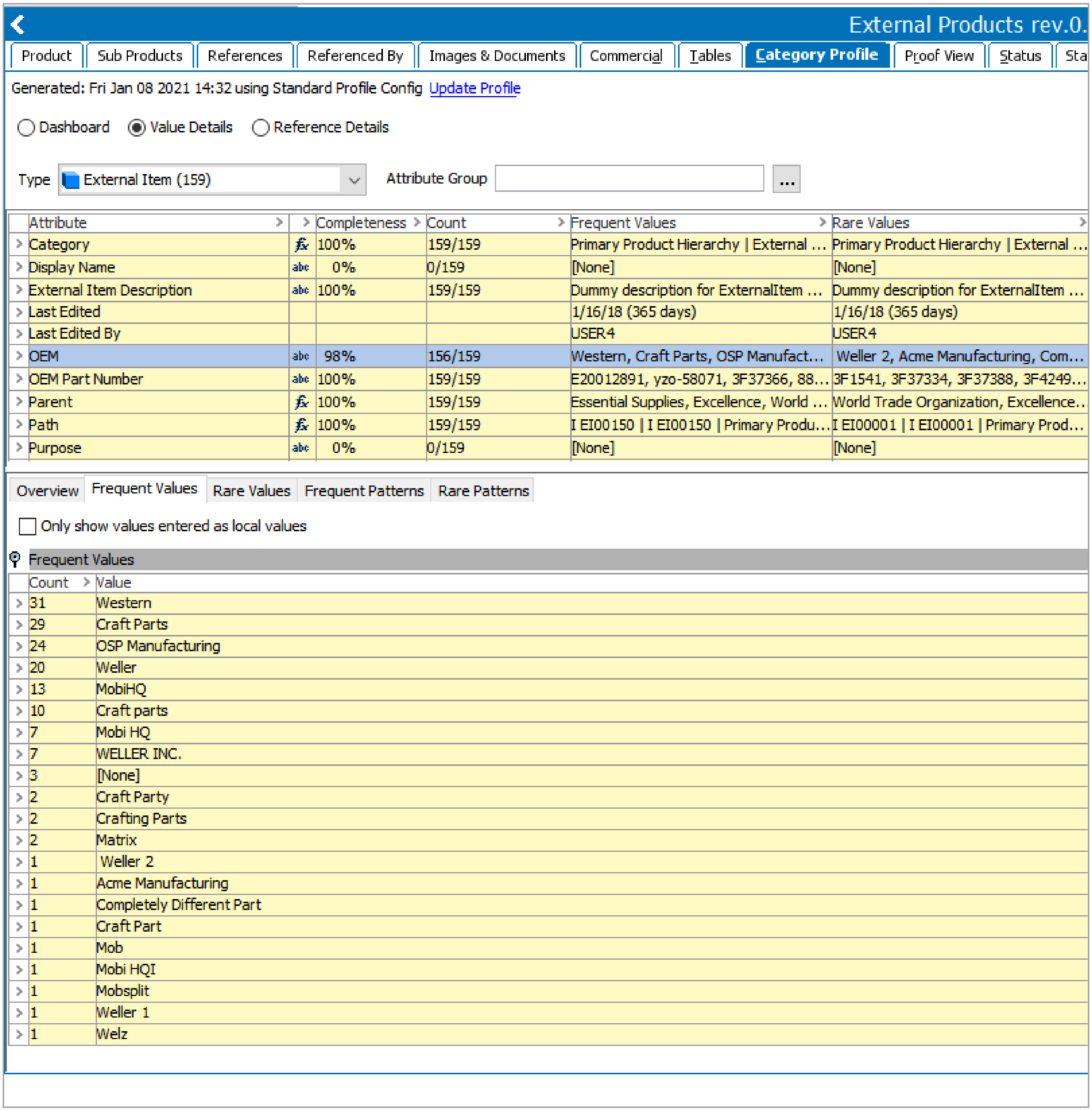
Data Profile Analysis Example

In this example, the data profile (shown in the product's Category Profile tab in the image below) is used to determine match codes, data elements and normalization, matchers, and rules. OEM and OEM Part Number are used to compare the products.

Observations

A profile is generated from the 'External Products' node and the following observations are made:

- The Completeness column indicates there are missing values for OEM. Missing values result in missing match codes and could lead to the objects not being compared if all match codes depend on OEM.
- The Frequent Values tab for the OEM attribute row shows that the OEM values include obvious duplicates like 'Craft Parts' / 'Craft parts' and 'Weller' / 'WELLER INC'. Normalization is required for the OEM data element.



External Products rev.0.
Product | Sub Products | References | Referenced By | Images & Documents | Commercial | Tables | Category Profile | Proof View | Status | Sta

Generated: Fri Jan 08 2021 14:32 using Standard Profile Config Update Profile

○ Dashboard ● Value Details ○ Reference Details

Type: External Item (159)

Attribute Group:

Attribute		Completeness	Count	Frequent Values	Rare Values
Category	fx	100%	159/159	Primary Product Hierarchy | External ...	Primary Product Hierarchy | External ...
Display Name	abc	0%	0/159	[None]	[None]
External Item Description	abc	100%	159/159	Dummy description for ExternalItem ...	Dummy description for ExternalItem ...
Last Edited				1/16/18 (365 days)	1/16/18 (365 days)
Last Edited By				USER4	USER4
OEM	abc	98%	156/159	Western, Craft Parts, OSP Manufact...	Weller 2, Acme Manufacturing, Com...
OEM Part Number	abc	100%	159/159	E20012891, yzo-58071, 3F37366, 88...	3F1541, 3F37334, 3F37388, 3F4249...
Parent	fx	100%	159/159	Essential Supplies, Excellence, World ...	World Trade Organization, Excellence...
Path	fx	100%	159/159	I EI00150 | I EI00150 | Primary Produ...	I EI00001 | I EI00001 | Primary Prod...
Purpose	abc	0%	0/159	[None]	[None]

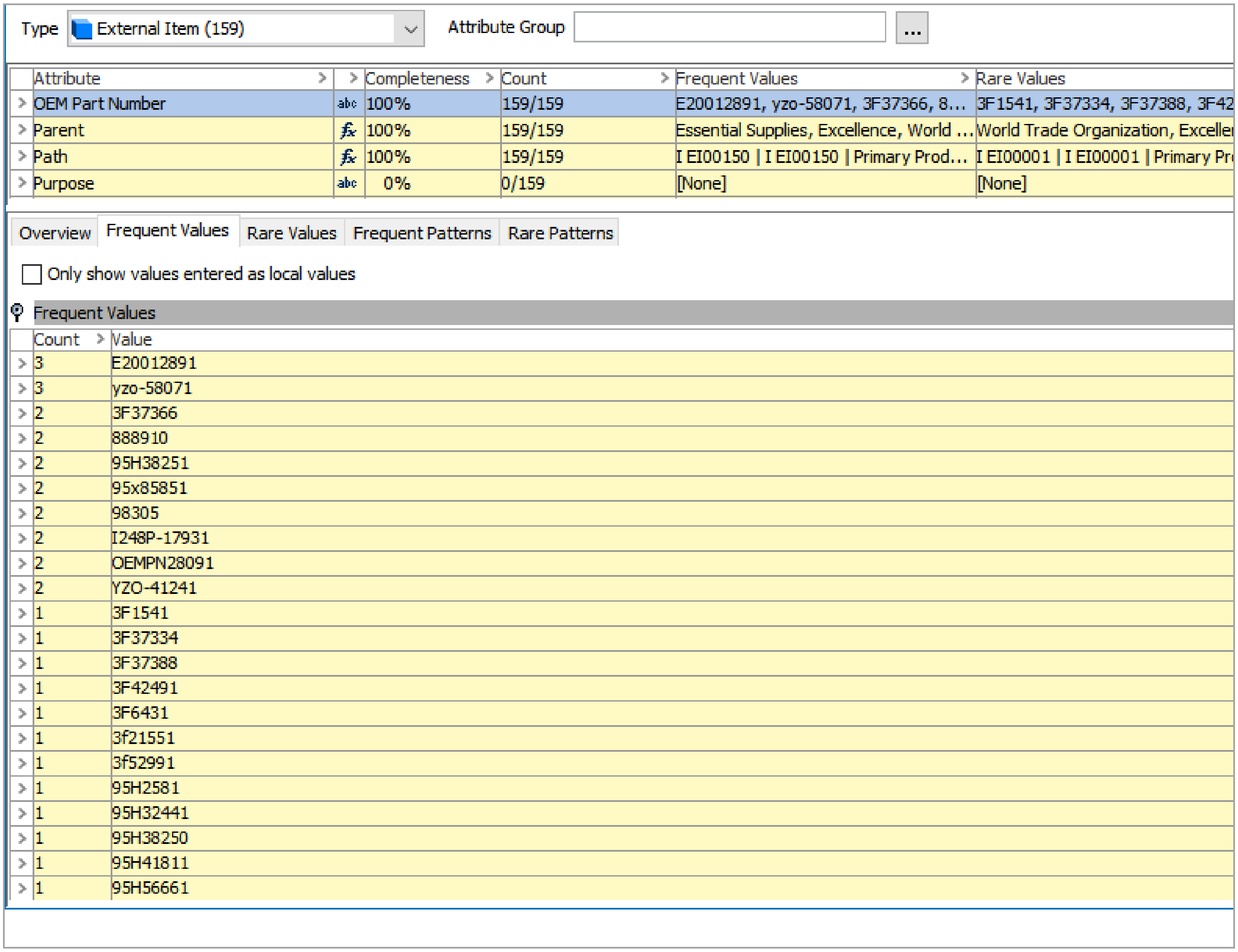
Overview | Frequent Values | Rare Values | Frequent Patterns | Rare Patterns

□ Only show values entered as local values

Frequent Values

Count	Value
31	Western
29	Craft Parts
24	OSP Manufacturing
20	Weller
13	MobiHQ
10	Craft parts
7	Mobi HQ
7	WELLER INC.
3	[None]
2	Craft Party
2	Crafting Parts
2	Matrix
1	Weller 2
1	Acme Manufacturing
1	Completely Different Part
1	Craft Part
1	Mob
1	Mobi HQI
1	Mobsplit
1	Weller 1
1	Welz

- The OEM Part Number attribute row Count column (shown below) indicates there are 159 values. Since there are more than 100 distinct values, the workbench data profile default settings do not provide exact statistics. Although the Web UI would show the exact statistics, in this case it is not necessary. The displayed values show that both uppercase and lowercase letters are used, and that punctuation is used in some values but not in others. Normalization is required to create match codes for OEM Part Number.



Type: External Item (159)

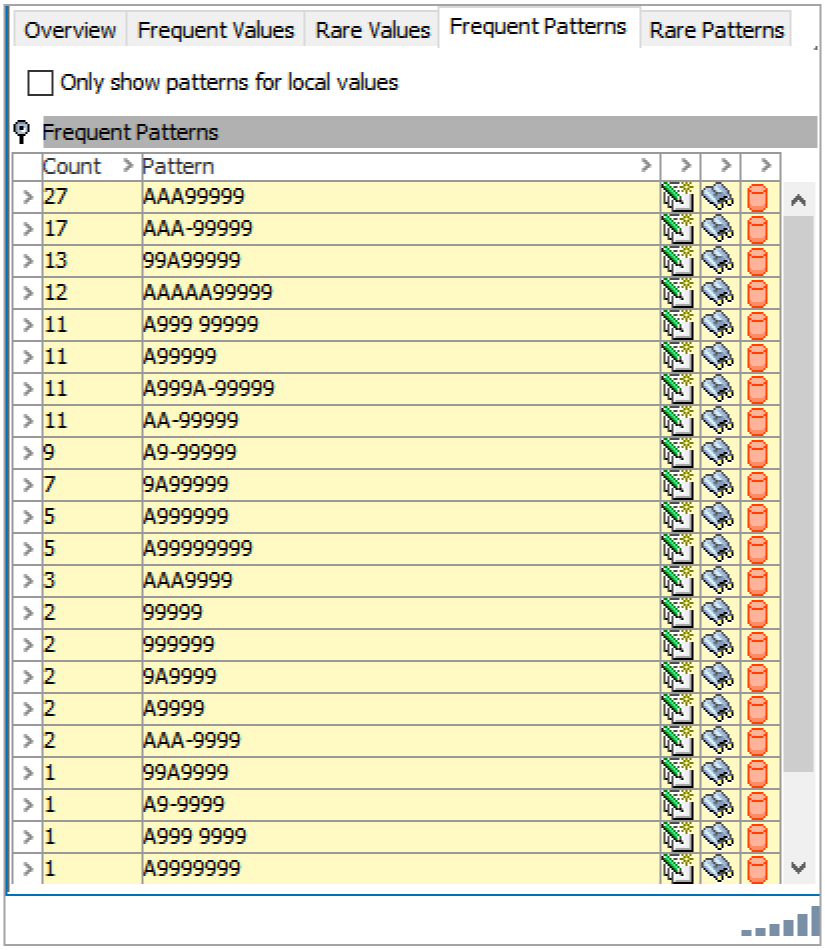
Attribute		Completeness	Count	Frequent Values	Rare Values
OEM Part Number	abc	100%	159/159	E20012891, yzo-58071, 3F37366, 8…	3F1541, 3F37334, 3F37388, 3F42
Parent	fx	100%	159/159	Essential Supplies, Excellence, World …	World Trade Organization, Exceller
Path	fx	100%	159/159	I EI00150 | I EI00150 | Primary Prod…	I EI00001 | I EI00001 | Primary Pr
Purpose	abc	0%	0/159	[None]	[None]

□ Only show values entered as local values

Frequent Values

Count	Value
3	E20012891
3	yzo-58071
2	3F37366
2	888910
2	95H38251
2	95x85851
2	98305
2	I248P-17931
2	OEMPN28091
2	YZO-41241
1	3F1541
1	3F37334
1	3F37388
1	3F42491
1	3F6431
1	3f21551
1	3f52991
1	95H2581
1	95H32441
1	95H38250
1	95H41811
1	95H56661

- The Frequent Patterns tab shows that there are no clear, distinct patterns in the values.



□ Only show patterns for local values

Frequent Patterns

Count	Pattern
27	AAA99999
17	AAA-99999
13	99A99999
12	AAAAA99999
11	A999 99999
11	A99999
11	A999A-99999
11	AA-99999
9	A9-99999
7	9A99999
5	A999999
5	A9999999
3	AAA9999
2	99999
2	999999
2	9A9999
2	A9999
2	AAA-9999
1	99A9999
1	A9-9999
1	A999 9999
1	A999999

Match Code Strategy Options

The following describes potential match code strategies and the faults or recommendations of each:

1. **Two match codes - one for OEM and one for OEM part number:** While two match codes could be used, this is not the best strategy because the number of different OEM values is quite low, especially if they are normalized. Also, 31 values of 'Western' would lead to a very large Match Code Group. Not recommended since using the OEM alone as Match Code would lead to significant performance problems.

2. **One match code combining OEM Part Number and OEM:** Even using a calculated attribute of the values to include in the larger data profile, since some objects will not get a match code because the OEM is not 100 percent complete in our small data sample. Not recommended since OEM cannot stand by itself.

3. **One match code for OEM Part Number and other attribute values:** Since there is a significant spread in OEM Part Number values, generating match codes based solely on these values could work. However, a larger dataset would need to be profiled using the Web UI's exact uniqueness. This could result in larger Match Code Groups, but, based on this subset of products, the largest group size would be 3, which is acceptable. **Recommended** based on the larger data profile, if OEM Part Number was combined with other attribute values in the final match codes to ensure small match code groups.

Matcher Strategy Options

The following describes potential matcher strategies and the faults or recommendations of each:

- **A matcher on OEM + OEM Part Number:** Not recommended since the matcher must handle the missing OEM values.
- **Separate matchers for OEM and OEM Part Number:** Use Rules to combine the scores, ensuring that a match where one is missing the OEM would go to clerical review. This would need to be clarified with the business.

Match Criteria Match Code Generators

Match codes (as defined in the **Match Codes** topic) are created by match code generators. There are a number of built-in match code generators for party data. For other cases, use the Business Function Match Code Generator, which maps to a business function that returns a list of text strings that will each become a separate match code.

Note: Match code generators are used for matching algorithms that have been created with the Embed Match Code checkbox enabled.

Match Codes provide an efficient method for a matching algorithm to identify potential duplicate records and are created as follows:

- Embedded match codes are created via Match Code Generators when the matching algorithms is created and the Embed Match Code checkbox option is checked, as defined in this topic.
- Stand-alone match codes are a legacy option, necessary when the matching algorithm is created and the Embed Match Code checkbox option is not checked, as defined in the Configuring a Legacy External Match Code topic.

Multiple match code generators can exist on the match criteria at the same time. Only match code generators set as 'Active' are used in the matching algorithm.

The types of match code generators include:

- Preconfigured generators to create match codes for typical party data information.
- A general-purpose generator that maps to a business function which returns a list of text strings that become separate match codes.

Match Code Generator	Match Code Generator Type	Object Type Allowed
Business Function	General Purpose	Entities Products
Address	Preconfigured	Entities
Email	Preconfigured	Entities

Match Code Generator	Match Code Generator Type	Object Type Allowed
Natural Key	Preconfigured	Entities
Organization Name and Address	Preconfigured	Entities
Person Name and Address	Preconfigured	Entities
Phone	Preconfigured	Entities

Match Code Generator: Address

The Address match code generator is used to ensure comparison of addresses when an organization entity spans several address numbers on the same street, so it does not include street numbers.

Recommendations

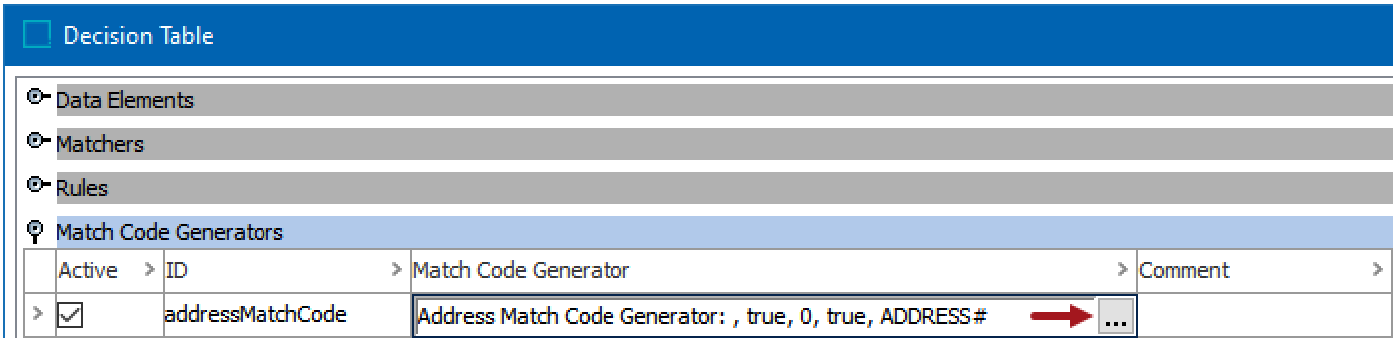
It is recommended to combine an address with a name for better matching.

- For individuals and B2C customers, address as a match code alone compares spouses and anyone living on the same street. Instead, use the **Match Code Generator: Person Name and Address** so the match code combines the address with the person name.
- For organizations, match codes including addresses are often used together with match rules that score addresses and score person or organization names. Instead, use the **Match Code Generator: Organization Name and Address** so the match code combines the address with the organization name.

Configuring an Address Match Code Generator

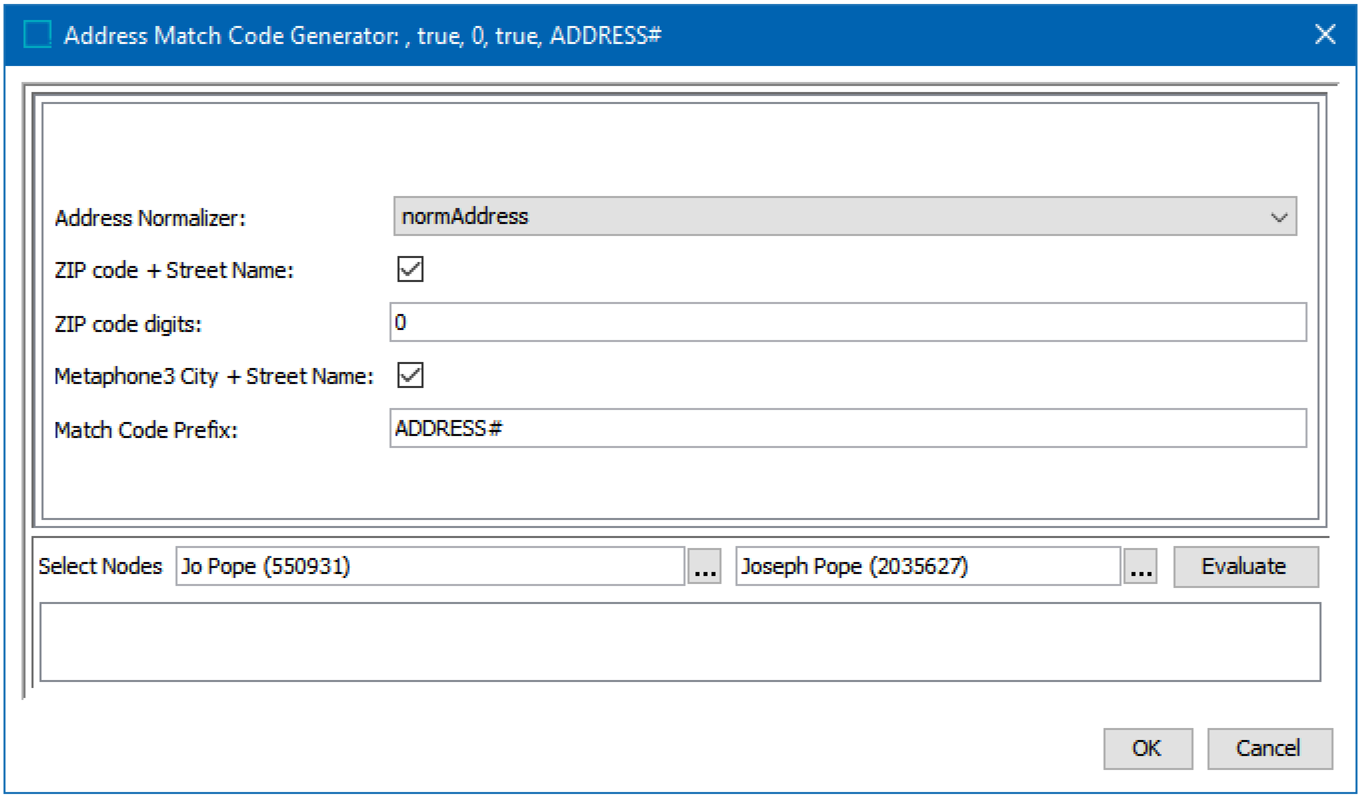
After adding the match code generator in the Match Code Generators flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Match Code Generator column and click the ellipsis button (...) to access the configuration dialog.



Active	ID	Match Code Generator	Comment
☑	addressMatchCode	Address Match Code Generator: , true, 0, true, ADDRESS#	

2. On the Address Match Code Generator dialog:



Address Match Code Generator: , true, 0, true, ADDRESS#

Address Normalizer: normAddress

ZIP code + Street Name: ☑

ZIP code digits: 0

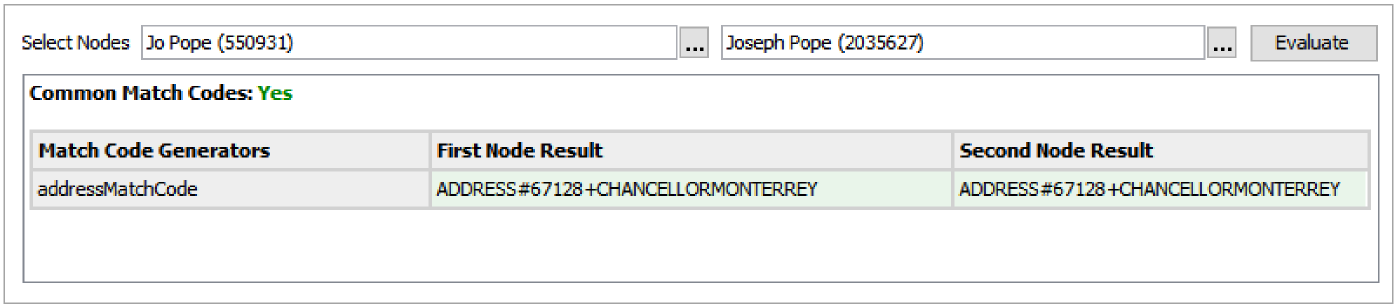
Metaphone3 City + Street Name: ☑

Match Code Prefix: ADDRESS#

Select Nodes Jo Pope (550931) ... Joseph Pope (2035627) ...

- For the required **Address Normalizer**, use the dropdown to select the associated normalizer or enter a case-sensitive ID for the normalizer. Refer to the **Data Element: Address Normalizer** topic.
- For the **ZIP code + Street Name**, when checked, the ZIP code digits defined in the following parameter are appended to the street name for the match code.
- For the optional **ZIP code digits**, enter the number of ZIP code digits, starting from the beginning, to include in the match code. When this entry is 0, the entire zip code is used in the match code.
- For the **Metaphone3 City + Street Name**, when checked, the Metaphone 3 algorithm (which expands on Soundex) compares names based on their pronunciation. It works well on English words, non-English words familiar to Americans, first names, and family names commonly found in the United States. For more information on Metaphone 3, search the web.
- For the optional **Match Code Prefix**, enter a prefix to identify the source of the match code.

3. To test the configuration:



Common Match Codes: Yes

Match Code Generators	First Node Result	Second Node Result
addressMatchCode	ADDRESS#67128+CHANCELLORMONTERREY	ADDRESS#67128+CHANCELLORMONTERREY

- Click the ellipsis button (...) for each **Select Nodes** parameter and select two objects for comparison.

- Click the **Evaluate** button.

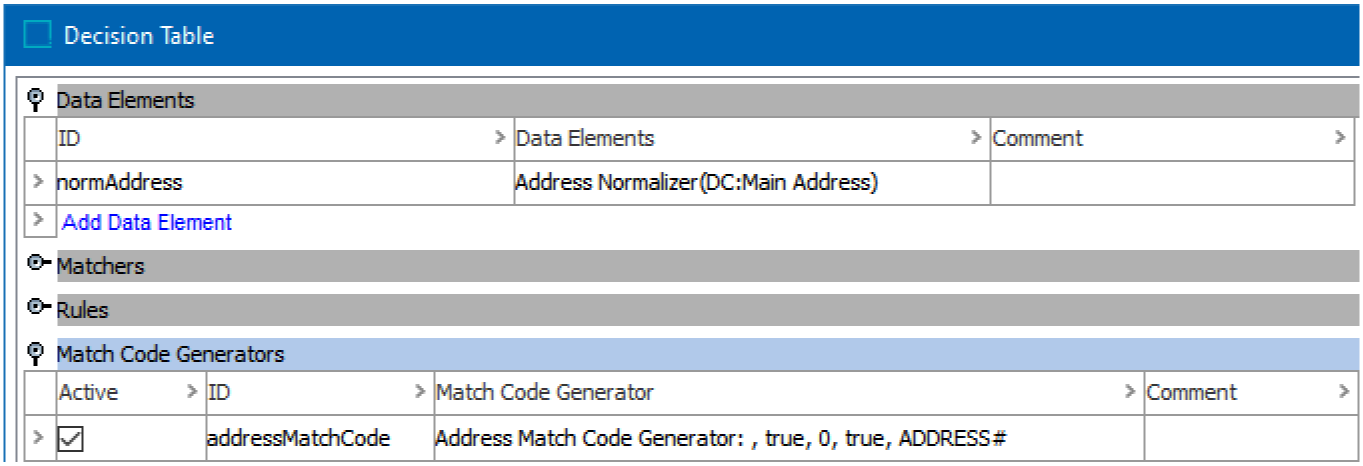
  The Common Match Codes parameter displays 'Yes' when the same match code is generated for the selected nodes. When 'No' is displayed, either one or both nodes have no value, or the generated match codes are not the same.

  When red text is displayed, hover to review information about the record. For example, a record that has been deactivated, and so it produces no match code and thus no match score.

  Adjust as indicated by the Evaluator results and repeat the evaluation if necessary.

4. Click **OK** to save and display the configuration in the Match Code Generators flipper.

5. Ensure the **Active** checkbox is checked to create match codes with this generator.



ID	Data Elements	Comment
normAddress	Address Normalizer(DC:Main Address)	

Active	ID	Match Code Generator	Comment
☑	addressMatchCode	Address Match Code Generator: , true, 0, true, ADDRESS#	

Match Code Generator: Business Function

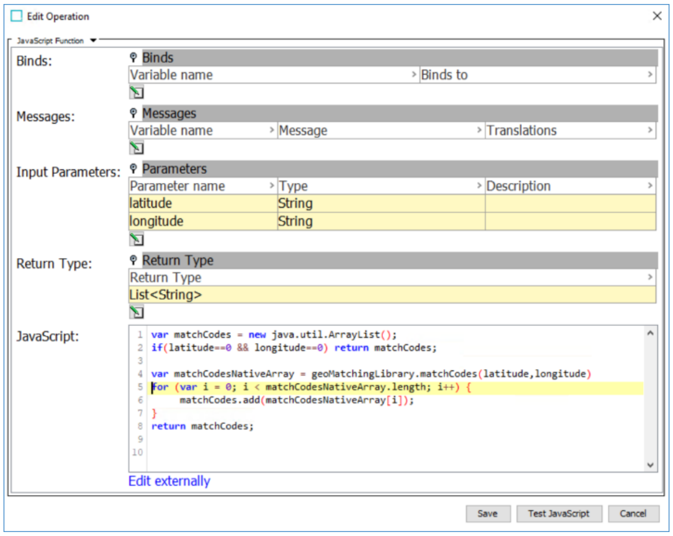
Using a Business Function match code generator provides the ultimate flexibility to ensure that two records are compared according to specific needs.

While a Business Function Match Code Generator may take any input, it must output List<String>.

Matching organization golden records based on main address location distance

In the following example, a large organization owns multiple street addresses and has access from several different roads. The Business Function Match Code Generator is used to ensure organizations are compared when their main address is less than 500 meters apart.

- The core JavaScript functions are provided by the GeoMatchingLibrary, a JavaScript library available from CMDM Enablement or your Stibo Systems account manager or partner manager
- The business function returns a number of match codes, and therefore has the return type List<String> and the input parameters are the latitude and longitude of the main address.



Input Parameters:

Parameters

Parameter name	Type	Description
latitude	String	
longitude	String	

Return Type:

Return Type
List<String>

JavaScript:

```
var matchCodes = new java.util.ArrayList();
if(latitude==0 && longitude==0) return matchCodes;

var matchCodesNativeArray = geoMatchingLibrary.matchCodes(latitude,longitude)
for (var i = 0; i < matchCodesNativeArray.length; i++) {
     matchCodes.add(matchCodesNativeArray[i]);
}
return matchCodes;
```

Edit externally

Configuring a Business Function Match Code Generator

1. Obtain latitude and longitude from the Main Address. During import, the main address is enriched by Loqate as part of the Standardization Business Action, which among other features adds latitude and longitude coordinates for the Main Address data container. This standardization is configured on the inbound endpoint.

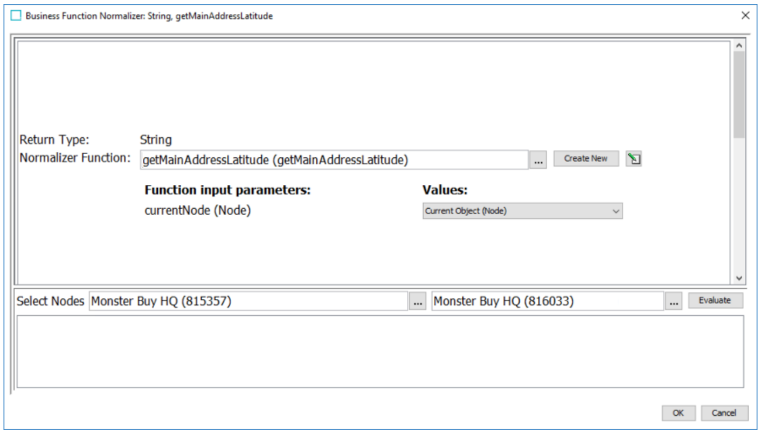
   For more information, refer to the following topics:

- **Creating an Inbound Integration Endpoint** in the **Data Exchange** documentation.
- **Web Service Endpoint - Match and Merge** in the **Data Exchange** documentation.
- **Loqate Integration** in the **Data Integration** documentation.

2. Configure Data Elements in the Match Criteria, using Business Function Normalizers to get the latitude and longitude from the Main Address.

   For more information, refer to the **Data Element: Business Function Normalizer** topic.

3. Configure the Data Elements as shown below:



Business Function Normalizer: String, getMainAddressLatitude

Return Type: String

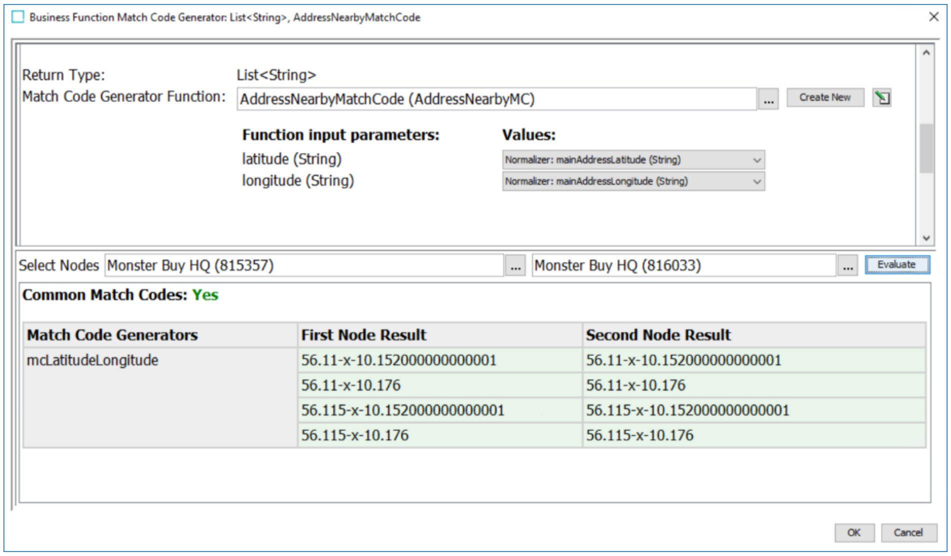
Normalizer Function: getMainAddressLatitude (getMainAddressLatitude)

Function input parameters: currentNode (Node)

Values: Current Object (Node)

Select Nodes Monster Buy HQ (815357) Monster Buy HQ (816033)

4. Create the Business Function Match Code Generator and select the Data Elements as input parameters.



Business Function Match Code Generator: List<String>, AddressNearbyMatchCode

Return Type: List<String>

Match Code Generator Function: AddressNearbyMatchCode (AddressNearbyMC)

Function input parameters: / **Values:**

latitude (String) Normalizer: mainAddressLatitude (String)

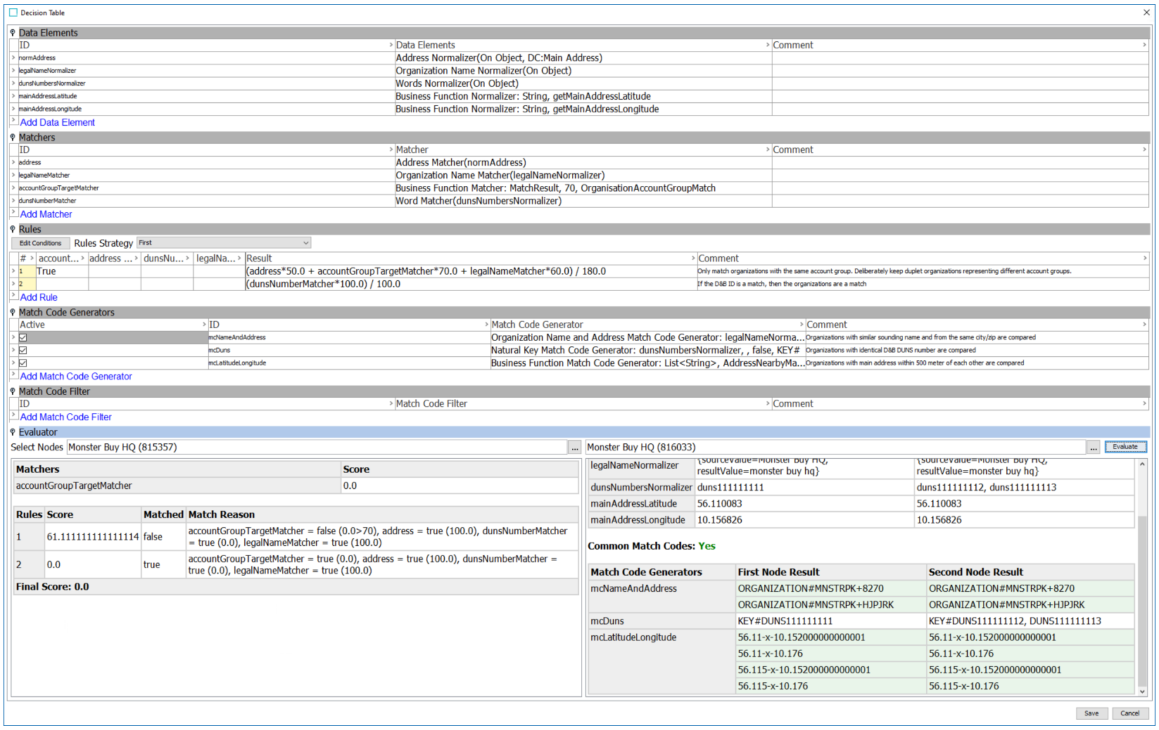
longitude (String) Normalizer: mainAddressLongitude (String)

Select Nodes Monster Buy HQ (815357) Monster Buy HQ (816033)

Common Match Codes: Yes

Match Code Generators	First Node Result	Second Node Result
mcLatitudeLongitude	56.11-x-10.152000000000001	56.11-x-10.152000000000001
	56.11-x-10.176	56.11-x-10.176
	56.115-x-10.152000000000001	56.115-x-10.152000000000001
	56.115-x-10.176	56.115-x-10.176

The image below shows the configuration for this example organization matching. The address distance is only used for generating match codes to ensure comparison. It does not affect the final match scores, as no Matcher evaluates the distance, and no Rules apply the distance between these main addresses.



Data Elements

ID	Data Elements	Comment
normAddress	Address Normalizer(On Object, DC:Main Address)	
legalNameNormalizer	Organization Name Normalizer(On Object)	
dunsNumbersNormalizer	Words Normalizer(On Object)	
mainAddressLatitude	Business Function Normalizer: String, getMainAddressLatitude	
mainAddressLongitude	Business Function Normalizer: String, getMainAddressLongitude	

Matchers

ID	Matcher	Comment
address	Address Matcher(normAddress)	
legalNameMatcher	Organization Name Matcher(legalNameNormalizer)	
accountGroupTargetMatcher	Business Function Matcher: MatchResult, 70, OrganisationAccountGroupMatch	
dunsNumberMatcher	Word Matcher(dunsNumbersNormalizer)	

Rules

Rules Strategy First

#	account...	address ...	dunsNu...	legalNa...	Result	Comment
1	True				(address*50.0 + accountGroupTargetMatcher*70.0 + legalNameMatcher*60.0) / 180.0	Only match organizations with the same account group. Deliberately keep duplet organizations representing different account groups.
2					(dunsNumberMatcher*100.0) / 100.0	If the D&B ID is a match, then the organizations are a match

Match Code Generators

Active	ID	Match Code Generator	Comment
☑	mcNameAndAddress	Organization Name and Address Match Code Generator: legalNameNorma...	Organizations with similar sounding name and from the same city/zip are compared
☑	mcDuns	Natural Key Match Code Generator: dunsNumbersNormalizer, , false, KEY#	Organizations with identical D&B DUNS number are compared
☑	mcLatitudeLongitude	Business Function Match Code Generator: List<String>, AddressNearbyMa...	Organizations with main address within 500 meter of each other are compared

Match Code Filter

ID	Match Code Filter	Comment

Evaluator

Select Nodes Monster Buy HQ (815357)   Monster Buy HQ (816033)

Matchers	Score
accountGroupTargetMatcher	0.0

Rules	Score	Matched	Match Reason
1	61.111111111111114	false	accountGroupTargetMatcher = false (0.0>70), address = true (100.0), dunsNumberMatcher = true (0.0), legalNameMatcher = true (100.0)
2	0.0	true	accountGroupTargetMatcher = true (0.0), address = true (100.0), dunsNumberMatcher = true (0.0), legalNameMatcher = true (100.0)
Final Score: 0.0			

legalNameNormalizer	{sourceValue=Monster Buy HQ, resultValue=monster buy hq}	{sourceValue=Monster Buy HQ, resultValue=monster buy hq}
dunsNumbersNormalizer	duns111111111	duns111111112, duns111111113
mainAddressLatitude	56.110083	56.110083
mainAddressLongitude	10.156826	10.156826

Common Match Codes: Yes

Match Code Generators	First Node Result	Second Node Result
mcNameAndAddress	ORGANIZATION#MNSTRPK+8270	ORGANIZATION#MNSTRPK+8270
	ORGANIZATION#MNSTRPK+HJPJRK	ORGANIZATION#MNSTRPK+HJPJRK
mcDuns	KEY#DUNS111111111	KEY#DUNS111111112, DUNS111111113
mcLatitudeLongitude	56.11-x-10.152000000000001	56.11-x-10.152000000000001
	56.11-x-10.176	56.11-x-10.176
	56.115-x-10.152000000000001	56.115-x-10.152000000000001
	56.115-x-10.176	56.115-x-10.176

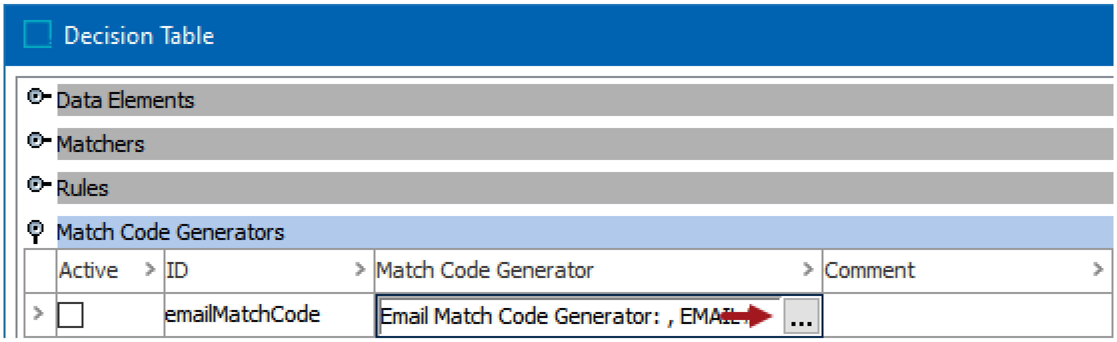
Match Code Generator: Email

Emails are often good match codes. The match code prefix is prepended to every email output by the email normalizer.

Configuring an Email Match Code Generator

After adding the match code generator in the Match Code Generators flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Match Code Generator column and click the ellipsis button (...) to access the configuration dialog.



Decision Table

Data Elements

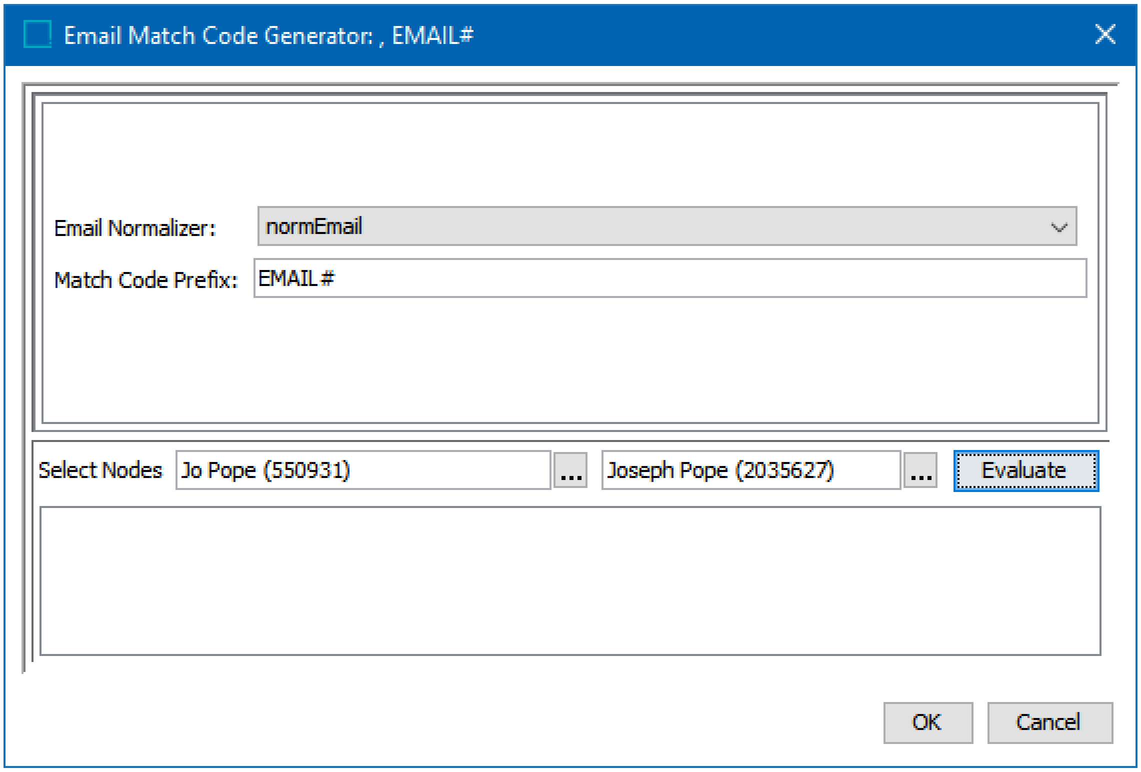
Matchers

Rules

Match Code Generators

Active	ID	Match Code Generator	Comment
□	emailMatchCode	Email Match Code Generator: , EMAIL ...	

2. On the Match Code Generator dialog:



Email Match Code Generator: , EMAIL#

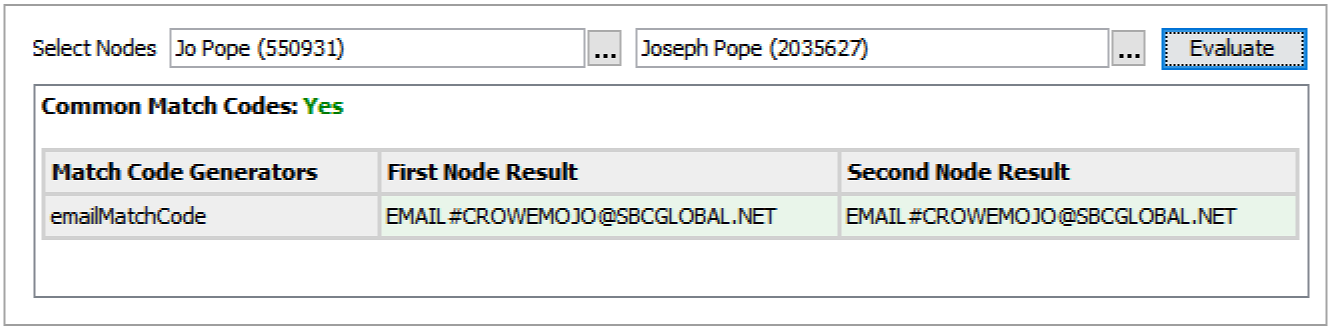
Email Normalizer: normEmail

Match Code Prefix: EMAIL#

Select Nodes Jo Pope (550931) Joseph Pope (2035627)

- For the required **Email Normalizer**, use the dropdown to select the associated normalizer or enter a case-sensitive ID for the normalizer. Refer to the **Data Element: Email Normalizer** topic.
- For the optional **Match Code Prefix**, enter a prefix to identify the source of the match code.

3. To test the configuration:



Select Nodes Jo Pope (550931) Joseph Pope (2035627)

Common Match Codes: Yes

Match Code Generators	First Node Result	Second Node Result
emailMatchCode	EMAIL#CROWEMOJO@SBCGLOBAL.NET	EMAIL#CROWEMOJO@SBCGLOBAL.NET

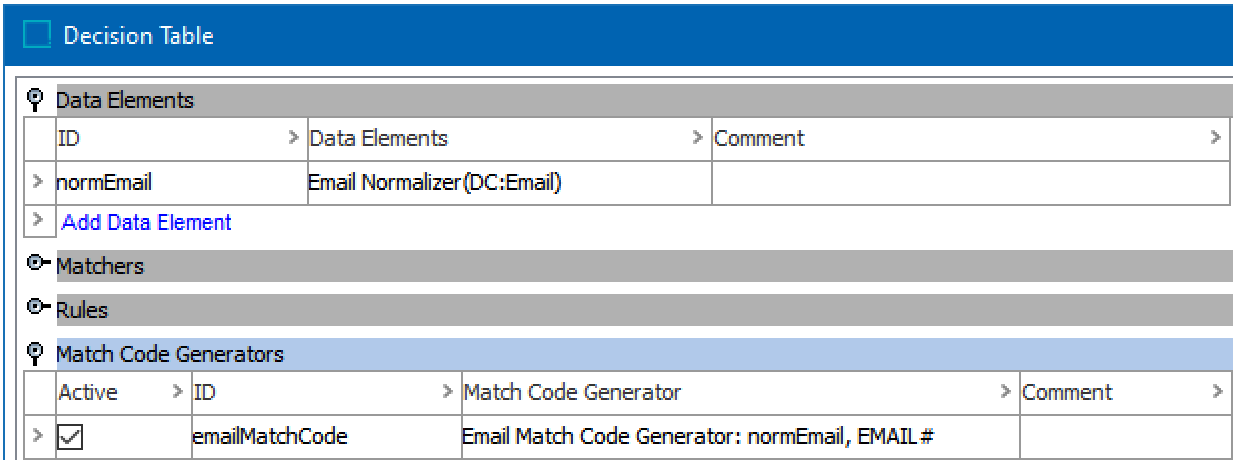
- Click the ellipsis button (...) for each **Select Nodes** parameter and select two objects for comparison.
- Click the **Evaluate** button.

- The **Common Match Codes** parameter displays 'Yes' when the same match code is generated for the selected nodes. When 'No' is displayed, either one or both nodes have no value, or when the generated match codes are not the same.

  Adjust as indicated by the Evaluator results and repeat the evaluation if necessary.

4. Click **OK** to save and display the configuration in the Match Code Generators flipper.

5. Ensure the Active checkbox is checked to create match codes with this generator.



Decision Table

Data Elements

ID	Data Elements	Comment
normEmail	Email Normalizer(DC:Email)	
Add Data Element		

Matchers

Rules

Match Code Generators

Active	ID	Match Code Generator	Comment
☑	emailMatchCode	Email Match Code Generator: normEmail, EMAIL#	

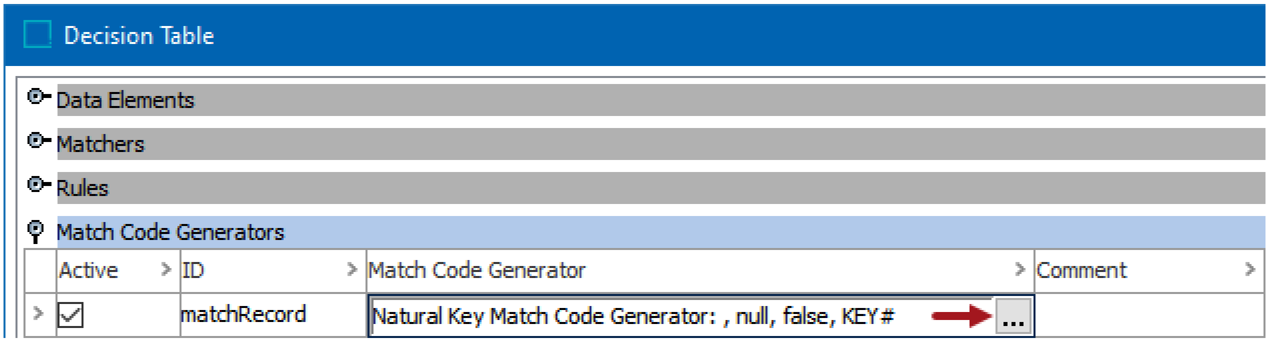
Match Code Generator: Natural Key

Often, objects have data that are naturally good keys, and those often also make good match codes. Some examples of natural keys include: health insurance ID numbers, organization DUNS numbers, VAT numbers, and other IDs from external systems that are not already used as source system IDs.

Configuring a Natural Key Match Code Generator

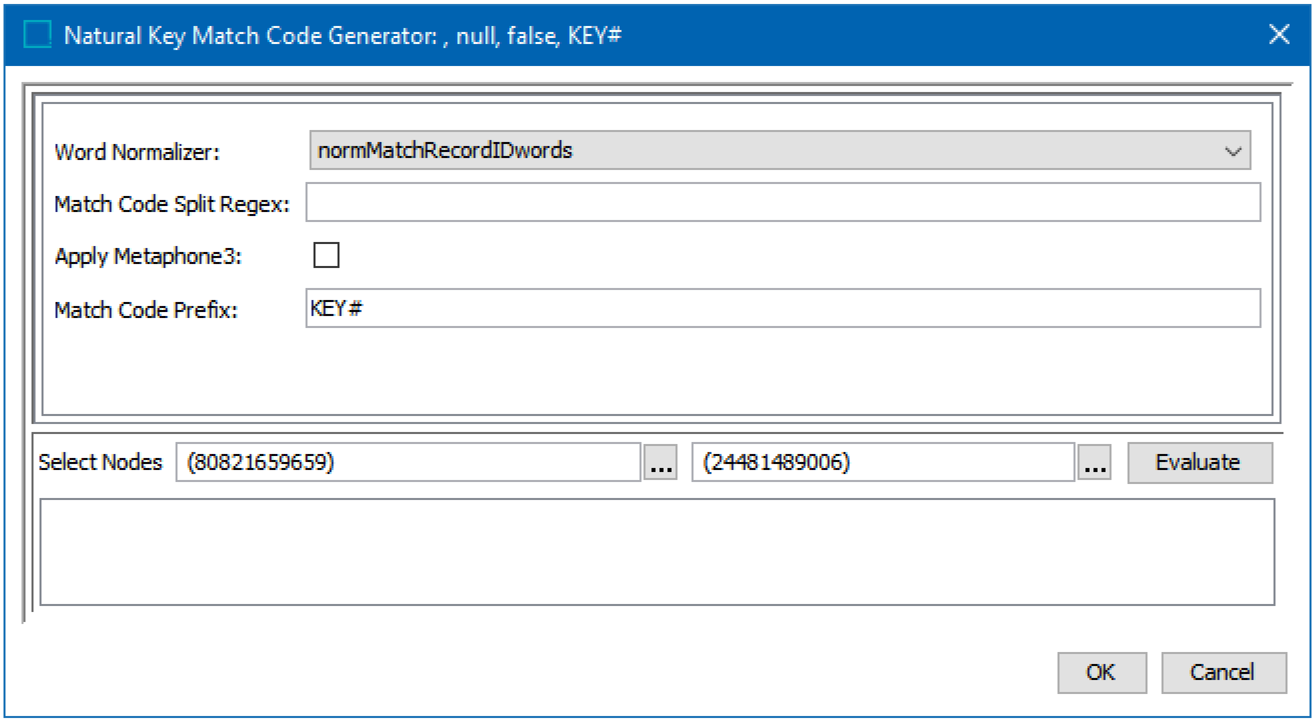
After adding the match code generator in the Match Code Generators flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Match Code Generator column and click the ellipsis button (...) to access the configuration dialog.



Active	ID	Match Code Generator	Comment
☑	matchRecord	Natural Key Match Code Generator: , null, false, KEY# ...	

2. On the Match Code Generator dialog:



Natural Key Match Code Generator: , null, false, KEY#

Word Normalizer: normMatchRecordIDwords

Match Code Split Regex:

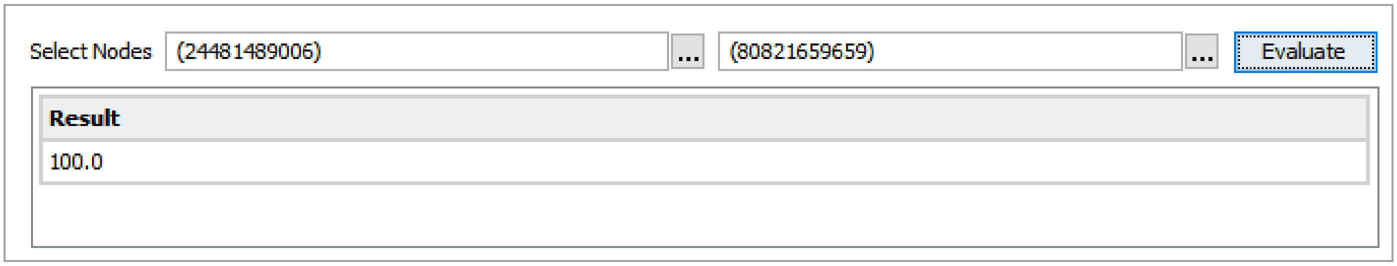
Apply Metaphone3: □

Match Code Prefix: KEY#

Select Nodes (80821659659) (24481489006)

- For the required **Word Normalizer**, use the dropdown to select the associated normalizer or enter a case-sensitive ID for the normalizer.
- For the optional **Match Code Split Regex**, add RegEx to split the natural key into multiple match codes. When blank, the natural key output by the selected Words normalizer is used.
- For **Apply Metaphone3**, when checked, the Metaphone 3 algorithm (which expands on Soundex) compares names based on their pronunciation. It works well on English words, non-English words familiar to Americans, first names, and family names commonly found in the United States. For more information on Metaphone 3, search the web.
- For the optional **Match Code Prefix**, enter a prefix to identify the source of the match code.

3. To test the configuration:



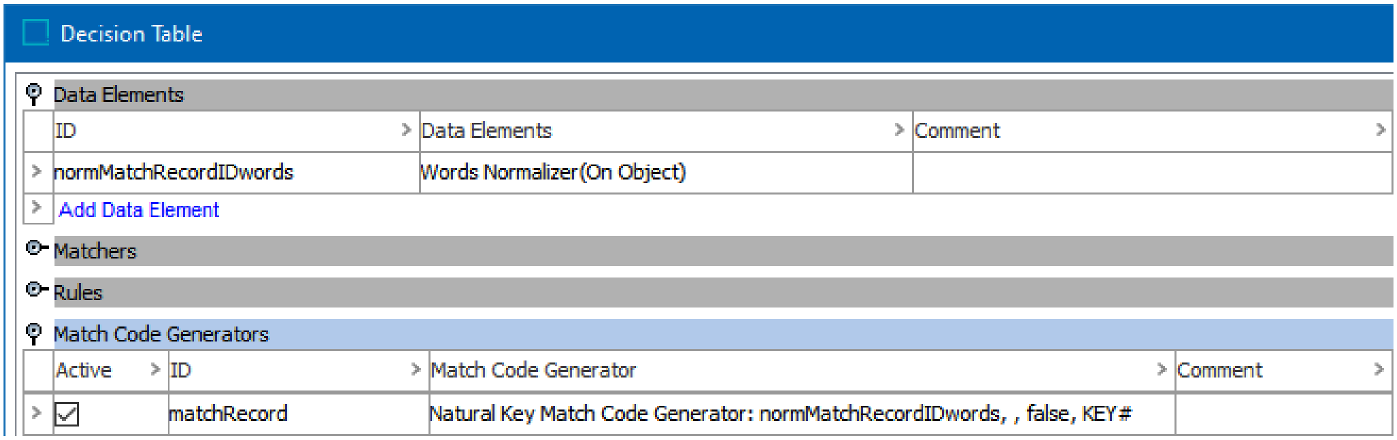
Select Nodes (24481489006) (80821659659)

Result
100.0

- Click the ellipsis button (**...**) for each **Select Nodes** parameter and select two objects for comparison.
- Click the **Evaluate** button.

  Adjust as indicated by the Evaluator results and repeat the evaluation if necessary.

4. Click **OK** to save and display the configuration in the Match Code Generators flipper.
5. Ensure the **Active** checkbox is checked to create match codes with this generator.



Data Elements

ID	Data Elements	Comment
normMatchRecordIDwords	Words Normalizer(On Object)	
Add Data Element		

Match Code Generators

Active	ID	Match Code Generator	Comment
☑	matchRecord	Natural Key Match Code Generator: normMatchRecordIDwords, , false, KEY#	

Match Code Generator: Organization Name and Address

The combination of organization name and address usually constitutes a good match code since organization names on their own are often not unique. For example, a supermarket chain can use the same organization name for each of their sites, so another piece of data is needed to achieve a unique 'name' to identify a site. A similar case can be made for matching on addresses alone.

Recommendations

Metaphone 3 (which expands on Soundex) Organization Name Token can be used to compare names based on their pronunciation. It works well on English words, non-English words familiar to Americans, first names, and family names commonly found in the United States. For more information on Metaphone 3, search the web.

Consider the following for better matching results:

- **Handle most aliases in the normalizer**, but completely ignore some aliases in the match codes while applying a high non-zero missing word score in the matcher. The Organization Name and Address Match Code Generator has the option to add a Name Alias replacement table which can be used to handle abbreviated organization names and names appended with terms like 'Inc.' that are often omitted.
- **Add the smallest number of match codes** to improve the performance of the system. Although the Organization Name and Address Match Code Generator can add up to four match codes for every organization (indicated by the checkboxes on the configuration screen):
  - Avoid enabling both the 'Metaphone3 Organization Name Token + ZIP code' and 'Metaphone3 Organization Name Token + ZIP code + Street name' since the first is a superset of the second.
  - If ' Metaphone3 Organization Name Token + ZIP code' is expected to create a lot of equal match codes where records should not be merged, instead use the 'Metaphone3 Organization Name Token + ZIP code + Street name' match code.

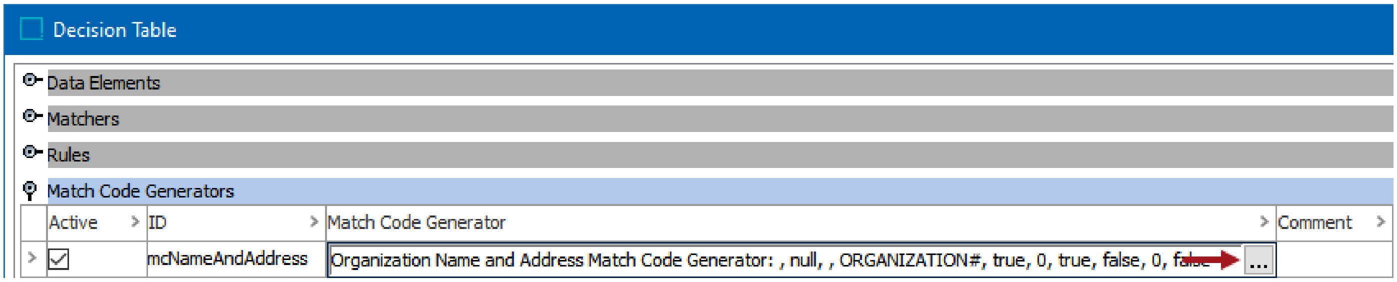
Considerations

An **Organization Name Aliases** table can be used to remove name elements that should only be removed in match codes. This can allow a matcher and match code generator to share an organization name normalizer.

Configuring an Organization Name and Address Match Code Generator

After adding the match code generator in the Match Code Generators flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

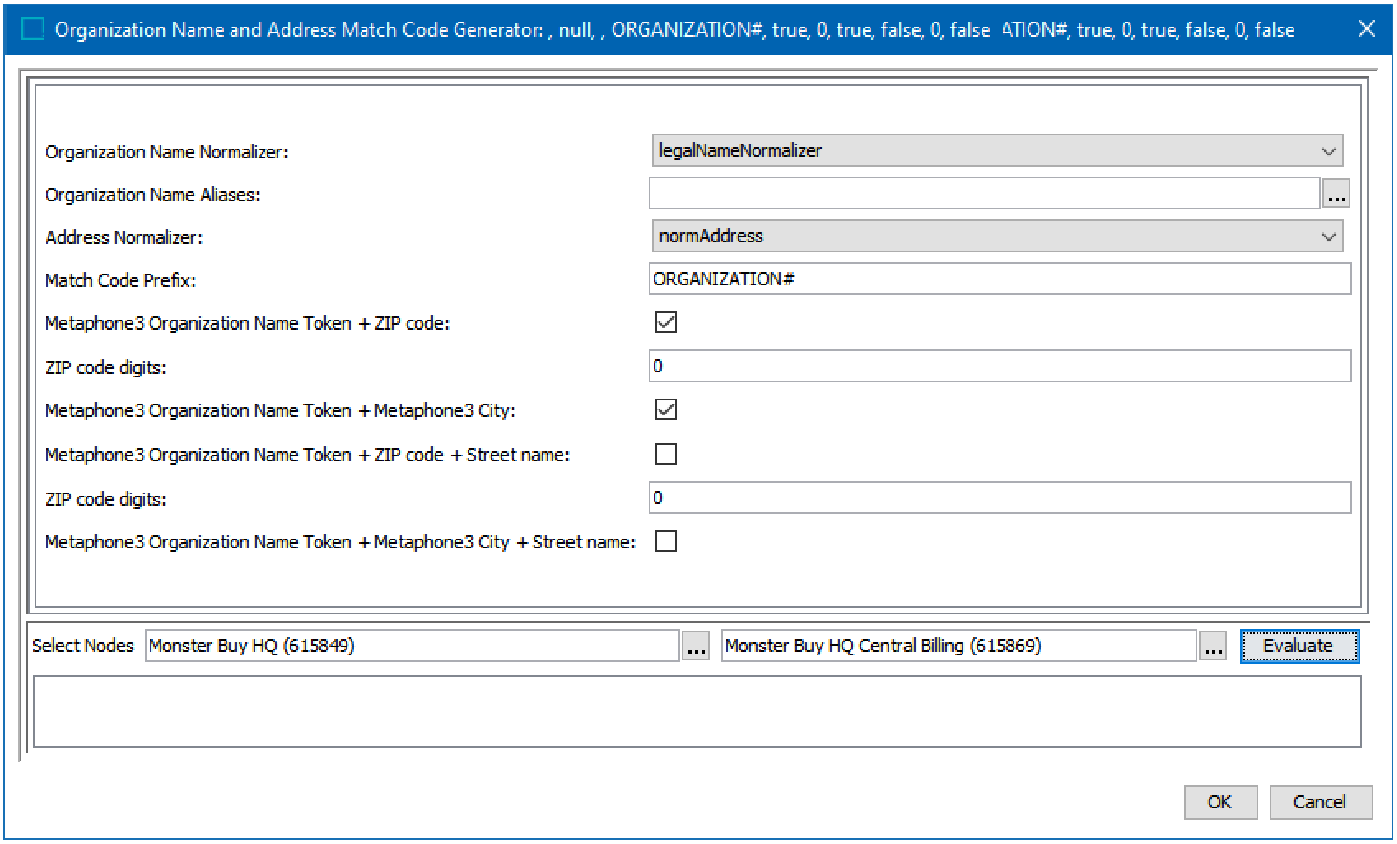
1. Click into the Match Code Generator column and click the ellipsis button (...) to access the configuration dialog.



Decision Table

Active	ID	Match Code Generator	Comment
☑	mcNameAndAddress	Organization Name and Address Match Code Generator: , null, , ORGANIZATION#, true, 0, true, false, 0, false ...	

2. On the Match Code Generator dialog:



Organization Name and Address Match Code Generator: , null, , ORGANIZATION#, true, 0, true, false, 0, false ATION#, true, 0, true, false, 0, false

Organization Name Normalizer:	legalNameNormalizer
Organization Name Aliases:	...
Address Normalizer:	normAddress
Match Code Prefix:	ORGANIZATION#
Metaphone3 Organization Name Token + ZIP code:	☑
ZIP code digits:	0
Metaphone3 Organization Name Token + Metaphone3 City:	☑
Metaphone3 Organization Name Token + ZIP code + Street name:	☐
ZIP code digits:	0
Metaphone3 Organization Name Token + Metaphone3 City + Street name:	☐

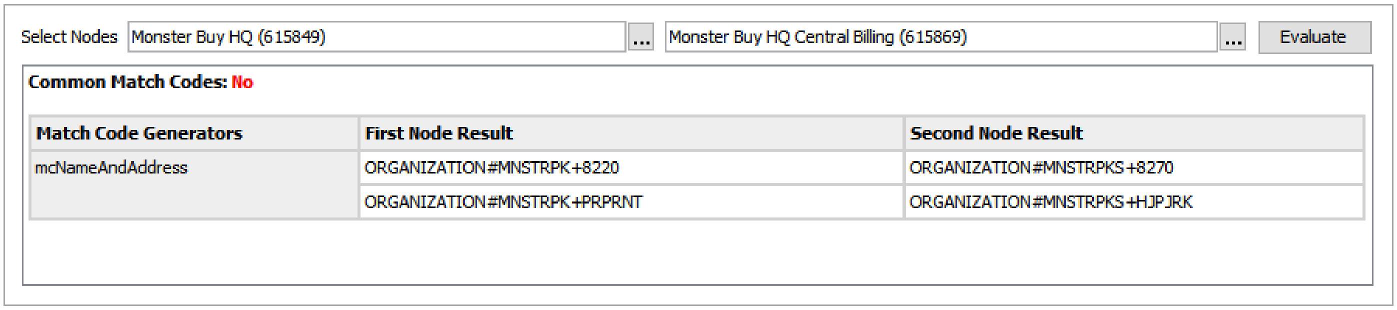
Select Nodes: Monster Buy HQ (615849) ... Monster Buy HQ Central Billing (615869) ...

- For the required **Organization Name Normalizer**, use the dropdown to select the associated normalizer or enter a case-sensitive ID for the normalizer. Refer to the **Data Element: Organization Name Normalizer** topic.

- For the optional **Organization Name Aliases**, click the ellipsis button to select a Name Alias replacement table. Refer to the **Considerations** section above.

- For the required **Address Normalizer**, use the dropdown to select the associated normalizer or enter a case-sensitive ID for the normalizer. Refer to the **Data Element: Address Normalizer** topic.
- For the optional **Match Code Prefix**, enter a prefix to identify the source of the match code.
- For the optional **Metaphone3 Organization Name Token + Zip code**, check to create this match code. Refer to the **Recommendations** section above.
- For the optional **ZIP code digits** (from the previous Metaphone3 option), enter the number of ZIP code digits, starting from the beginning, to include in the match code. When this entry is 0, the entire zip code is used in the match code.
- For the optional **Metaphone3 Organization Name Token + Metaphone3 City**, check to create this match code. Refer to the **Recommendations** section above.
- For the optional **Metaphone3 Organization Name Token + Zip code + Street name**, check to create this match code. Refer to the **Recommendations** section above.
- For the optional **ZIP code digits** (from the previous Metaphone3 option), enter the number of ZIP code digits, starting from the beginning, to include in the match code. When this entry is 0, the entire zip code is used in the match code.
- For the optional **Metaphone3 Organization Name Token + Metaphone3 City + Street name**, check to create this match code. Refer to the **Recommendations** section above.

3. To test the configuration:



Select Nodes Monster Buy HQ (615849) Monster Buy HQ Central Billing (615869)

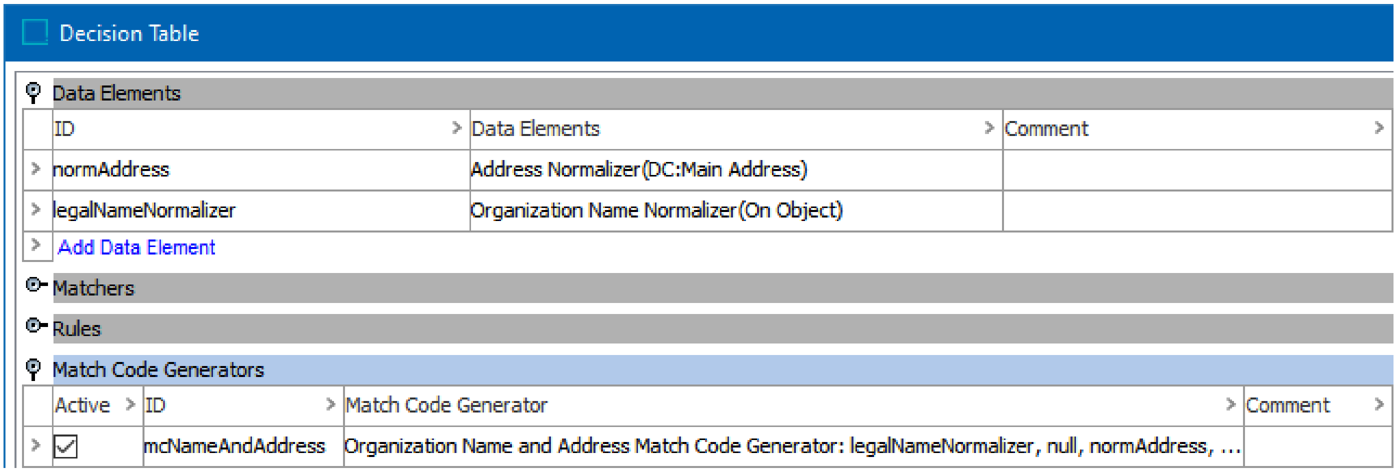
Common Match Codes: No

Match Code Generators	First Node Result	Second Node Result
mcNameAndAddress	ORGANIZATION#MNSTRPK+8220	ORGANIZATION#MNSTRPKS+8270
	ORGANIZATION#MNSTRPK+PRPRNT	ORGANIZATION#MNSTRPKS+HJPJRK

- Click the ellipsis button (...) for each **Select Nodes** parameter and select two objects for comparison.
- Click the **Evaluate** button.
- The **Common Match Codes** parameter displays 'Yes' when the same match code is generated for the selected nodes. When 'No' is displayed, either one or both nodes have no value, or when the generated match codes are not the same.

  Adjust as indicated by the Evaluator results and repeat the evaluation if necessary.

4. Click **OK** to save and display the configuration in the Match Code Generators flipper.
5. Ensure the **Active** checkbox is checked to create match codes with this generator.



Decision Table

Data Elements

ID	Data Elements	Comment
normAddress	Address Normalizer(DC:Main Address)	
legalNameNormalizer	Organization Name Normalizer(On Object)	

Add Data Element

Matchers

Rules

Match Code Generators

Active	ID	Match Code Generator	Comment
☑	mcNameAndAddress	Organization Name and Address Match Code Generator: legalNameNormalizer, null, normAddress, ...	

Match Code Generator: Person Name and Address

Ideally, two records with the same person names should be matched, but unfortunately, person names are not unique. Using person names as match codes leads to a lot of unnecessary comparisons. Also, any match code on address alone causes spouses to be compared. However, the combination of person name and address usually constitutes a good match code.

Recommendations

Metaphone 3 (which expands on Soundex) Last Name and Metaphone 3 First Name can be used to compare names based on their pronunciation. It works well on English words, non-English words familiar to Americans, first names, and family names commonly found in the United States. For more information on Metaphone 3, search the web.

Consider the following for better matching results:

- **Handle most aliases in the normalizer**. Although the normalizer is the place to account for aliases, sometimes, the normalizer for matching and the normalizer for match codes are very similar except for a few aliases that need to be completely ignored in the match code, while only assigned a lower score in the matcher. This can be solved by using two separate normalizers, or in some cases by using the extra alias table on the match code generator. The Person Name and Address Match Code Generator has the option to add a First Name Aliases replacement table which can be used to handle abbreviated names and exchanged names like 'Bob' in place of ‘Robert’. Use the alias table to ensure full replacement in match codes but still get the appropriate Unmatched Word Factor, Initials Match Factor, etc., from the matcher.
- **Consider match code group sizes** in B2C since there are often a lot of records
    - Use **Match Tuning** and **Data Profiling** to make good estimates of match code group sizes before importing.
    - Use the **Match Codes** Values tab on the Matching Algorithm to evaluate the soundness of match code group sizes on live systems.
- **Add the smallest number of match codes** to improve the performance of the system. Although the Person Name and Address Match Code Generator can add up to six match codes for every person (indicated by the checkboxes on the configuration screen), typically not all are necessary.

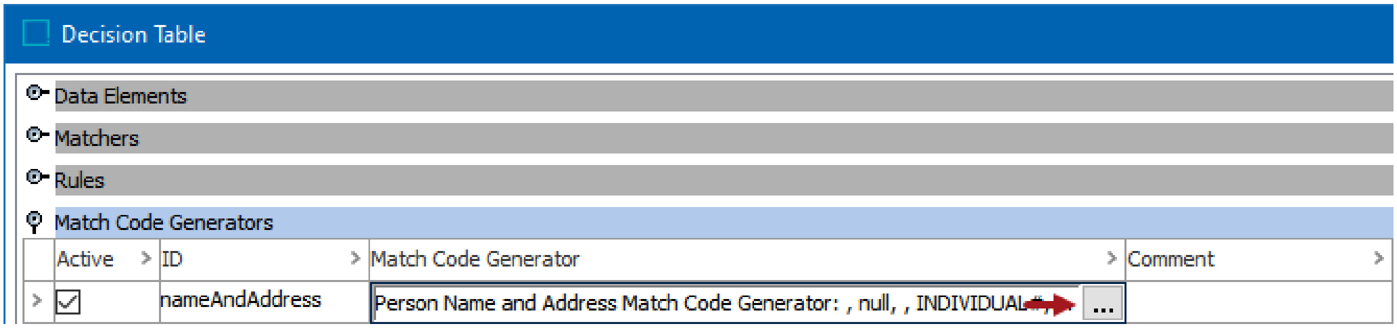
Considerations

A **Person Name Aliases** table can be used to remove name elements that should only be removed in match codes. This can allow a matcher and match code generator to share a person name normalizer.

Configuring a Person Name and Address Match Code Generator

After adding the match code generator in the Match Code Generators flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Match Code Generator column and click the ellipsis button (**...**) to access the configuration dialog.



Decision Table

Data Elements

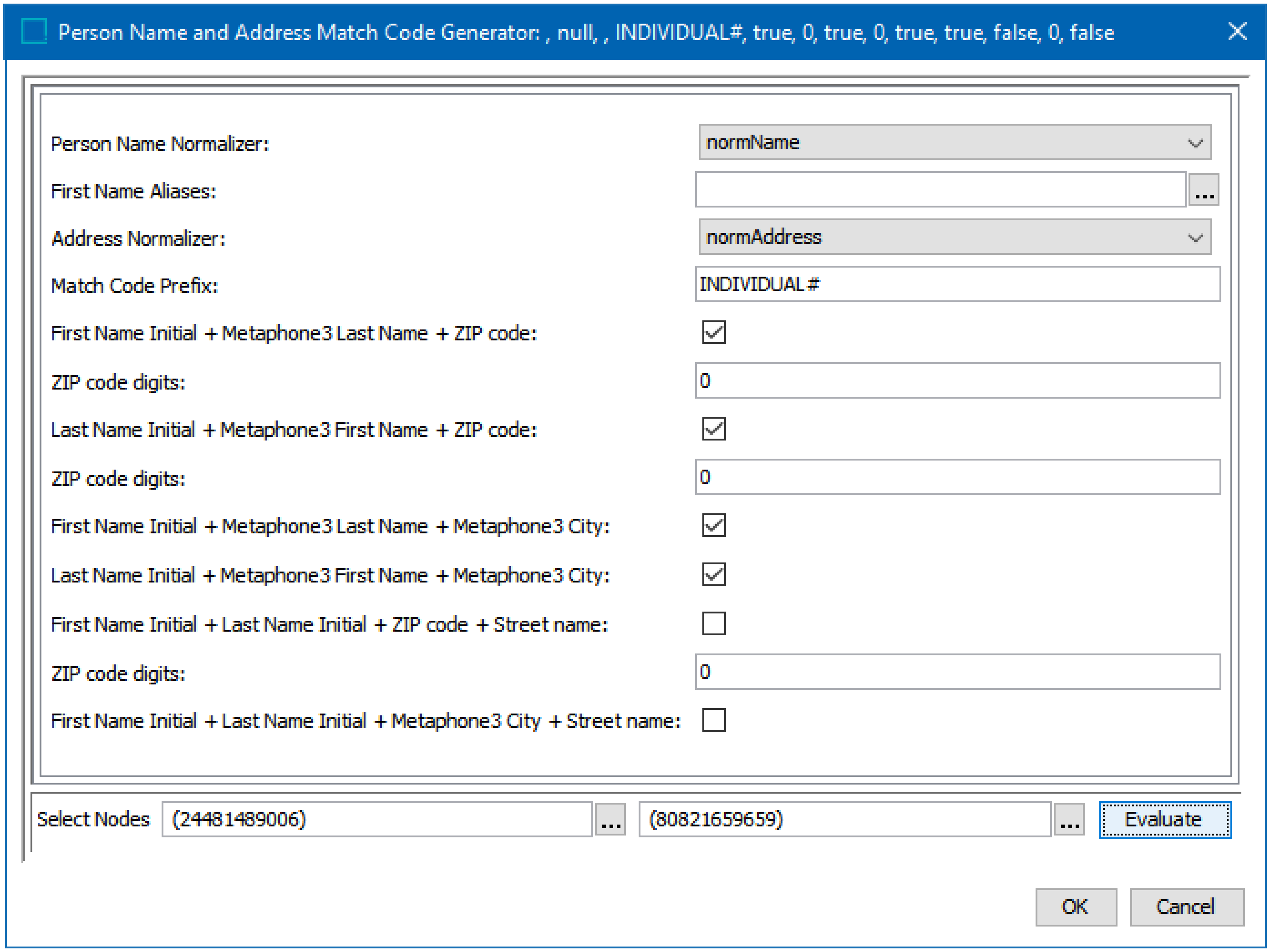
Matchers

Rules

Match Code Generators

	Active	ID	Match Code Generator	Comment
>	☑	nameAndAddress	Person Name and Address Match Code Generator: , null, , INDIVIDUAL#, ...	

2. On the Match Code Generator dialog:



Person Name and Address Match Code Generator: , null, , INDIVIDUAL#, true, 0, true, 0, true, true, false, 0, false

Person Name Normalizer:	normName
First Name Aliases:	
Address Normalizer:	normAddress
Match Code Prefix:	INDIVIDUAL#
First Name Initial + Metaphone3 Last Name + ZIP code:	☑
ZIP code digits:	0
Last Name Initial + Metaphone3 First Name + ZIP code:	☑
ZIP code digits:	0
First Name Initial + Metaphone3 Last Name + Metaphone3 City:	☑
Last Name Initial + Metaphone3 First Name + Metaphone3 City:	☑
First Name Initial + Last Name Initial + ZIP code + Street name:	☐
ZIP code digits:	0
First Name Initial + Last Name Initial + Metaphone3 City + Street name:	☐

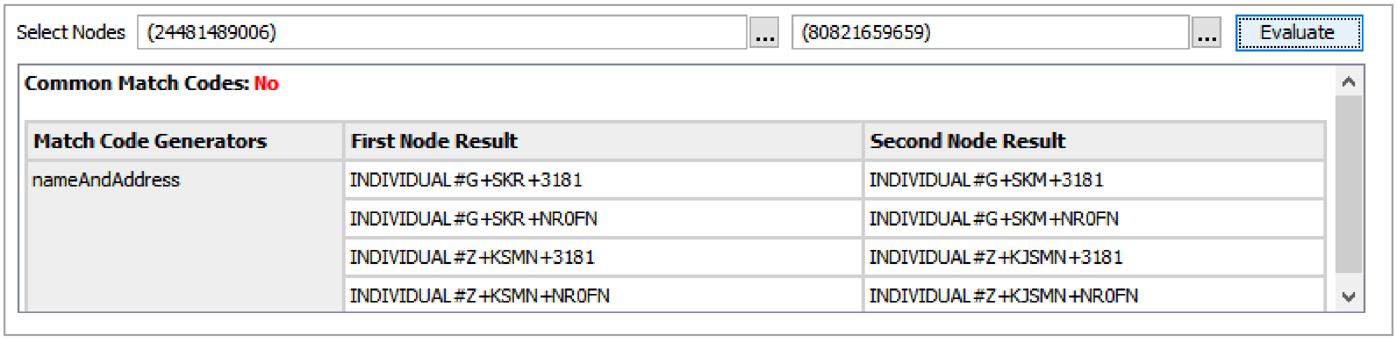
Select Nodes (24481489006) (80821659659)

- For the required **Person Name Normalizer**, use the dropdown to select the associated normalizer or enter a case-sensitive ID for the normalizer. Refer to the **Data Element: Person Name Normalizer** topic.
- For the optional **First Name Aliases**, click the ellipsis button to select either:
  - a Name Alias replacement table, or
  - a Person Name Alias Group Asset. Additional match codes will be generated which will be one extra ‘Person Name and Address’ match code for each nickname group the person name is part of. The name used in the match code will always be the longest name found in each nickname group. For detailed information about the component model configuration of the Person Name Alias Group Asset Type, refer to the Configuring Matching Component Model topic in theMatching, Linking, and Merging documentation.

    Refer to the **Considerations** section above.

- For the required **Address Normalizer**, use the dropdown to select the associated normalizer or enter a case-sensitive ID for the normalizer. Refer to the **Data Element: Address Normalizer** topic.
- For the optional **Match Code Prefix**, enter a prefix to identify the source of the match code.
- For the optional **First Name Initial + Metaphone3 Last Name + Zip code**, check to create this match code. Refer to the **Recommendations** section above.
- For the optional **ZIP code digits** (from the previous Metaphone3 option), enter the number of ZIP code digits, starting from the beginning, to include in the match code. When this entry is 0, the entire zip code is used in the match code.
- For the optional **Last Name Initial + Metaphone3 First Name + Zip code**, check to create this match code. Refer to the **Recommendations** section above.
- For the optional **ZIP code digits** (from the previous Metaphone3 option), enter the number of ZIP code digits, starting from the beginning, to include in the match code. When this entry is 0, the entire zip code is used in the match code.
- For the optional **First Name Initial + Metaphone3 Last Name + Metaphone3 City**, check to create this match code. Refer to the **Recommendations** section above.
- For the optional **Last Name Initial + Metaphone3 First Name + Metaphone3 City**, check to create this match code. Refer to the **Recommendations** section above.
- For the **First Name Initial + Last Name Initial + ZIP Code + Street Name**, check to create this match code. Refer to the **Recommendations** section above.
- For the optional **ZIP code digits** (from the previous Metaphone3 option), enter the number of ZIP code digits, starting from the beginning, to include in the match code. When this entry is 0, the entire zip code is used in the match code.
- For the **First Name Initial + Last Name Initial + Metaphone3 City + Street Name**, check to create this match code. Refer to the **Recommendations** section above.

3. To test the configuration:



Select Nodes (24481489006) (80821659659)

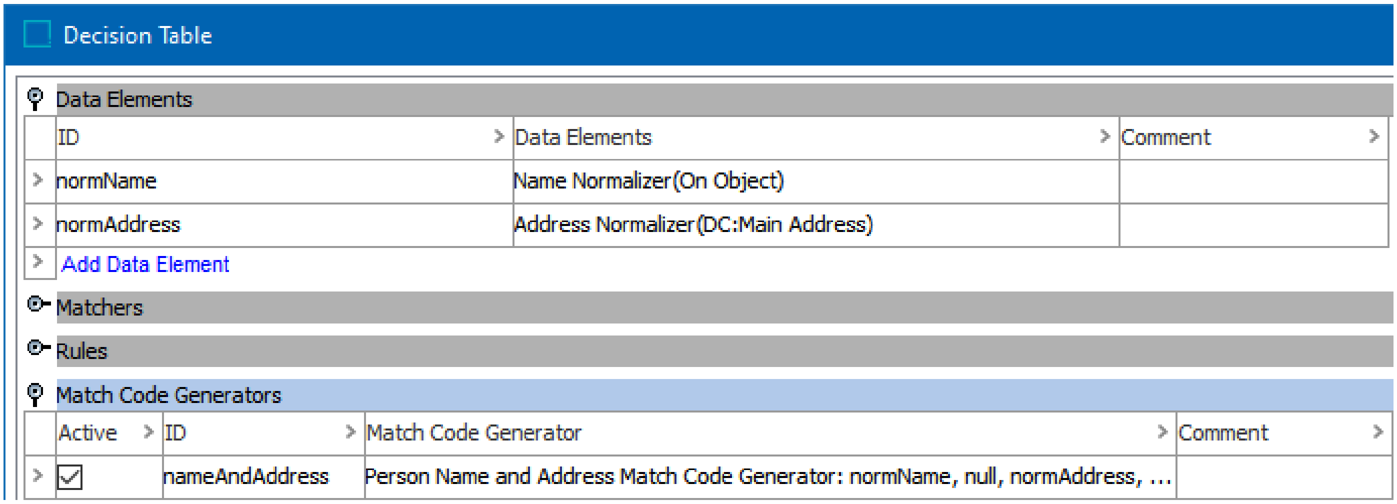
Common Match Codes: No

Match Code Generators	First Node Result	Second Node Result
nameAndAddress	INDIVIDUAL#G+SKR+3181	INDIVIDUAL#G+SKM+3181
	INDIVIDUAL#G+SKR+NR0FN	INDIVIDUAL#G+SKM+NR0FN
	INDIVIDUAL#Z+KSMN+3181	INDIVIDUAL#Z+KJSMN+3181
	INDIVIDUAL#Z+KSMN+NR0FN	INDIVIDUAL#Z+KJSMN+NR0FN

- Click the ellipsis button (**...**) for each **Select Nodes** parameter and select two objects for comparison.
- Click the **Evaluate** button.
- The **Common Match Codes** parameter displays 'Yes' when the same match code is generated for the selected nodes. When 'No' is displayed, either one or both nodes have no value, or when the generated match codes are not the same.

  Adjust as indicated by the Evaluator results and repeat the evaluation if necessary.

4. Click **OK** to save and display the configuration in the Match Code Generators flipper.
5. Ensure the **Active** checkbox is checked to create match codes with this generator.



Decision Table

Data Elements

ID	Data Elements	Comment
normName	Name Normalizer(On Object)	
normAddress	Address Normalizer(DC:Main Address)	
Add Data Element		

Matchers

Rules

Match Code Generators

Active	ID	Match Code Generator	Comment
☑	nameAndAddress	Person Name and Address Match Code Generator: normName, null, normAddress, ...	

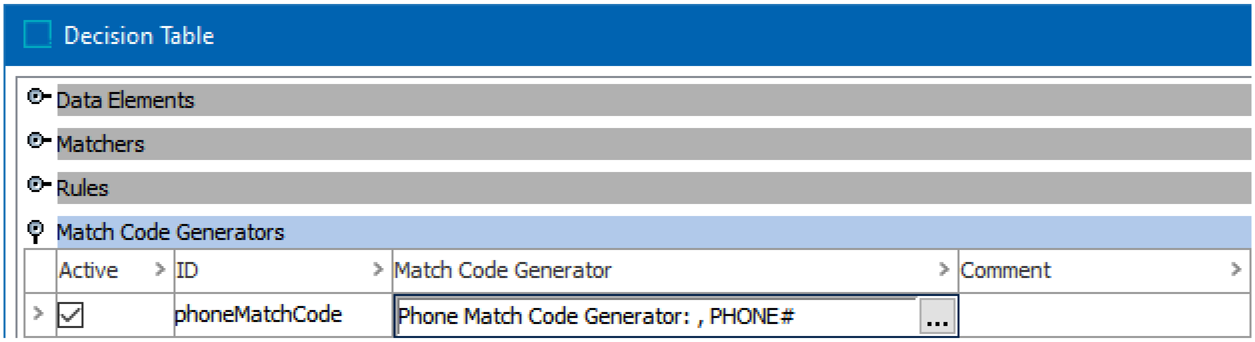
Match Code Generator: Phone

Phone numbers are naturally unique and are often very good match codes. The Phone Match Code Generator concatenates the prefix, country code, and phone number.

Configuring a Phone Match Code Generator

After adding the match code generator in the Match Code Generators flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. Click into the Match Code Generator column and click the ellipsis button (...) to access the configuration dialog.



Decision Table

Data Elements

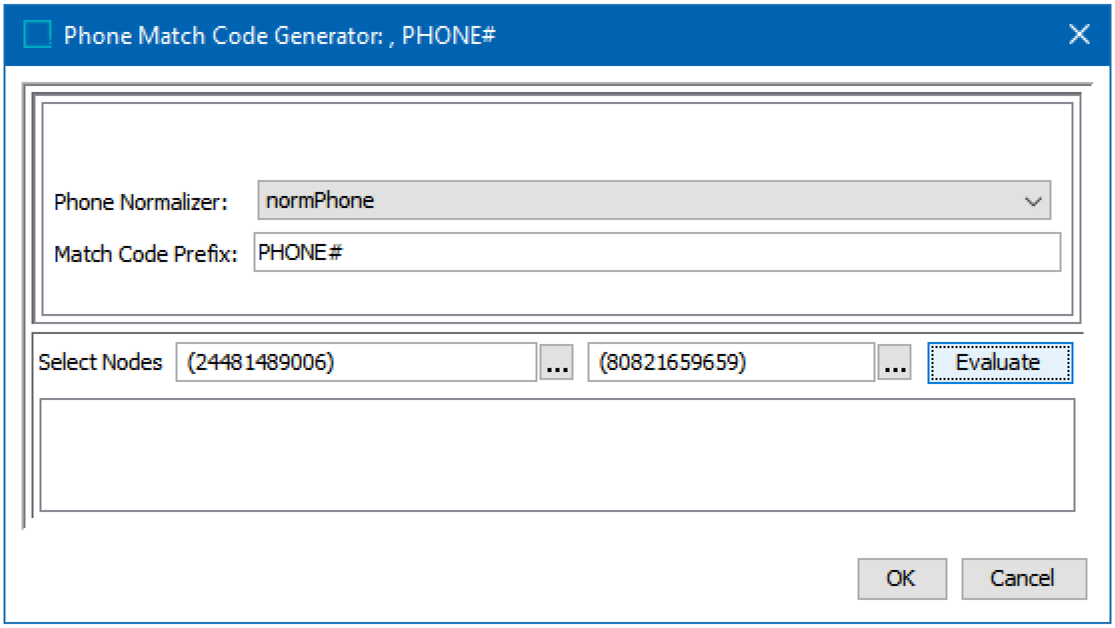
Matchers

Rules

Match Code Generators

Active	ID	Match Code Generator	Comment
☑	phoneMatchCode	Phone Match Code Generator: , PHONE#	

2. On the Match Code Generator dialog:



Phone Match Code Generator: , PHONE#

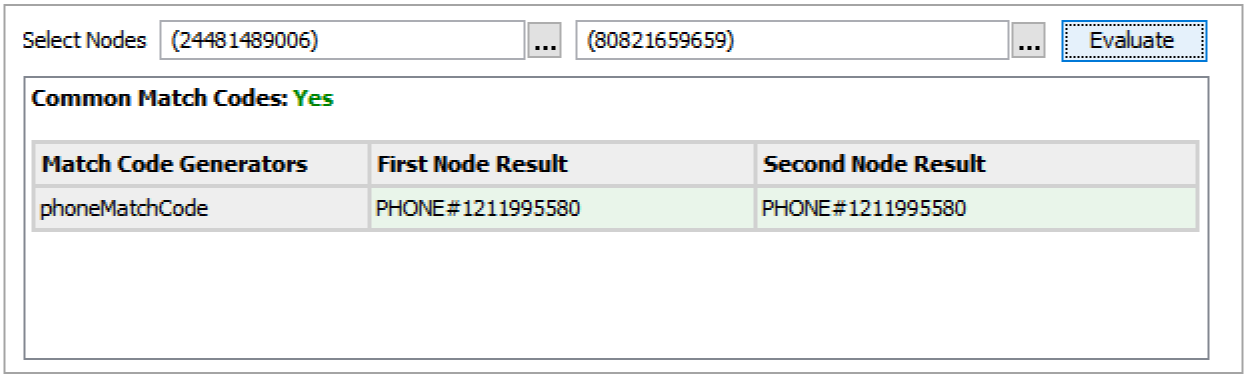
Phone Normalizer: normPhone

Match Code Prefix: PHONE#

Select Nodes (24481489006) (80821659659)

- For the required **Phone Normalizer**, use the dropdown to select the associated normalizer or enter a case-sensitive ID for the normalizer. Refer to the **Data Element: Phone Normalizer** topic.
- For the optional **Match Code Prefix**, enter a prefix to identify the source of the match code.

3. To test the configuration:



Select Nodes (24481489006) ... (80821659659) ... Evaluate

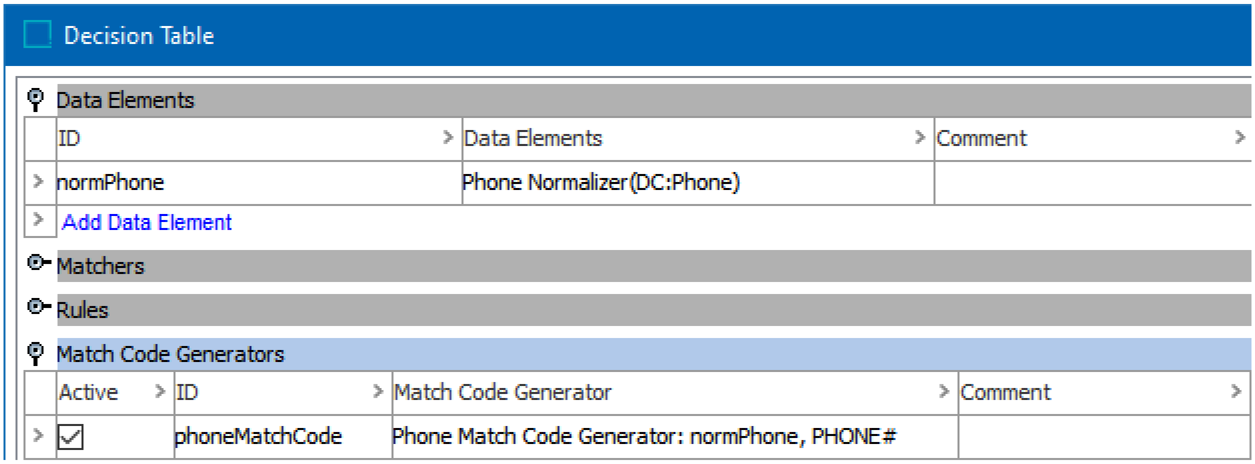
Common Match Codes: Yes

Match Code Generators	First Node Result	Second Node Result
phoneMatchCode	PHONE#1211995580	PHONE#1211995580

- Click the ellipsis button (...) for each **Select Nodes** parameter and select two objects for comparison.
- Click the **Evaluate** button.
- The **Common Match Codes** parameter displays 'Yes' when the same match code is generated for the selected nodes. When 'No' is displayed, either one or both nodes have no value, or when the generated match codes are not the same.

  Adjust as indicated by the Evaluator results and repeat the evaluation if necessary.

4. Click **OK** to save and display the configuration in the Match Code Generators flipper.
5. Ensure the **Active** checkbox is checked to create match codes with this generator.



Decision Table

Data Elements

ID	Data Elements	Comment
normPhone	Phone Normalizer(DC:Phone)	
Add Data Element		

Matchers

Rules

Match Code Generators

Active	ID	Match Code Generator	Comment
☑	phoneMatchCode	Phone Match Code Generator: normPhone, PHONE#	

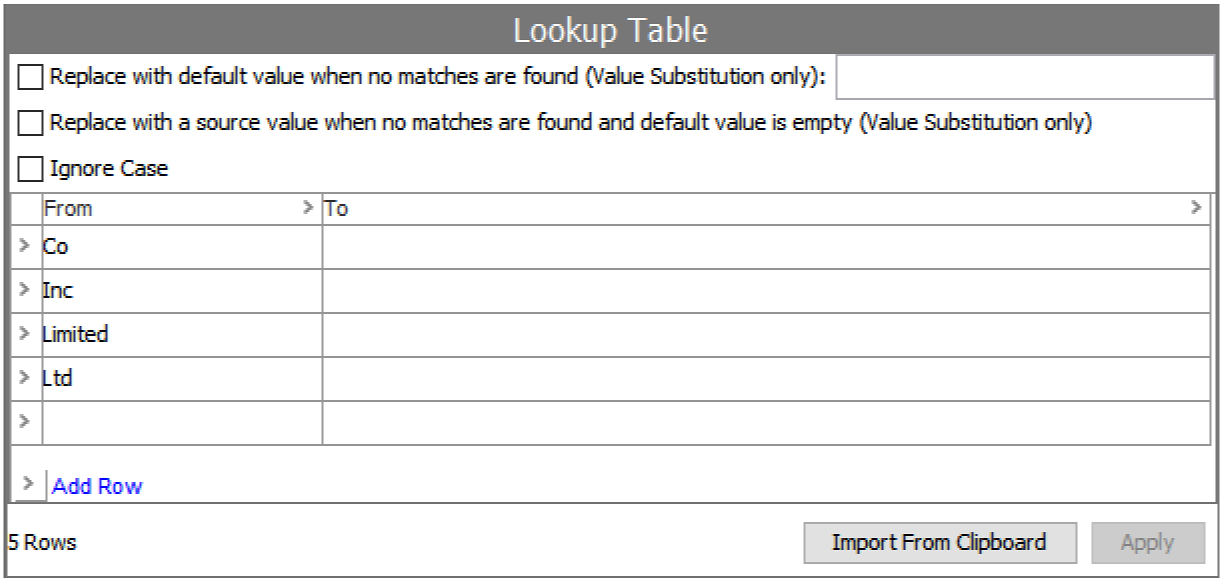
Match Criteria Match Code Filter

Data exceptions can create large match code groups that result in comparing all records in the group. Large match code groups can be identified using the Match Code Values tab in the matching algorithm object. A match code filter is based on a table of specific match codes that should be filtered out.

> **Note:** Match code filters can only be used for matching algorithms that have been created with the Embed Match Code checkbox selected.

Prerequisites

A transformation lookup table is required to identify all the match codes to be excluded. Enter the codes to be excluded in the 'From' column and leave the 'To' column empty. For more information, refer to the **Transformation Lookup Tables** topic in the **Resource Materials** section of the online help.



Lookup Table

☐ Replace with default value when no matches are found (Value Substitution only):

☐ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

☐ Ignore Case

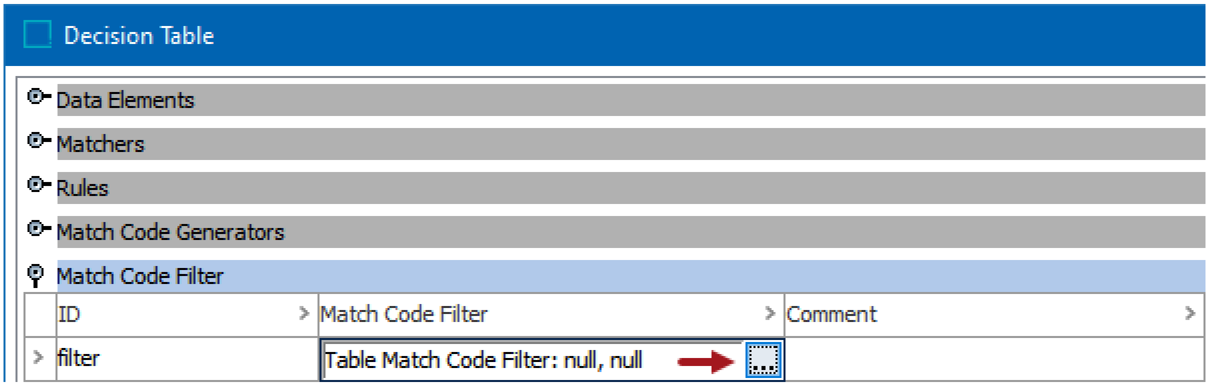
From	To
Co	
Inc	
Limited	
Ltd	

5 Rows

Configuring a Match Code Filter

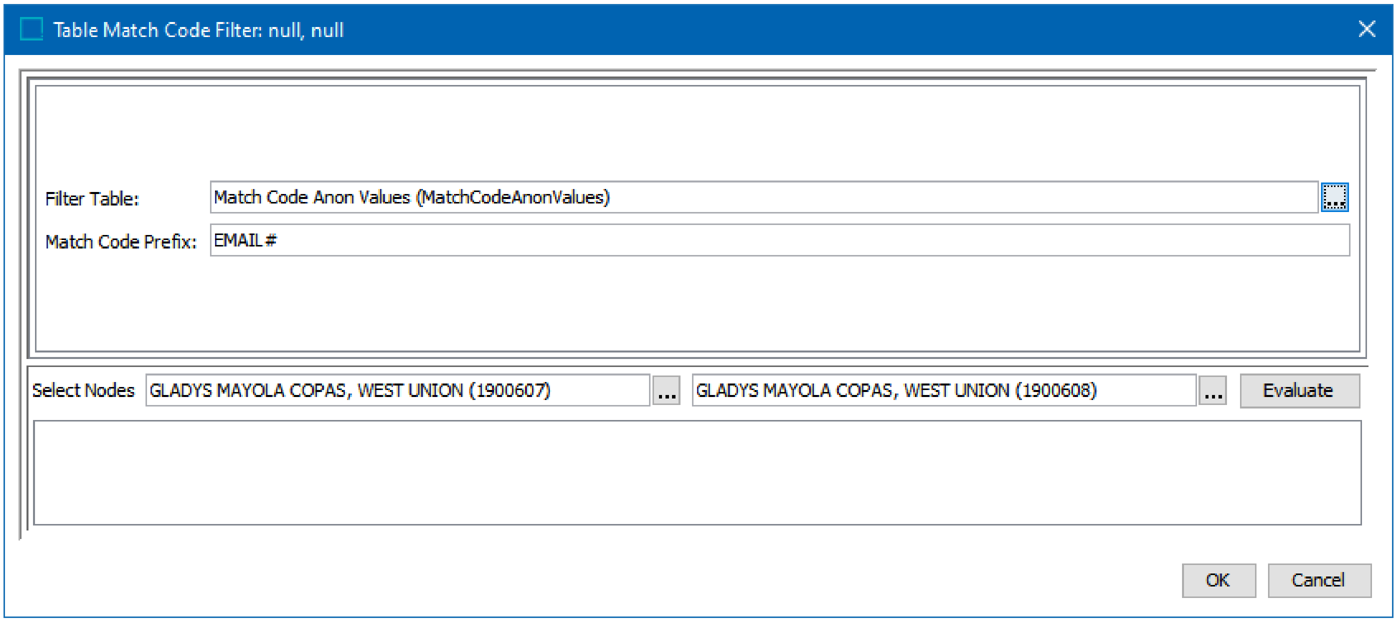
After adding the filter in the Match Code Filter flipper of the Decision Table dialog (defined in the **Match Criteria** topic), configure it as follows:

1. In the Match Code Filter column, on the table match code filter row, click the ellipsis button (**...**) to edit the filter.



ID	Match Code Filter	Comment
filter	Table Match Code Filter: null, null	

2. On the Table Match Code Filter dialog:



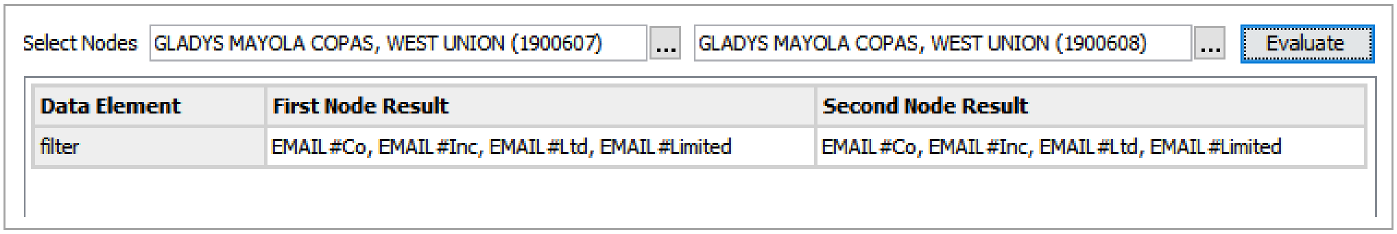
Filter Table: Match Code Anon Values (MatchCodeAnonValues)

Match Code Prefix: EMAIL#

Select Nodes GLADYS MAYOLA COPAS, WEST UNION (1900607) GLADYS MAYOLA COPAS, WEST UNION (1900608)

- For the **Filter Table** parameter, select the configured transformation table.
- For the **Match Code Prefix** parameter, add text to be prepended to all match codes. Leave this parameter blank if all the match codes in the transformation lookup table already have a common prefix.

3. To test the configuration, for the **Select Nodes** parameters:



Select Nodes GLADYS MAYOLA COPAS, WEST UNION (1900607) GLADYS MAYOLA COPAS, WEST UNION (1900608)

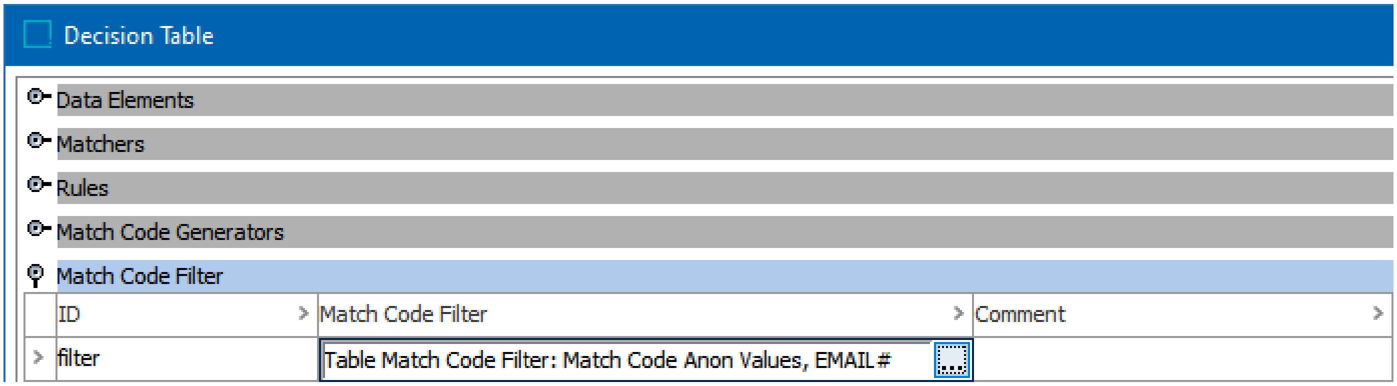
Data Element	First Node Result	Second Node Result
filter	EMAIL#Co, EMAIL#Inc, EMAIL#Ltd, EMAIL#Limited	EMAIL#Co, EMAIL#Inc, EMAIL#Ltd, EMAIL#Limited

- Click the ellipsis button (...) for each field and select two objects for comparison.
- Click the **Evaluate** button.

  0.0 is displayed when a value is not available in one of the selected nodes or when the addresses do not match. Adjust as indicated by the Evaluator results and repeat the evaluation.

  When red text is displayed, hover to review information about the record. For example, a record that has been deactivated, and so it produces no match code and thus no match score.

4. Click **OK** to save and display the configuration in the Match Code Filter flipper.



ID	Match Code Filter	Comment
filter	Table Match Code Filter: Match Code Anon Values, EMAIL#	

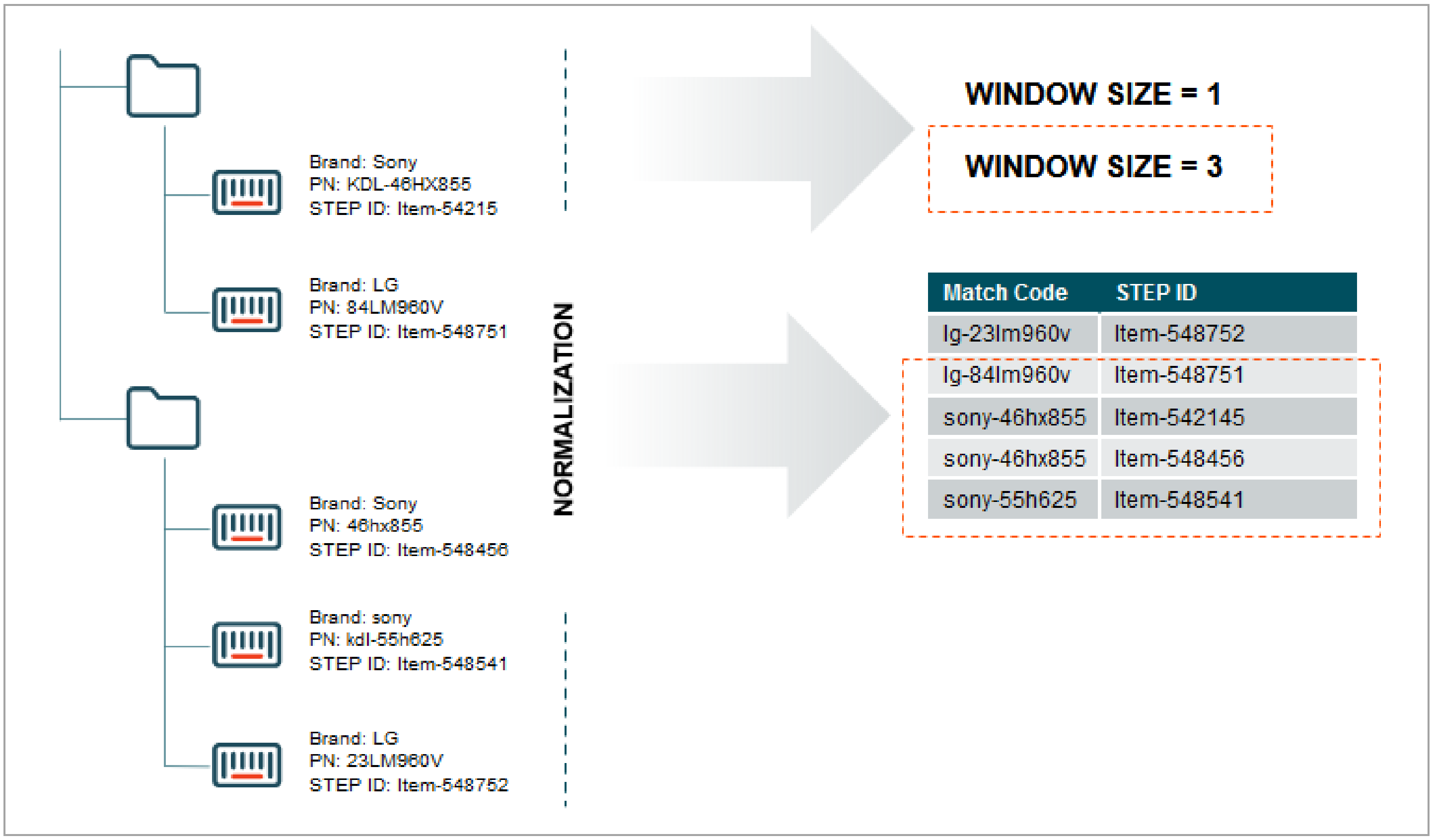
Legacy Match Criteria Without Embedded Match Codes

Match codes defined outside the matching algorithm are legacy functionality but are still supported.

Note: External Match Codes can only be used for matching algorithms that have been created without the Embed Match Code checkbox selected. It is recommended to use the process described in the **Configuring Matching Algorithms** topic.

Window Size

The window size option on legacy external match codes allows configuring match codes to include near-matches. For example, with a window size of '3,' Item-548456 is compared to the object with the match code immediately prior to and the match code immediately following it in the list.



Brand: Sony
PN: KDL-46HX855
STEP ID: Item-54215

Brand: LG
PN: 84LM960V
STEP ID: Item-548751

Brand: Sony
PN: 46hx855
STEP ID: Item-548456

Brand: sony
PN: kdl-55h625
STEP ID: Item-548541

Brand: LG
PN: 23LM960V
STEP ID: Item-548752

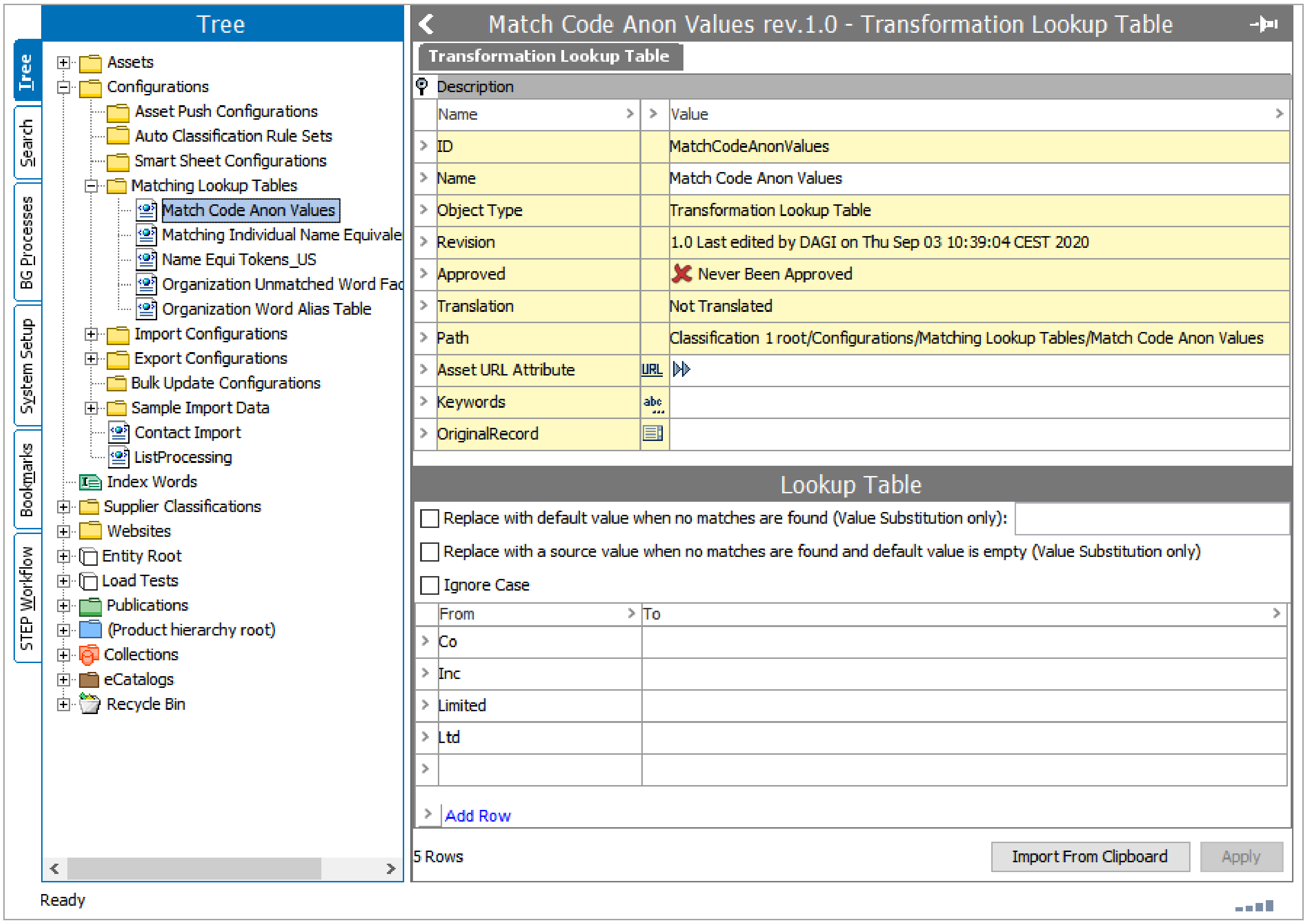
Match Code	STEP ID
lg-23lm960v	Item-548752
lg-84lm960v	Item-548751
sony-46hx855	Item-542145
sony-46hx855	Item-548456
sony-55h625	Item-548541

Configuring a Legacy External Match Code

Match codes defined outside the matching algorithm are legacy functionality but are still supported.

To create an external match code:

1. Create a new transformation lookup table and enter all the match codes to exclude in the 'from' column. Leave the 'to' column empty.



Match Code Anon Values rev.1.0 - Transformation Lookup Table

Transformation Lookup Table

Description

Name	Value
ID	MatchCodeAnonValues
Name	Match Code Anon Values
Object Type	Transformation Lookup Table
Revision	1.0 Last edited by DAGI on Thu Sep 03 10:39:04 CEST 2020
Approved	✗ Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Configurations/Matching Lookup Tables/Match Code Anon Values
Asset URL Attribute	
Keywords	
OriginalRecord	

Lookup Table

☐ Replace with default value when no matches are found (Value Substitution only):

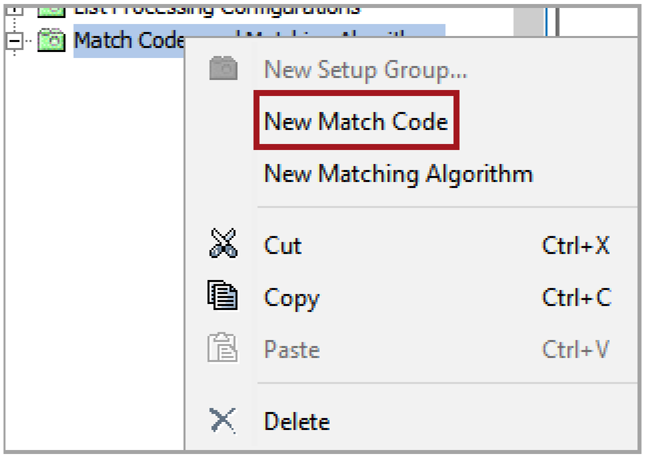
☐ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

☐ Ignore Case

From	To
Co	
Inc	
Limited	
Ltd	

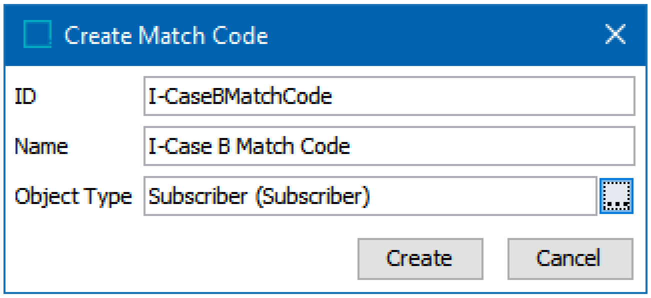
2. In System Setup, open the matching algorithm for the match code filter.

3. In System Setup, right-click the node configured to house match codes and select **New Match Code**.



4. In System Setup, open the matching algorithm for the match code filter.

5. In the Create Match Code dialog, add an **ID** a **Name**, an **Object Type**, and click **Create**. Additional object types can be identified in the Match Code editor after creation.

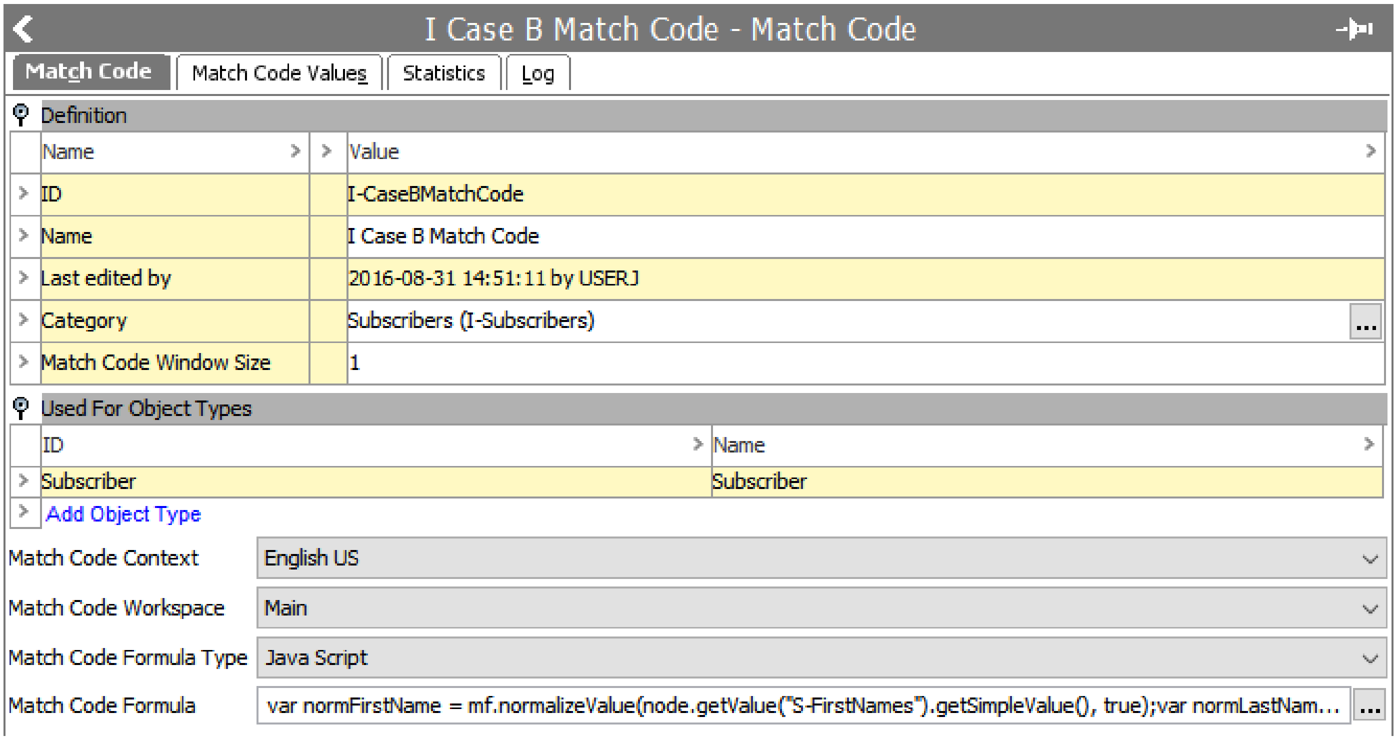


ID: I-CaseBMatchCode

Name: I-Case B Match Code

Object Type: Subscriber (Subscriber)

6. On the new match code editor, navigate to the Match Code tab and click the ellipsis button (...) in the Category field. In the dialog, select a node to indicate which objects will have match codes generated.



Definition

Name	Value
ID	I-CaseBMatchCode
Name	I Case B Match Code
Last edited by	2016-08-31 14:51:11 by USERJ
Category	Subscribers (I-Subscribers)
Match Code Window Size	1

Used For Object Types

ID	Name
Subscriber	Subscriber

Add Object Type

Match Code Context: English US

Match Code Workspace: Main

Match Code Formula Type: Java Script

Match Code Formula: var normFirstName = mf.normalizeValue(node.getValue("S-FirstNames").getSimpleValue(), true);var normLastNam...

7. In the **Match Code Window Size** parameter, specify the window size to be used by the matching algorithm. Refer to the **Window Size** section above for details.

8. If additional object types are required, in the Used For Object Types flipper, click the **Add Object Type** link and choose additional object types for the match code.

9. In the **Match Code Context** parameter, if the data is dimension dependent, specify the context to run the match code formula. By default, the current context is selected.

10. In the **Match Code Workspace** parameter, specify the workspace to run the match code formula. By default, Main workspace is selected.

11. In the **Match Code Formula Type** parameter, specify JavaScript or Calculated as the format. This selection determines the dialog display by the Match Code Formula parameter.

12. In the **Match Code Formula** parameter, click the ellipsis button (**...**) to open the formula editor and add your match code formula. Refer to the following **JavaScript Formula Type** and **Calculated Formula Type** sections for details about the selected formula type.

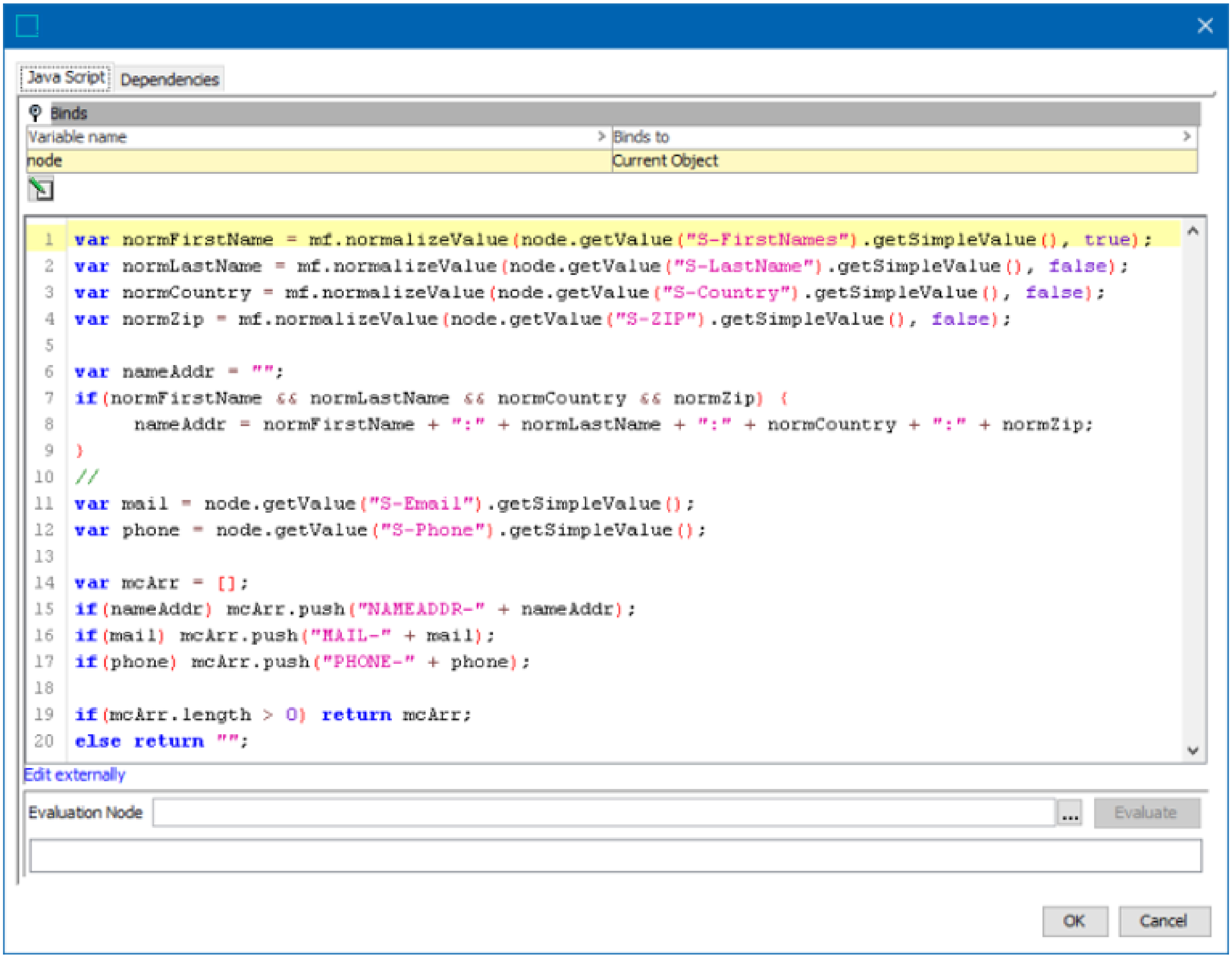
JavaScript Formula Type

The following elements and methods are available for a JavaScript formula:

- **Binds -** On the JavaScript tab, to add binds, open the Binds flipper and click the **Edit** button to display the Edit Binds dialog. Binds give the match code formula access to attributes and values that are created offline for offline matching or matching records on import. Declare variables and bind them to a STEP element or object as determined by the selected formula type. For more information, refer to the **JavaScript Binds** topic in the online help **Resource Materials** documentation.

- **JavaScript -** Bind the current object to a variable. The goal should be to return the match code value of an object from the JavaScript. If a string is returned, it is used as a match code value. If a JavaScript array is returned, all values in the array are used as match code values for that object. Additional utility functions for match codes can be accessed by binding Matching Functions to the context variable in JavaScript, for example, or by binding 'Lookup Table Home' to 'lth.' For more information, refer to the **Text Functions** topic in the online help **Resource Materials** documentation.

Method	Description
`context.soundex('Stibo')`	Returns the Soundex.
`context.metaphone3('Stibo')`	Returns the primary value for the Metaphone 3.
`context.metaphone3alternate('Stibo')`	Returns the alternate value for the Metaphone 3.
`lth.getLookupTableValue('<asset-id>', 'LookupValue')`	For more information, refer to the **Transformation Lookup Tables** topic in the online help **Resource Materials** documentation.



Variable name	Binds to
node	Current Object

```
var normFirstName = mf.normalizeValue(node.getValue("S-FirstNames").getSimpleValue(), true);
var normLastName = mf.normalizeValue(node.getValue("S-LastName").getSimpleValue(), false);
var normCountry = mf.normalizeValue(node.getValue("S-Country").getSimpleValue(), false);
var normZip = mf.normalizeValue(node.getValue("S-ZIP").getSimpleValue(), false);

var nameAddr = "";
if(normFirstName && normLastName && normCountry && normZip) {
     nameAddr = normFirstName + ":" + normLastName + ":" + normCountry + ":" + normZip;
}
//
var mail = node.getValue("S-Email").getSimpleValue();
var phone = node.getValue("S-Phone").getSimpleValue();

var mcArr = [];
if(nameAddr) mcArr.push("NAMEADDR-" + nameAddr);
if(mail) mcArr.push("MAIL-" + mail);
if(phone) mcArr.push("PHONE-" + phone);

if(mcArr.length > 0) return mcArr;
else return "";
```

Calculated Formula Type

When defining the formula via the calculated attribute language, all functions are available. An object's match code value can be a single string derived from the value of the formula or it can be a list where all the values in the list are used as match code values for that object.

Below is an example of a simple STEP Function.

The match code value for each object is a concatenation of the value for a Manufacturer attribute, the string ':' and the value for a ManufacturerPartNumber attribute. The Manufacturer value is normalized via a transformation lookup table with ID 'ManufacturerNormalization.'

```
concatenate(
    replacevaluebylookup("ManufacturerNormalization", value("Manufacturer")),
":",
    value("ManufacturerPartNumber")
)
```

Alternatively, to return two match code values for each object, one for the Manufacturer and one for Manufacturer Part Number, each prefixed with either 'MAN-' or 'MPN-' follow this example, which has no normalization:

```
listconcatenate(
    concatenate("MAN-", value("Manufacturer")),
    concatenate("MPN-", value("ManufacturerPartNumber"))
)
```

The prefix makes it possible avoid comparing objects with match code values from completely different domains.

Notice that in these examples only rudimentary normalization is applied, and missing values are not handled. Matching code values that only consist of the hardcoded prefixes is not beneficial, so checking for empty values is added to the last example below.

```
{
    man:= value("Manufacturer"),
    mpn:= value("ManufacturerPartNumber")
}
listconcatenate(
if(len(man)!=0, concatenate("MAN-", man), ""),
if(len(mpn)!=0, concatenate("MPN-", mpn), "")
)
```

Customizing Match Criteria with JavaScript Functions

Many cases require expanding on the existing normalizers or matchers with functionality specific to the dataset and sources at hand. Match criteria can be expanded using JavaScript business functions and JavaScript functions support this implementation.

Below are example normalizers and matchers implemented in JavaScript to showcase some of the available tools. These functions can be used for both pure JavaScript matching algorithms and JavaScript in decision tables.

> **Important:** The below functions are examples and likely cannot be used in their current form for your business case. Test thoroughly with your own data before implementing in your production STEP system.

normalizeValue

The `normalizeValue` function uses JavaScript and regular expressions to make a text lowercase and leave only letters and digits characters.

```
function normalizeValue(value) {
  if(value) {
    var normVal = value + "";
    normVal = normVal.toLowerCase();
    normVal = normVal.replace(/[^\w]|_/g, "");
    return normVal;
  }
  else {
    return "";
  }
}
```

normalizeStreet

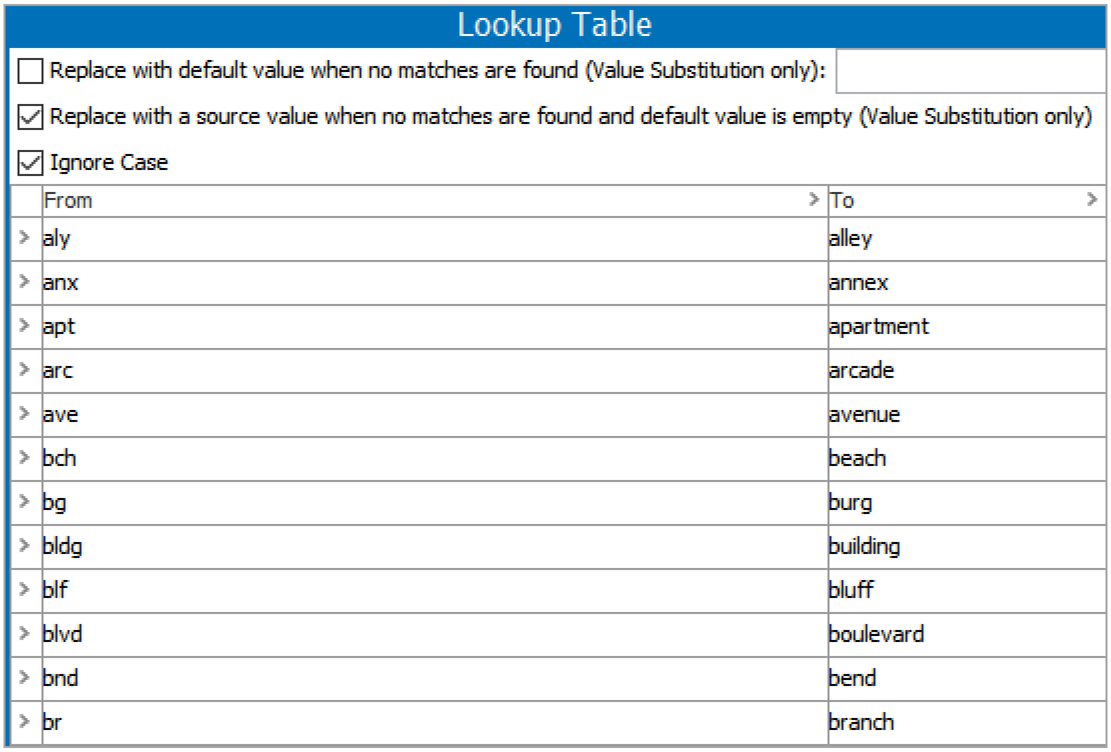
This example demonstrates how to access lookup tables. For more information on lookup tables, refer to the **Transformation Lookup Tables** topic in online help **Resource Materials** documentation.

The `normalizeStreet` function applies basic normalization to 'Street' values and uses a transformation lookup table with ID 'AddressAbbreviations' to replace common abbreviations like 'rd,' 'ave,' and 'ap' with their full-word counterpart.

```
function normalizeStreet(input, lookupTableHome) {
  var output = "";
  if(input) {
    input = input + "";
    input = input.toLowerCase();
    input = input.replace(/[\.\,#]|_/g, "");
    var inArr = input.split(" ");
    var outArr = [];
    for(var i = 0; i < inArr.length; i++) {
      outArr.push(lookupTableHome.getLookupTableValue("AddressAbbreviations", inArr[i]));
    }
    for(var j = 0; j < outArr.length; j++) {
      output += outArr[j];
      if(j != outArr.length - 1) {
        output += " ";
      }
    }
  }
  return output;
}
```

The logic reads:

- Convert input to JavaScript string,
- Convert to lowercase,
- Remove all instances of (.), (,), and (#) (more characters may be removed, but be careful removing dashes if used in street number ranges),
- Split the string by space characters and loop through the array of words applying the lookup table,
- Piece together the string again and return it.



Lookup Table

☐ Replace with default value when no matches are found (Value Substitution only):

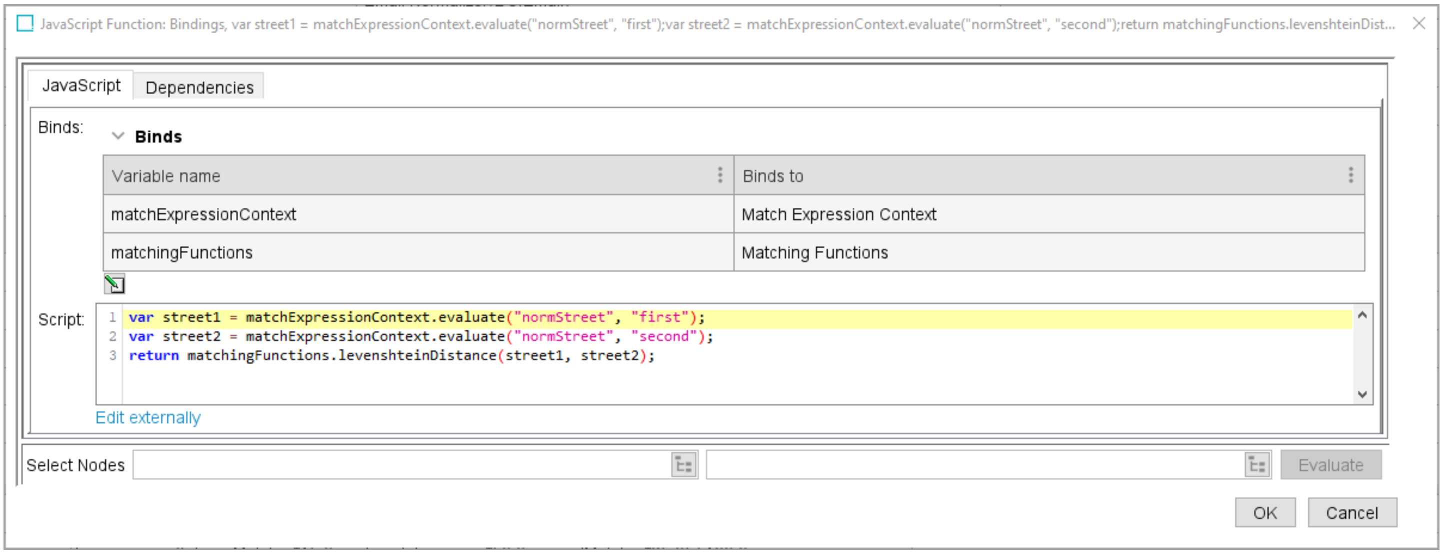
☑ Replace with a source value when no matches are found and default value is empty (Value Substitution only)

☑ Ignore Case

From	To
aly	alley
anx	annex
apt	apartment
arc	arcade
ave	avenue
bch	beach
bg	burg
bldg	building
blf	bluff
blvd	boulevard
bnd	bend
br	branch

Core Matching Functions

The example below uses the built-in levenshteinDistance function to get the edit distance between normalized street values. 'Matching Functions' is bound to 'matchingFunctions.'



JavaScript Function: Bindings, var street1 = matchExpressionContext.evaluate("normStreet", "first");var street2 = matchExpressionContext.evaluate("normStreet", "second");return matchingFunctions.levenshteinDist...

Binds:

Binds

Variable name	Binds to
matchExpressionContext	Match Expression Context
matchingFunctions	Matching Functions

Script:

```
var street1 = matchExpressionContext.evaluate("normStreet", "first");
var street2 = matchExpressionContext.evaluate("normStreet", "second");
return matchingFunctions.levenshteinDistance(street1, street2);
```

```
    var street1 = matchExpressionContext.evaluate("normStreet", "first");
    var street2 = matchExpressionContext.evaluate("normStreet", "second");
    return matchingFunctions.levenshteinDistance(street1, street2);
```

Legacy Match Criteria Options

Match codes defined outside the matching algorithm are legacy functionality but are still supported.

The following are supported legacy alternatives to decision tables. They are available when Match Criteria is presented as a flipper on the Matching Algorithm tab, instead of the decision table option which is available on a Match Criteria tab.

Important: These match criteria cannot be used by a matching algorithm with embedded match codes.

String Comparison Algorithms

While developing a matching, linking, and merging strategy, a string comparison algorithm can serve as the foundation for the matching process. The available string comparison algorithms include:

- **Levenshtein distance** – A metric for how many edits (substitution, insertion, deletion) it takes to make one string look like another. For example, the Levenshtein distance between the strings 'AXR55487' and '8XRT5487' is 2 because the first and fourth digits are different. In STEP terms, the strings would be 75 percent alike (6/8*100).
- **Damerau-Levenshtein distance** – Like the Levenshtein distance except that the transposition of two adjacent characters counts as one edit, not two. For example, the Levenshtein distance between the strings 'AA67' and 'A6A7' is 2 while the Damerau Levenshtein distance is 1.
- **Jaro / Jaro-Winkler distance** – Outputs 0 or 1 where 0 is no similarity and 1 an exact match. These algorithms are available and can be made accessible in STEP via JavaScript but are not included in the STEP core.

Note: The Levenshtein / Damerau-Levenshtein distance must be manually converted into a percentage.

When the preferred string comparison algorithm is insufficient, it is possible to apply the Levenshtein / Damerau-Levenshtein distance directly to strings built using STEP functions and automatically output an equality metric. Several criteria can be added and assigned weights to calculate the total equality. The available criterion types are described as follows.

Multi Word Damerau-Levenshtein Distance

The Multi Word Damerau-Levenshtein distance is equal to the Damerau-Levenshtein distance except that the transposition of two words does not count as an edit. For example, the distance between 'Paul Johnson' and 'Johnson Paul' is 0. This criterion is useful when working with names where first name and surname are in the same attribute value, yet the order differs between objects.

Number Distance

The Number Distance criterion returns the relative distance between two numbers expressed as a percentage: lowest number / highest number * 100. This is a simplistic way of calculating a difference. For example, the numbers 1 and 2 will be as different or equal as 50 and 100.

Special cases:

- If one or both strings are not numerical values, the criterion returns '0.'
- If only one of the strings is '0,' the criterion returns '0.'
- If both strings are '0,' the criterion returns '100.'
- If both strings are negative the calculation is the highest number / lowest number * 100.
- If one value is positive and the other negative, the criterion returns '0.'

Use STEP functions to generate the data that requires the number distance calculation.

JavaScript

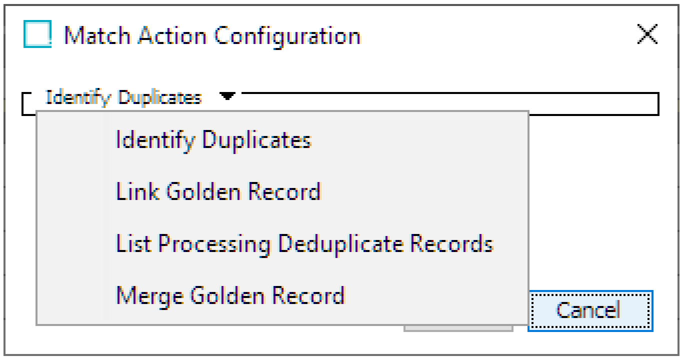
The JavaScript criterion allows you to define your own algorithm for comparing objects. The only requirement is that the result is a number between 0 and 100 to represent the percentage of equality.

From the JavaScript criterion, use functions defined in business libraries in addition to the objects made available via bindings.

For more information, refer to the **JavaScript Binds** topic of the online help **Resource Materials** documentation.

Match Actions

This page assumes you have read and understood how match actions fit into the bigger picture of a matching, linking, and merging solution. For more information, refer to the **Matching, Linking, and Merging** topic.



The choice of match action defines the entire workflow around the golden records. The following match actions exist and are paired with the following component models and matching functionality:

Match Action	Component Model(s)	Matching Functionality
Identify Duplicates	Matching	Identify Duplicates
Link Golden Record	Matching Matching - Link Golden Record	Match and Link
Link Processing Deduplicate Records	List Processing	List Processing Deduplicate Records
Merge Golden Record	Matching Matching - Merge Golden Record	Match and Merge

Identify Duplicates

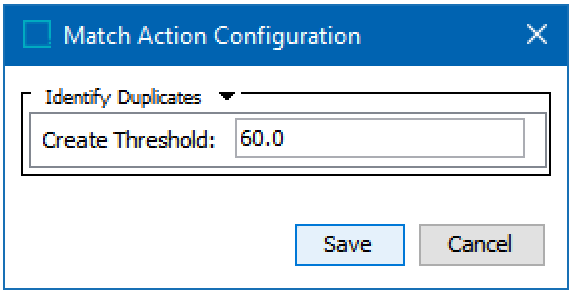
The Identify Duplicates match action helps determine if duplicates exist in a dataset and allows users to manually confirm, reject, merge, and delete duplicates with limited impact on existing functionality.

> **Note:** A matching algorithm using the Identify Duplicates match action only links records. While it is possible to set up workflows and UIs for manually merging the identified duplicate records in STEP, if those actions are needed, the Identify Duplicates match action is probably not the best choice. For match actions with configurable automatic actions, refer to the **Match and Link** or **Match and Merge** topics.

With the Identify Duplicates match action, as matchable objects are created and modified, events are sent to a matching event processor. In an asynchronous process, the Match Event Processor matches these objects with other matchable objects, as defined by the matching algorithm. When two objects score above the create threshold, a match result is stored for future handling.

Configuration

The Create Threshold parameter is required for the Identify Duplicates match action and specifies 'how equal' objects must be to be marked as possible duplicates.



Create Threshold: 60.0

> **Note:** Identify duplicates uses many of the same workbench and Web UI tools as the match and link match action.

Identify Duplicates in Workbench

For information, refer to the **Match and Link in Workbench** topic.

Identify Duplicates in Web UI

The Web UI supports actions on identified duplicates, as defined in these topics:

- **Potential Duplicates List** topic
- **Merging Confirmed Matches** topic
- **Confirmed Matches Component** topic.

Update Match Scores on Save in Web UI

The global configuration 'Matching on Save Configuration' holds a list of identifying matching algorithms that will be used to calculate match codes and rank scores during a save operation in the Web UI.

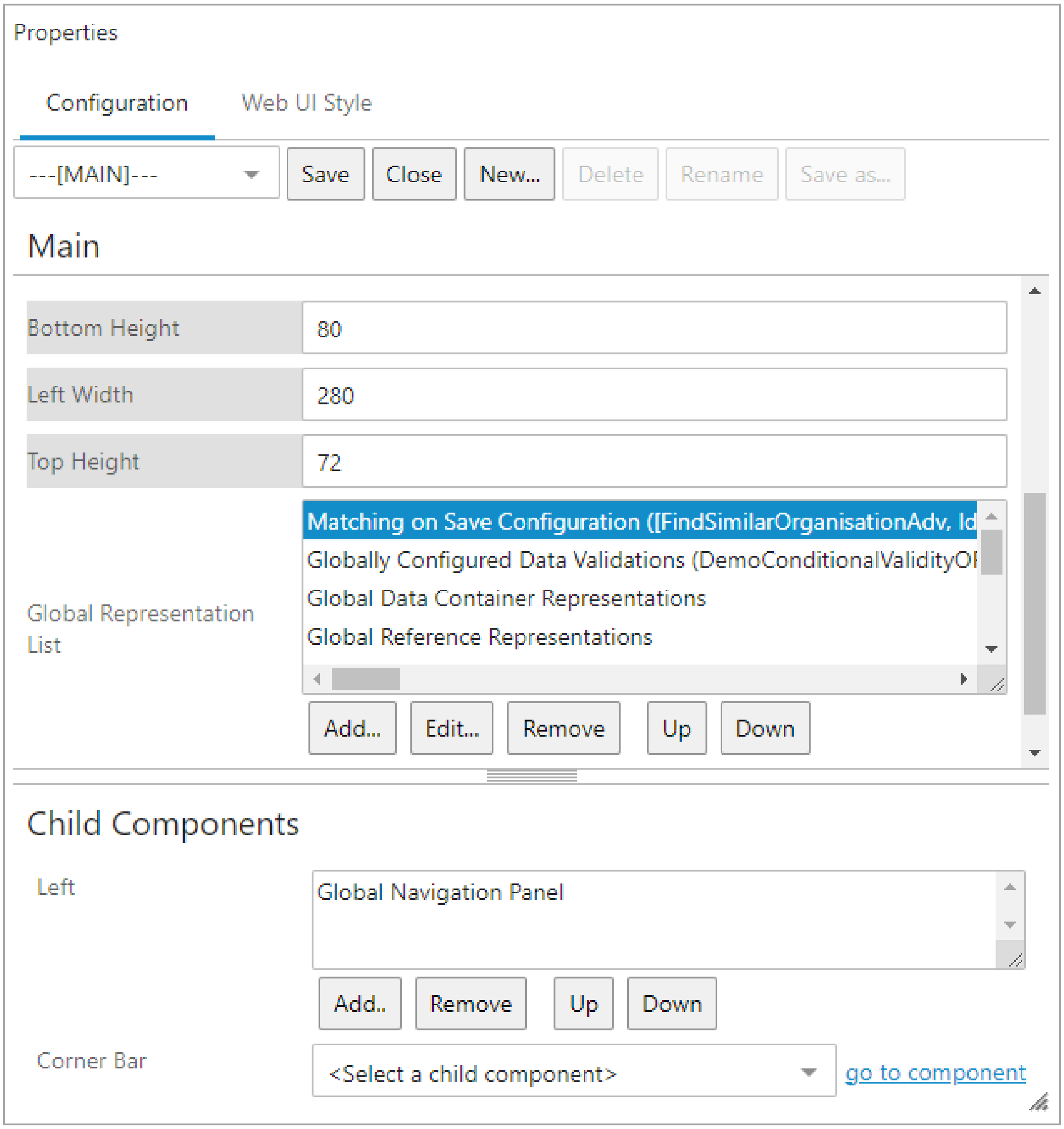
> **Note:** Only identifying matching algorithms are valid. The use of non identifying matching algorithms will result in an error.

For further information on the use of matching algorithms, refer to the **Matching Algorithms and Match Expressions** topic in the **Matching, Linking, and Merging** documentation.

'Matching on Save Configuration' is available on The 'Global Representation List' in the Web UI. For further information on the Global Representation List, refer to the **Main Properties** topic in the **Web User Interfaces** documentation.

To configure 'Matching on Save Configuration', log in to Web UI Design Mode:

1. In the designer window, select the 'MAIN' screen.
2. Go to the 'Global Representation List' parameter.



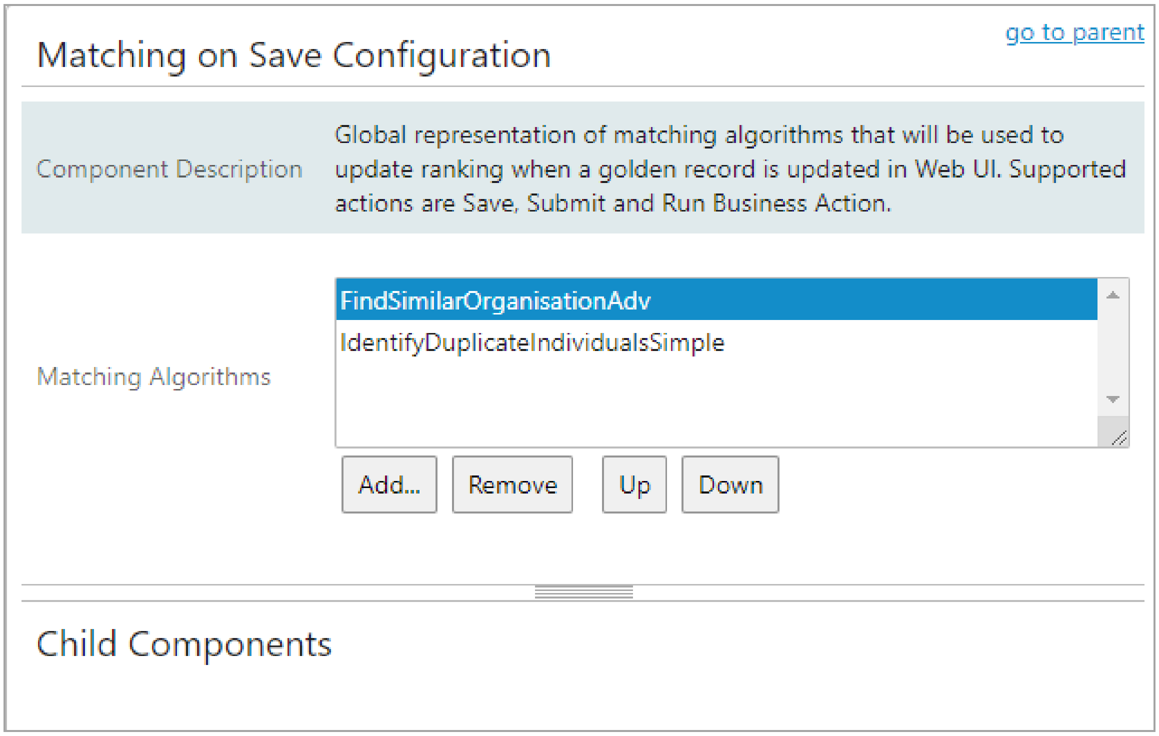
Main

Bottom Height	80
Left Width	280
Top Height	72
Global Representation List	Matching on Save Configuration ([FindSimilarOrganisationAdv, Id Globally Configured Data Validations (DemoConditionalValidityOf Global Data Container Representations Global Reference Representations

Child Components

Left	Global Navigation Panel
Corner Bar	<Select a child component> go to component

In the below example, matching algorithms for 'FindSimilarOrganisationAdv' and 'IdentifyDuplicateIndividualsSimple' have been added to the global configuration. When the user saves new object types in a 'Find Similar in Workflows' process, the configuration calculates match scores for potential duplicates. For further information, refer to the **Find Similar in Workflows** topic in the **Matching, Linking, and Merging** documentation.



Matching on Save Configuration

Component Description: Global representation of matching algorithms that will be used to update ranking when a golden record is updated in Web UI. Supported actions are Save, Submit and Run Business Action.

Matching Algorithms:
- FindSimilarOrganisationAdv
- IdentifyDuplicateIndividualsSimple

Child Components

Match and Link

Using an asynchronous process, Match and Link creates and maintains a set of 'golden records' as an aggregation of matching 'source records'.

- In Product MDM, Match and Link is commonly used in automating the creation and maintenance of sell-side products as golden records, based on buy-side products as source records.
- In Customer MDM, Match and Link is commonly used for resolving household entities as golden records using individual customer entities as source records.

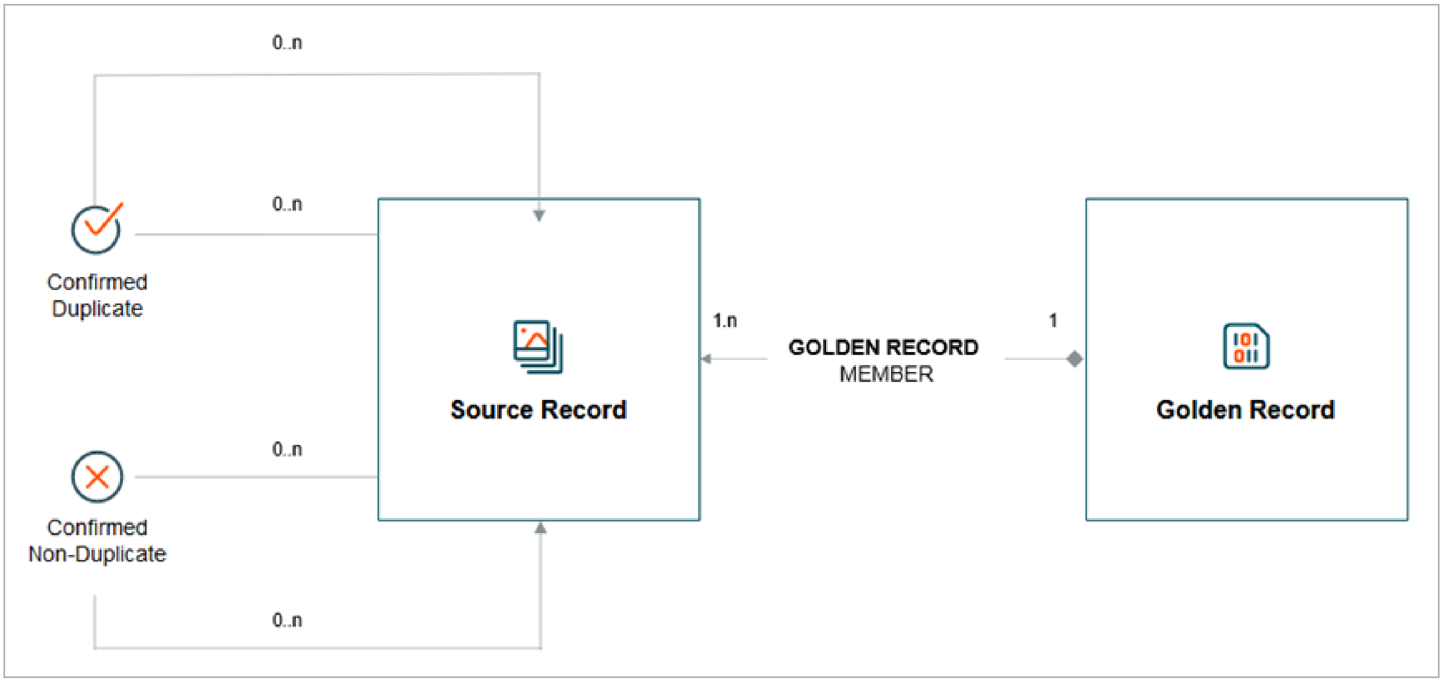
A detailed setup using Match and Link is described in the **Accelerator for Retail Data Onboarding** topic of the **Accelerator for Retail** section of the **Product MDM Solution Enablement** documentation.

For details about the use of Match and Link in household entities, refer to the **Algorithm & Match Codes - Household** topic in the **Solution Enablement: Customer & Supplier MDM** documentation.

Data Model

In a Match and Link solution, source records and golden records will be separate records of different object types.

The golden records are created by survivorship rules, and every source record belongs to exactly one golden record.



Confirming a duplicate or non-duplicate in a Match and Link solution results in a reference being created on the source record level. In the Match and Link solution, the Confirmed Duplicate is a reference between two source records which permanently identifies two specific source records as duplicates. The Confirmed Non-Duplicate is the opposite, permanently confirming that two source records should never belong to the same golden record object.

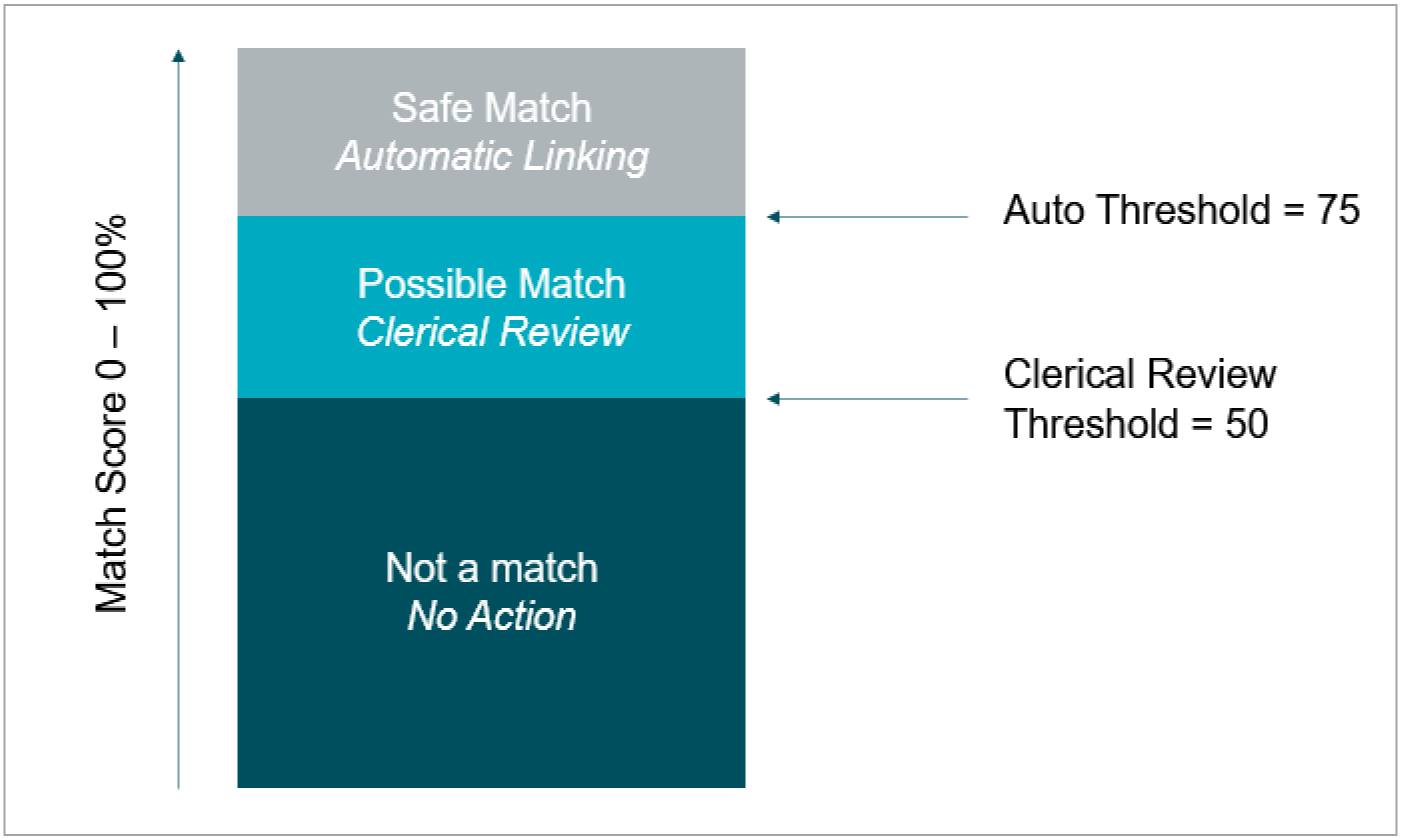
Match Score

In a link solution, thresholds determine if records can be automatically linked or if manual review is required. The match score (also called the 'rank score' in Web UI) is the percentage of equality between the two records being compared as potential duplicates. Configuring a linking solution includes setting thresholds to determine the required percentage of equality for records to be linked.

- The **Auto Threshold** is the equality percentage for automatic linking. Two source objects that meet the defined percentage are automatically linked to the same golden record.
- The **Clerical Review Threshold** is the equality percentage equal to or below the Auto Threshold setting that triggers a manual review. Two objects that are within this range are sent to the clerical review workflow to be manually reviewed as potential duplicates.

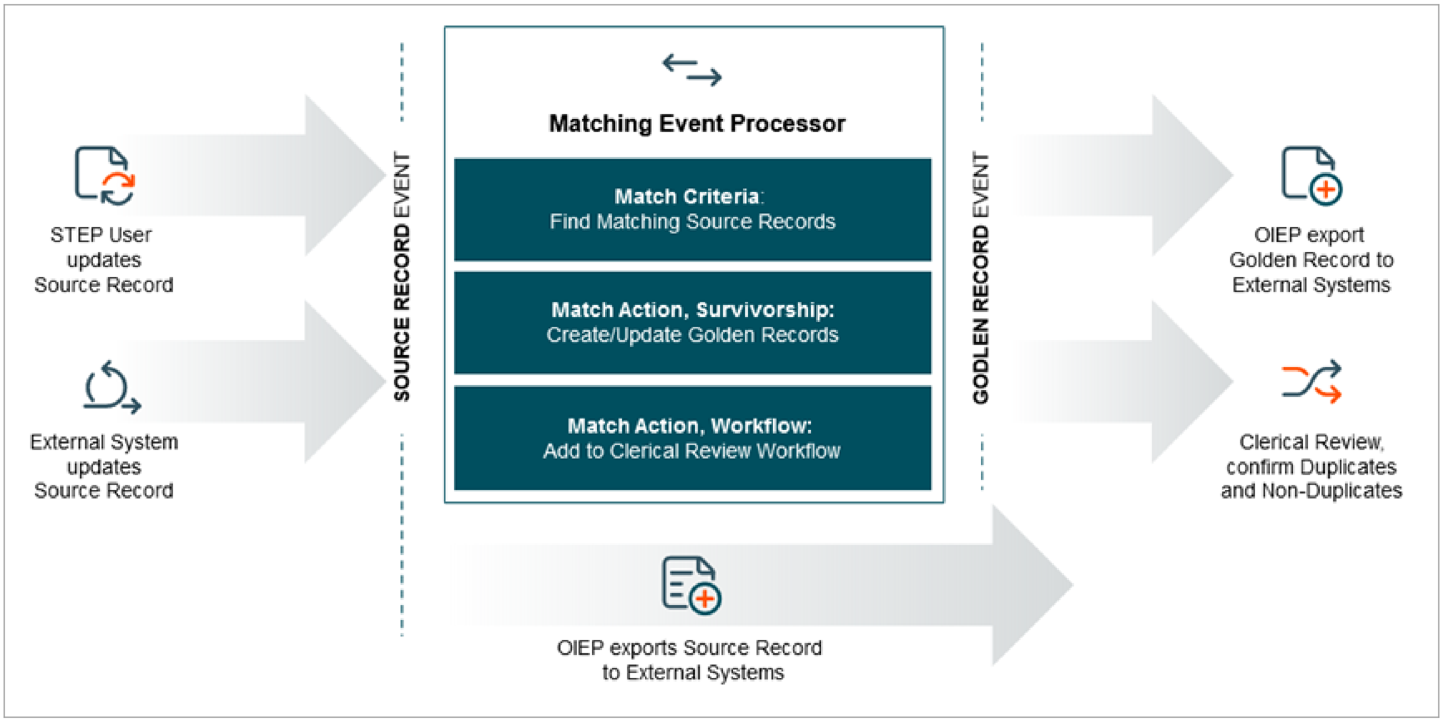
  Potential duplicates enter the selected clerical review workflow where a user then sets one of the following reference types:

  - Confirm Duplicate - A user manually confirms the records are duplicates. The duplicate source records are linked together by a 'Confirmed Duplicate' reference and will remain part of the same golden record from that point.
  - Reject Duplicate - A user manually rejects the records as duplicates. The source records are linked by a 'Confirmed non-duplicate' reference and will never again be made part of the same golden record.



Information Flow

When a user or a source system updates a source record, events are written to a Matching event processor. The Matching event processor lets the matching algorithm run a match on the source record against all existing source records that share a match code.


Matching Event Processor
Match Criteria: Find Matching Source Records
Match Action, Survivorship: Create/Update Golden Records
Match Action, Workflow: Add to Clerical Review Workflow
STEP User updates Source Record
External System updates Source Record
SOURCE RECORD EVENT
GODLEN RECORD EVENT
OIEP export Golden Record to External Systems
Clerical Review, confirm Duplicates and Non-Duplicates
OIEP exports Source Record to External Systems

Source records with a match score above an Auto-Link Threshold will be linked to the same golden record. The golden record will be updated with information from all linked source records, according to a set of survivorship rules. For more information, refer to **Survivorship in Match and Link** topic. The resulting golden record updates can trigger events that export the golden record to external systems.

Records with match scores between the Auto-Link Threshold and the Clerical Review Threshold are added to a Clerical Review Workflow. This allows a data steward user to manually identify if this is a Confirmed Duplicate or a Confirmed Non-Duplicate. A decision by the data steward is considered an update to the source record and can invoke the flow again depending on triggering events on the Matching event processor.

The golden record in a match and link solution should be considered a system-owned object. Users should not perform manual updates to the golden record since survivorship rules overwrite this information and the golden record may be deleted by the Matching event processor.

It is common to enrich golden records with information through an additional 'internal data' source record (sometimes referred to as a 'silver record' or an 'enrichment record') that is created and maintained in association to the golden record.

Information from an internal data source record is promoted to the golden record with survivorship rules by the Matching event processor.

Internal Data Source Objects

In Match and Link setups, there is often a need to maintain data on the golden record. Since the golden record is a system-owned object, data maintenance is performed on 'enrichment records' or 'internal data source objects' according to the following rules:

- A unique object type is required, one that is different from the object types of golden record and other source objects.
- Do not generate match codes for internal data source objects.
- In the Matching component model configuration, Source Object Type aspect, add the object type of the internal data source object.
- Golden records should use the same reference types for internal source objects and for other source objects.

To update the golden record automatically when an internal data source object changes:

1. Configure the event processor to listen on events for internal data source objects.
2. Create a business action to find the golden record for the internal data source object, identify one of the other source objects for the golden record, and then generate an event for that object for the event processor.
3. Create an event filter condition that is always false since the original event for the internal data source object will not go onto the queue.

User Actions

Match and Link is supported by a range of tools in workbench and Web UI so the expert user can analyze the results of the matching algorithm and take actions.

The Match and Link specific actions are:

Confirm Duplicates: If two objects are confirmed as duplicates, a reference of the 'Duplicate Reference Type' specified in the component model and in the matching algorithm will be created, the pair will be removed from the 'Match Result' tab, and instead, will show up on the 'Confirmed Duplicates' tab on the matching algorithm.

Confirm Non Duplicates: If two objects are rejected as being duplicates, a reference of the 'Non-Duplicate Reference Type' will be created and the pair will be shown on the 'Confirmed Non Duplicates' tab on the matching algorithm.

It is important to understand that if a pair has been confirmed as duplicate / non-duplicate, the pair will not be considered when the matching algorithm is reapplied, regardless if the data on the objects has changed. The confirmed duplicate / non-duplicate relationship can be updated either via the 'Remove From List' options or by deleting the references.

Manual Merge of source records: If by Identify Duplicates or by 'Link golden record' two source objects are confirmed as duplicates, it is possible to manually merge them into a single object.

Configuring Match and Link

The Match and Link setup uses two component models, an object type for golden records, a matching algorithm with match action and survivorship rules, and an event processor. These elements work together to identify potentially duplicate records and to ultimately provide golden records that hold the best data from your source records.

Prerequisites

1. Complete the one-time setup defined in the **Initial Setup for Matching Algorithms** topic.
2. Configure a matching algorithm, as defined in the **Configuring Matching Algorithms** topic.
3. Complete the one-time setup defined in the **Initial Setup for Match Tuning** topic.
4. Configure a match tuning configuration, as defined in the **Configuring Match Tuning** topic.

Configure a Match and Link Solution

Use the following steps to configure your matching and linking solution.

1. Configure the Matching component model, as defined in the **Configuring Matching Component Model** topic.
2. Configure the Link Golden Record object type, as defined in the **Configuring the Link Golden Record Object Type** topic.
3. Configure the Matching - Link Golden Record component model, as defined in the **Configuring the Match - Link Golden Record Component Model** topic.
4. Configure the match criteria, as defined in the **Match Criteria** topic.
5. Configure the link golden record match action, as defined in the **Configuring the Link Golden Record Match Action** topic.
6. Set up survivorship rules, as defined in the **Survivorship in Match and Link** topic.
7. Set up an event processor, as defined in the **Configure the Link Event Processor** topic.
8. Set up Web UI, as defined in the **Match and Link in Web UI** topic.

For more information on how to optimize the Match and Link configuration, refer to the **Matching and Linking Recommendations** topic in the **System Administration** documentation.

Configuring Matching Component Model

The Matching component model specifies the object types shared by all defined matching types. Other individual matching component models further specify object types for the specific matching being performed, such as the matching defined in the Match and Link topic or the Match and Merge topic.

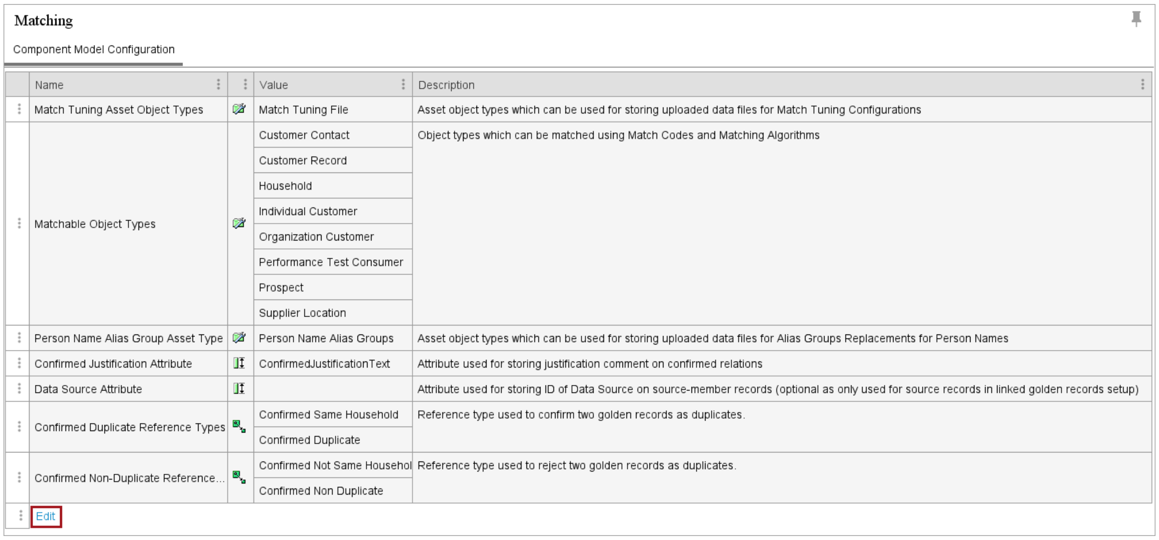
Prerequisites

Create all relevant object types, attributes, and references to make them available for selection in the component model.

Configuration

To configure the component model:

1. In System Setup, open the Component Models node and click the **Matching** component. The Component Model Configuration editor displays the aspects of the matching component.

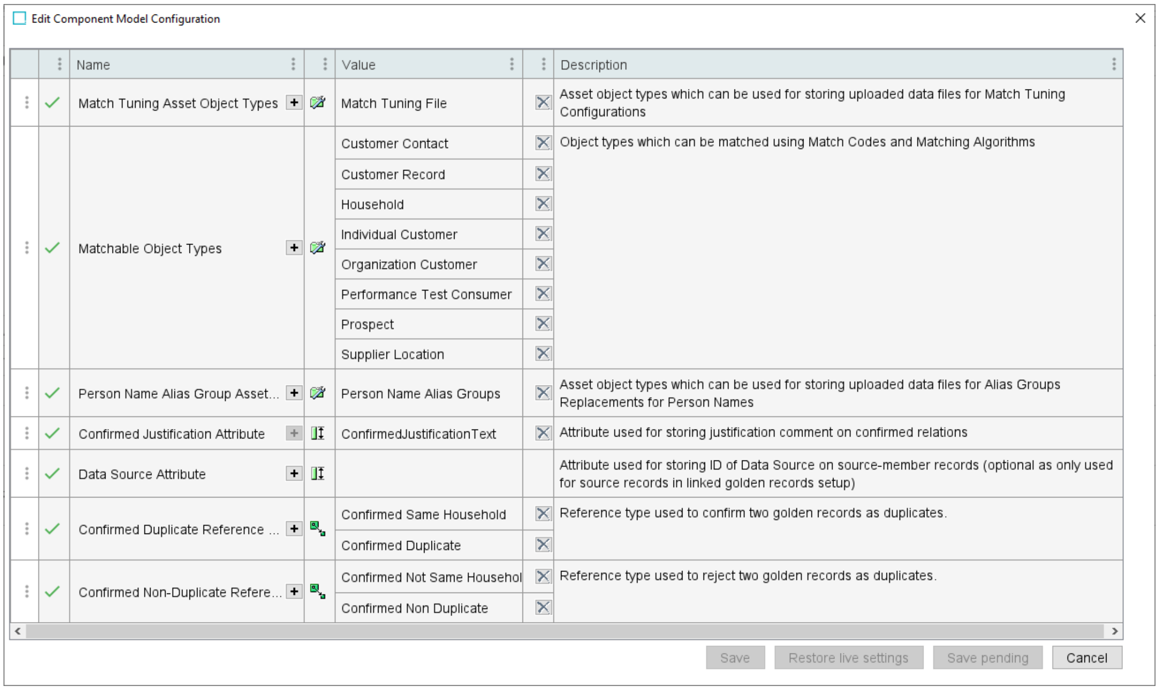


Matching

Component Model Configuration

Name	Value	Description
Match Tuning Asset Object Types	Match Tuning File	Asset object types which can be used for storing uploaded data files for Match Tuning Configurations
Matchable Object Types	Customer Contact Customer Record Household Individual Customer Organization Customer Performance Test Consumer Prospect Supplier Location	Object types which can be matched using Match Codes and Matching Algorithms
Person Name Alias Group Asset Type	Person Name Alias Groups	Asset object types which can be used for storing uploaded data files for Alias Groups Replacements for Person Names
Confirmed Justification Attribute	ConfirmedJustificationText	Attribute used for storing justification comment on confirmed relations
Data Source Attribute		Attribute used for storing ID of Data Source on source-member records (optional as only used for source records in linked golden records setup)
Confirmed Duplicate Reference Types	Confirmed Same Household Confirmed Duplicate	Reference type used to confirm two golden records as duplicates.
Confirmed Non-Duplicate Reference...	Confirmed Not Same Househol Confirmed Non Duplicate	Reference type used to reject two golden records as duplicates.

2. Click the **Edit** link (or the **Edit (pending changes)** link) to display the Edit Component Model Configuration dialog.



	Name	Value	Description
✓	Match Tuning Asset Object Types	Match Tuning File	Asset object types which can be used for storing uploaded data files for Match Tuning Configurations
✓	Matchable Object Types	Customer Contact Customer Record Household Individual Customer Organization Customer Performance Test Consumer Prospect Supplier Location	Object types which can be matched using Match Codes and Matching Algorithms
✓	Person Name Alias Group Asset...	Person Name Alias Groups	Asset object types which can be used for storing uploaded data files for Alias Groups Replacements for Person Names
✓	Confirmed Justification Attribute	ConfirmedJustificationText	Attribute used for storing justification comment on confirmed relations
✓	Data Source Attribute		Attribute used for storing ID of Data Source on source-member records (optional as only used for source records in linked golden records setup)
✓	Confirmed Duplicate Reference ...	Confirmed Same Household Confirmed Duplicate	Reference type used to confirm two golden records as duplicates.
✓	Confirmed Non-Duplicate Refere...	Confirmed Not Same Househol Confirmed Non Duplicate	Reference type used to reject two golden records as duplicates.

To edit an aspect:

- Double click the plus button (⊞) on an aspect to display the 'Select ... for aspect' dialog and select an object type, attribute, or reference type. The button remains active for aspects that allow multiple selections.
- Double click the delete button (☒) to remove a selection.

A green check (✓) means the aspect has no errors; a red X (✗) means additional setup is required. Hover over the X for additional information.

3. For each of the following aspects choose to add object(s), attribute(s), or reference(s), and click the **Select** button.
   - **Match Tuning Asset Object Types** – Select the object types to store the input data for match tuning.
   - **Matchable Object Types** – Select the object types that need to be matched. Only the object types configured can be used as object types for match codes. On objects of these types, the 'Matching' tab is automatically enabled. The 'Matching' tab shows match code values, potential duplicates, and confirmed relations for the selected object.
   - **Person Name Alias Group Asset Type** – Select the asset object types to store uploaded data files for person name alias groups. These asset object types can be used with the Machine Learning Matcher and the Person Name and Address Match Code Generator. For more information, refer to theMatcher: Machine Learning Matcher topic and the Match Code Generator: Person Name and Address topic, both in the Matching, Linking, and Merging documentation.
   - **Confirmed Justification Attribute** – Select a description attribute valid for all reference types specified in the 'Duplicate Reference Types' and 'Non-Duplicate Reference Types' fields. This attribute stores a description explaining why two objects are marked as duplicates or non-duplicates in a match and link solution.

- **Data Source Attribute** – Select one or more description attributes valid for all source object types specified in the 'Source Object Types' field. This attribute contains the source ID of the source objects. If you select more than one attribute in this field, then exactly one of these attributes must be valid per source object type chosen in the 'Source Object Types' field. This field is only required for Link Golden Records solutions with **Trusted Source** survivorship rules configured.
- **Duplicate Reference Types** – Select one or more reference types to store the manually maintained confirmed duplicate references. These references store the reason for confirming two objects as duplicates specified in the attribute selected in the 'Confirmed Justification Attribute' field. All the selected reference types must have exactly one valid attribute from the 'Confirmed Justification Attribute' field. Only the duplicate reference types you select can be used as 'Duplicate Type' on a matching algorithm. In a typical scenario, you will have different duplicate reference types for different matching algorithms. If you reuse duplicate reference type between algorithms, the confirmed duplicates will be reused between those algorithms. Confirmed duplicate references are used in match and link solutions.
- **Non-Duplicate Reference Types** – Select one or more reference types used by the system for storing the manually maintained confirmed non-duplicate references. These references store the reason for confirming two objects as non-duplicates specified in the attribute selected in the 'Confirmed Justification Attribute' field. All the selected reference types must have exactly one valid attribute from the 'Confirmed Justification Attribute' field. Only reference types selected can be used as 'Non-Duplicate Type' on a matching algorithm. In a typical scenario, you will have different duplicate reference types for different matching algorithms. If you reuse the non-duplicate reference type between algorithms, the confirmed non-duplicates will be reused between those algorithms as well.

4. Save or cancel your work:

- Click the **Save** button to save a configuration once it has no errors.
- When enabled, click the **Save pending** button to save your work while errors exist.
- When enabled, click the **Restore live settings** button to undo the changes made to a previously error-free, saved configuration.
- Click the **Cancel** button to undo all changes made in this dialog.

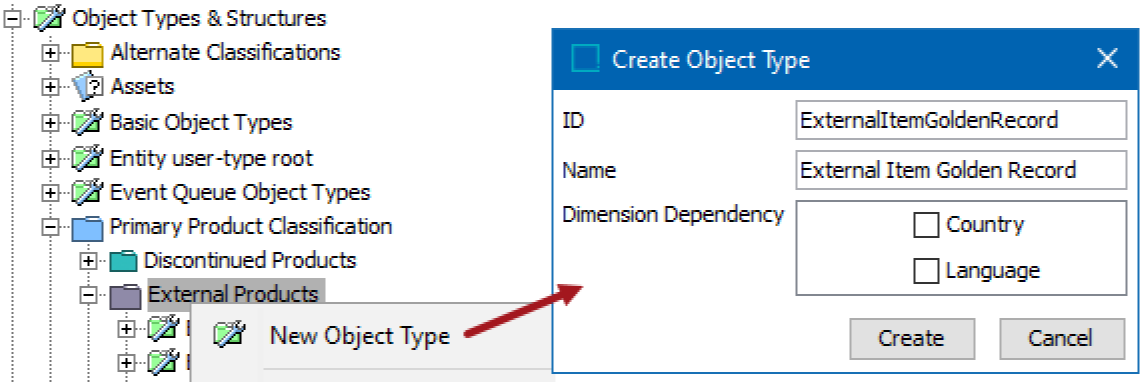
Configuring the 'Link Golden Record' Object Type

The 'link golden record object type' is used by the matching functionality to automatically create 'link golden record' objects. This object type allows golden records to refer back to their source objects.

> **Important:** The 'link golden record' object type must be different from the object type used for source objects.

To create a 'link golden record' object type:

1. In System Setup, open the Object Types & Structure node, right-click on the node that identifies the type of golden record object (product or entity), and select the **New Object Type** option. In this example, the golden record is an 'ExternalItemGoldenRecord' product.
   - Add an **ID** and a **Name**.
   - Set **Dimension Dependency** as necessary.
   - Click the **Create** button.



Create Object Type

ID: ExternalItemGoldenRecord

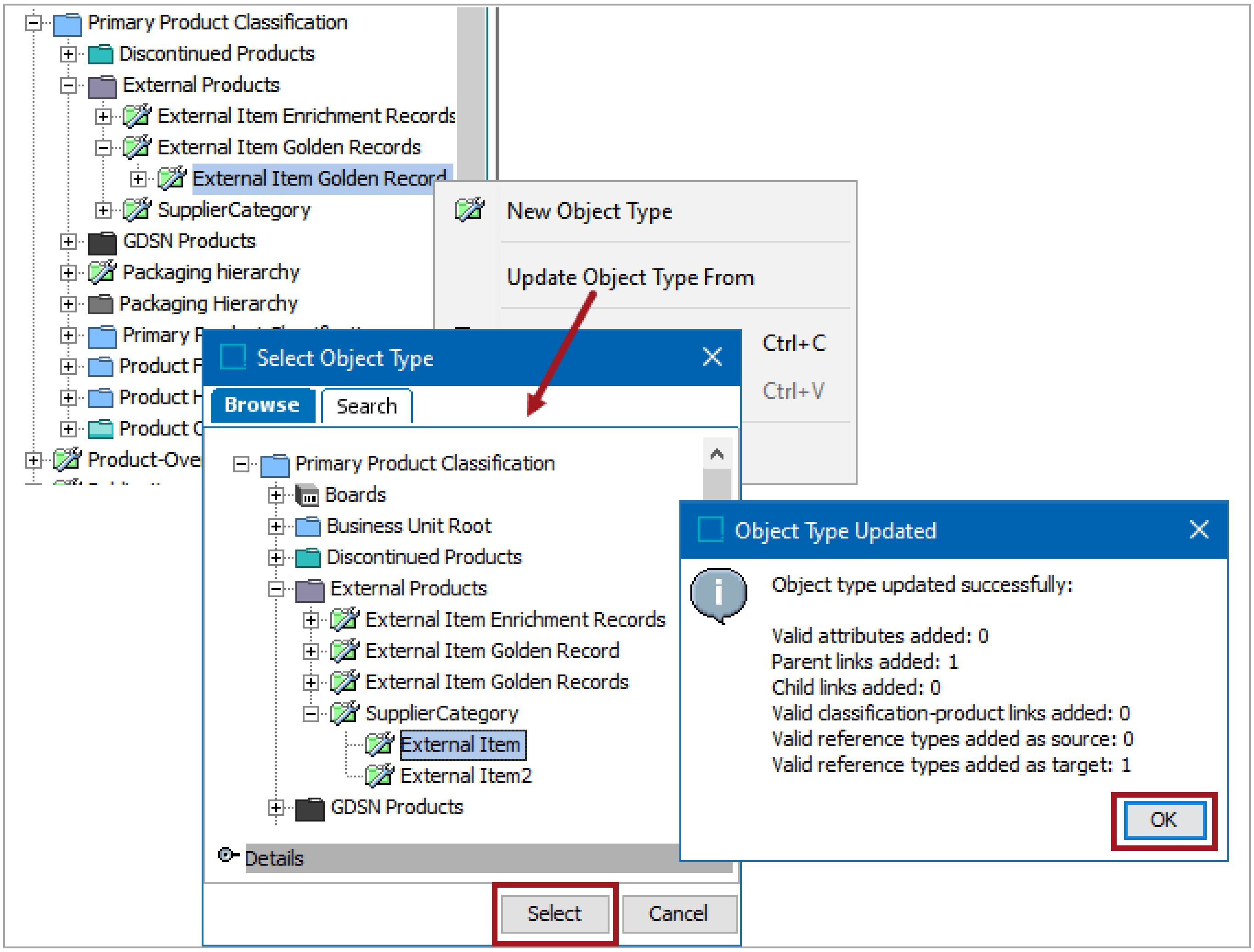
Name: External Item Golden Record

Dimension Dependency: □ Country □ Language

2. If you intend to copy all data from source records, including attribute values and references, ensure the 'link golden record' object type has the same valid attributes and is a valid source for the same reference / link types by using the Update Object Type From option. In this example, 'External Item' is the object type for source records.
   - Right-click the new link golden record and choose the 'Update Object Type From' option.
   - On the 'Select Object Type' dialog, select the source record object type.
   - Click the **Select** button to duplicate validity for attributes and reference / link types from the source record

to the link golden record object type.

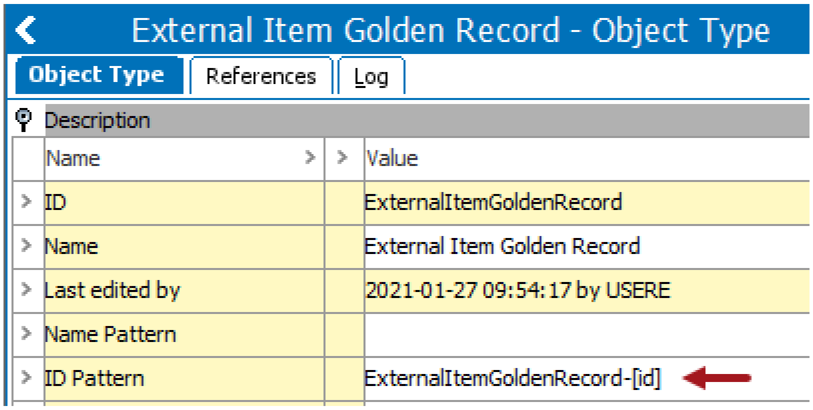
- Click the **OK** button.



Object type updated successfully:

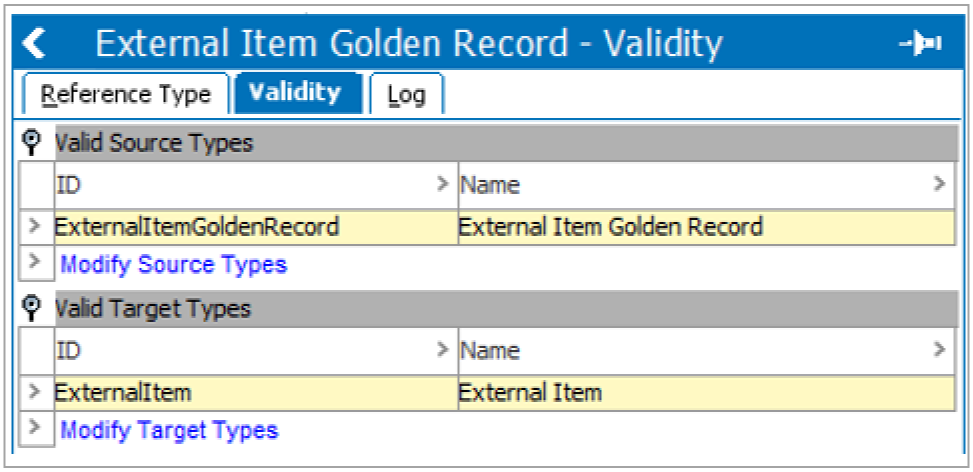
Valid attributes added: 0
Parent links added: 1
Child links added: 0
Valid classification-product links added: 0
Valid reference types added as source: 0
Valid reference types added as target: 1

3. On the Description flipper, set the **ID Pattern** parameter to use the **[id]** variable.



Name	Value
ID	ExternalItemGoldenRecord
Name	External Item Golden Record
Last edited by	2021-01-27 09:54:17 by USERE
Name Pattern	
ID Pattern	ExternalItemGoldenRecord-[id]

4. Verify that the reference type for linking 'source records' with 'link golden records' has the following settings:
   - On the Reference Type tab, set the **Allow multiple references** parameter to 'Yes.'
   - On the Validity tab, under the **Valid Source Types** flipper add the golden record object type (such as ID=ExternalItemGoldenRecord).
   - On the Validity tab, under the **Valid Target Types** flipper add to the source object type (such as ID=ExternalItem).



External Item Golden Record - Validity

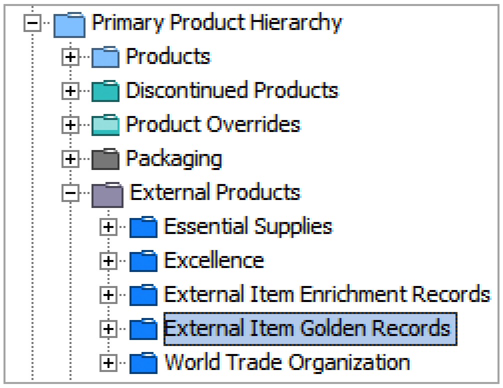
Valid Source Types

ID	Name
ExternalItemGoldenRecord	External Item Golden Record

Valid Target Types

ID	Name
ExternalItem	External Item

5. In Tree, create a root node for the link golden records. Initially, all link golden records will be created as children of this node.



- Primary Product Hierarchy
  - Products
  - Discontinued Products
  - Product Overrides
  - Packaging
  - External Products
    - Essential Supplies
    - Excellence
    - External Item Enrichment Records
    - External Item Golden Records
    - World Trade Organization

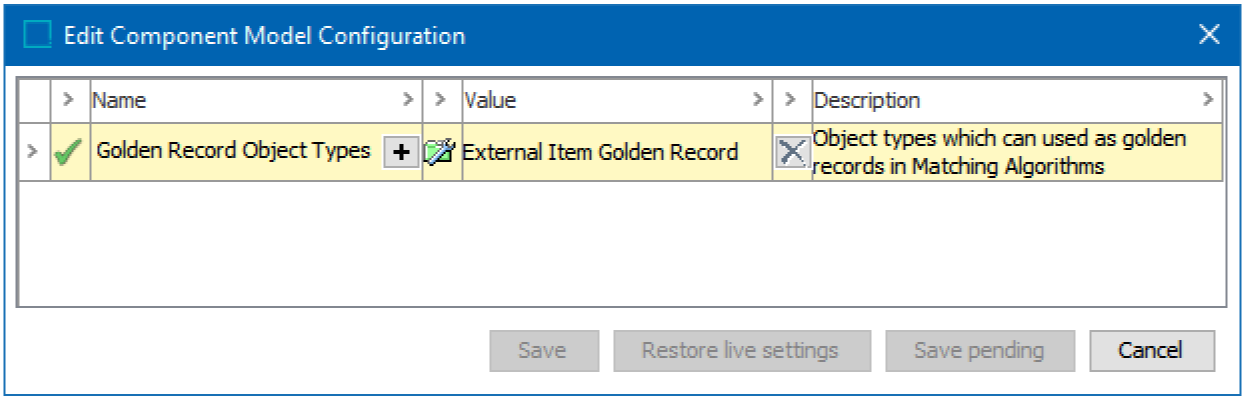
Configuring the Matching - Link Golden Record Component Model

The 'Matching - Link Golden Record' component model identifies all the golden record object types applicable to the link golden record solution and enables Match and Link functionality.

Important: Only the object types added to the component mode can be used as golden records for link golden record configurations. On objects of these object types, the Golden Record tab is automatically enabled and displays the golden record together with its member records.

To configure the component model:

1. In System Setup, expand the 'Component Models' node and select the **Matching - Link Golden Record** node.
2. On the 'Component Model Configuration' tab, click the **Edit** link (or the **Edit (pending changes)** link) to display the 'Edit Component Model Configuration' dialog.



Edit Component Model Configuration

	Name	Value	Description
✓	Golden Record Object Types	External Item Golden Record	Object types which can used as golden records in Matching Algorithms

   - Double click the plus button (+) on an aspect to display the 'Select ... for aspect' dialog and select an object type, attribute, or reference type. The button remains active for aspects that allow multiple selections.
   - Double click the delete button (☒) to remove a selection.

   A green check (✓) means the aspect has no errors; a red X (✗) means additional setup is required. Hover over the X for additional information.

3. For the 'Golden Record Object Types' aspect choose the object types allowed for link golden records and click the **Select** button.

> **Note:** These object types must have all of the attribute's reference types and data container types valid for survivorship rules used to promote from source records.

4. Save or cancel your work:
   - Click the **Save** button to save a configuration once it has no errors.
   - When enabled, click the **Save pending** button to save your work while errors exist.
   - When enabled, click the **Restore live settings** button to undo the changes made to a previously error-free, saved configuration.
   - Click the **Cancel** button to undo all changes made in this dialog.

Configuring the Link Golden Record Match Action

Before setting up the Link Golden Record match action, first configure the match criteria as defined in the **Match Criteria** topic.

The Match Action defines which records are automatically set as matches or non-matches, and which records must be reviewed manually to determine their status.

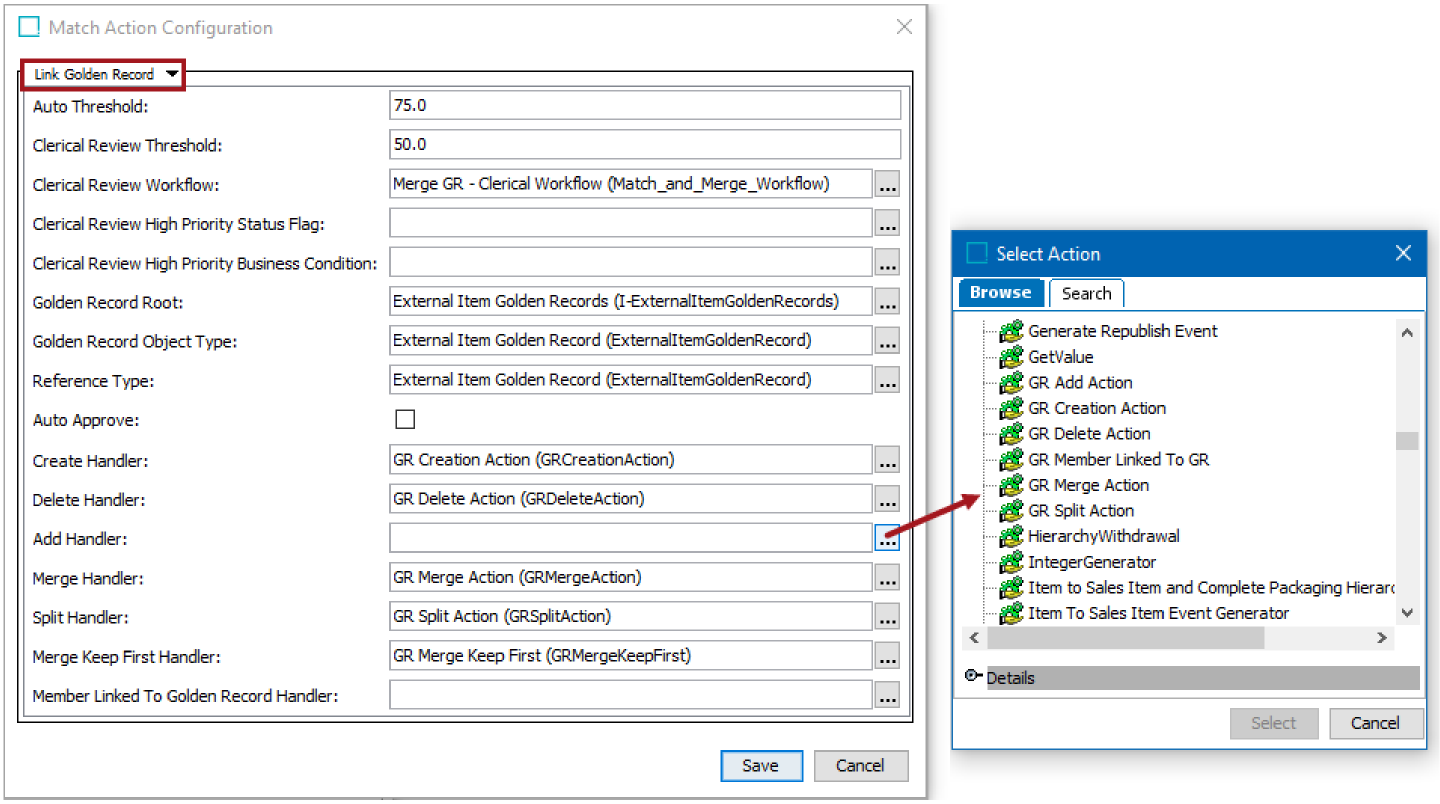
Working with a golden record setup often requires specific actions to handle a golden record change (created, deleted, merged, split, etc.). In these cases, the matching algorithm can be configured to call a business rule via a handler in order to allow for more granular processing of events. For example, when two existing golden records are merged, in addition to the survivorship rules, other actions may be needed.

Configuration

To set up the link golden record match action:

1. In System Setup, on the matching algorithm node open the appropriate Matching Algorithm.
2. On the Matching Algorithm tab, open the Match Action flipper and click the **Edit Match Action** link.
3. On the Match Action Configuration dialog, select **Link Golden Record** from the dropdown.

   For information on a parameter, hover over the parameter field to display help text.



Match Action Configuration

Link Golden Record

Auto Threshold:	75.0
Clerical Review Threshold:	50.0
Clerical Review Workflow:	Merge GR - Clerical Workflow (Match_and_Merge_Workflow)
Clerical Review High Priority Status Flag:	
Clerical Review High Priority Business Condition:	
Golden Record Root:	External Item Golden Records (I-ExternalItemGoldenRecords)
Golden Record Object Type:	External Item Golden Record (ExternalItemGoldenRecord)
Reference Type:	External Item Golden Record (ExternalItemGoldenRecord)
Auto Approve:	□
Create Handler:	GR Creation Action (GRCreationAction)
Delete Handler:	GR Delete Action (GRDeleteAction)
Add Handler:	
Merge Handler:	GR Merge Action (GRMergeAction)
Split Handler:	GR Split Action (GRSplitAction)
Merge Keep First Handler:	GR Merge Keep First (GRMergeKeepFirst)
Member Linked To Golden Record Handler:	

4. Configure the following parameters.
    - For the **Auto Threshold** parameter, specify the equality measurement for automatic linking; namely, how equal two source objects must be to have them automatically linked to the same golden record.
    - For the **Clerical Review Workflow** parameter, click the ellipsis button (...), and select the relevant clerical review workflow. A clerical review workflow can be as simple or elaborate as needed. For more information, refer to the **Creating a Workflow** topic in the **Workflows** documentation.
    - For the **Clerical Review High Priority Status Flag** parameter, click the ellipsis button (...) and select the STEP workflow status flag that is used to designate high priority tasks in the clerical review workflow.

        > **Important:** The matching algorithm determines which Status Flags are set (or not set) so no other Status Flags should be configured in the Clerical Review Workflow.

    - For the **Clerical Review High Priority Business Condition** parameter, click the ellipsis button (...) and select the business condition that is used to verify if a task is of high priority.

        > **Note:** If a status flag is configured, but a business condition is not configured, then the status flags behave as if a business condition evaluated to true.
        >
        > If a business condition is configured, and a status flag is not configured, the business condition is ignored.
        >
        > The business condition is evaluated on each object in the clerical review task (each potential duplicate) in the context of the matcher and has access to the Current Object bind.
        >
        > Though the business condition runs as a part of matching and it involves a clerical review, no matching or Workflow binds are available.

    - For the **Golden Record Root** and **Golden Record Object Type** parameters, specify the root node under which golden records should be stored and the golden record object type.
    - For the **Auto Approve** parameter, check to automatically approve the golden records being created.
5. Click the ellipsis button (...) to supply the appropriate handler(s) for your matching and linking solution:
    - For the **Create Handler** parameter, the selected business action runs on the golden record after it has been created and has initial source object links, but before survivorship rules run.
    - For the **Delete Handler** parameter, the selected business action runs after the golden record is deleted. For example, when merging two golden records, one is deleted. The delete handler runs after the merge handler, which means that the golden record has no linked source records. Alternatively, in this case, if the delete handler field is blank, then the incoming references of the surviving golden record are re-targeted and re-approved (if they were approved before); the golden record is deleted and, if auto-approve is enabled, the deletion is approved.

- For the **Add Handler** parameter, the selected business action runs on the golden record after a new source is added, but before any survivorship rules run.

- For the **Merge Handler** parameter, the selected business action runs when two golden records are merged (because their sources match). The source(s) are moved to the golden record that will be kept and the delete handler is called for the golden record that will be deleted.

- For the **Split Handler** parameter, the selected business action runs when a golden record is split (because one or more of its sources no longer match). The split handler runs after the new golden record is created and its source record links are updated, but before survivorship rules run. The original and new golden records each reflect the correct source records. The create handler is not called when golden records split.

- For the **Merge Keep First Handler** parameter, the selected business condition runs when two golden records are being merged and allows identification of the golden record that should be kept. Use the Current Object and Secondary Object binds in the condition and return one of the following options:

  - null = default behavior; keep the golden record with the most members; if there is an equal number, keep the oldest golden record.

  - true = the golden record bound to the Secondary Object is deleted.

  - false = the golden record bound to the Current Object is deleted.

  For more information on, refer to the **Current Object Bind** topic and the **Secondary Object Bind** topic in the **Resource Materials** documentation.

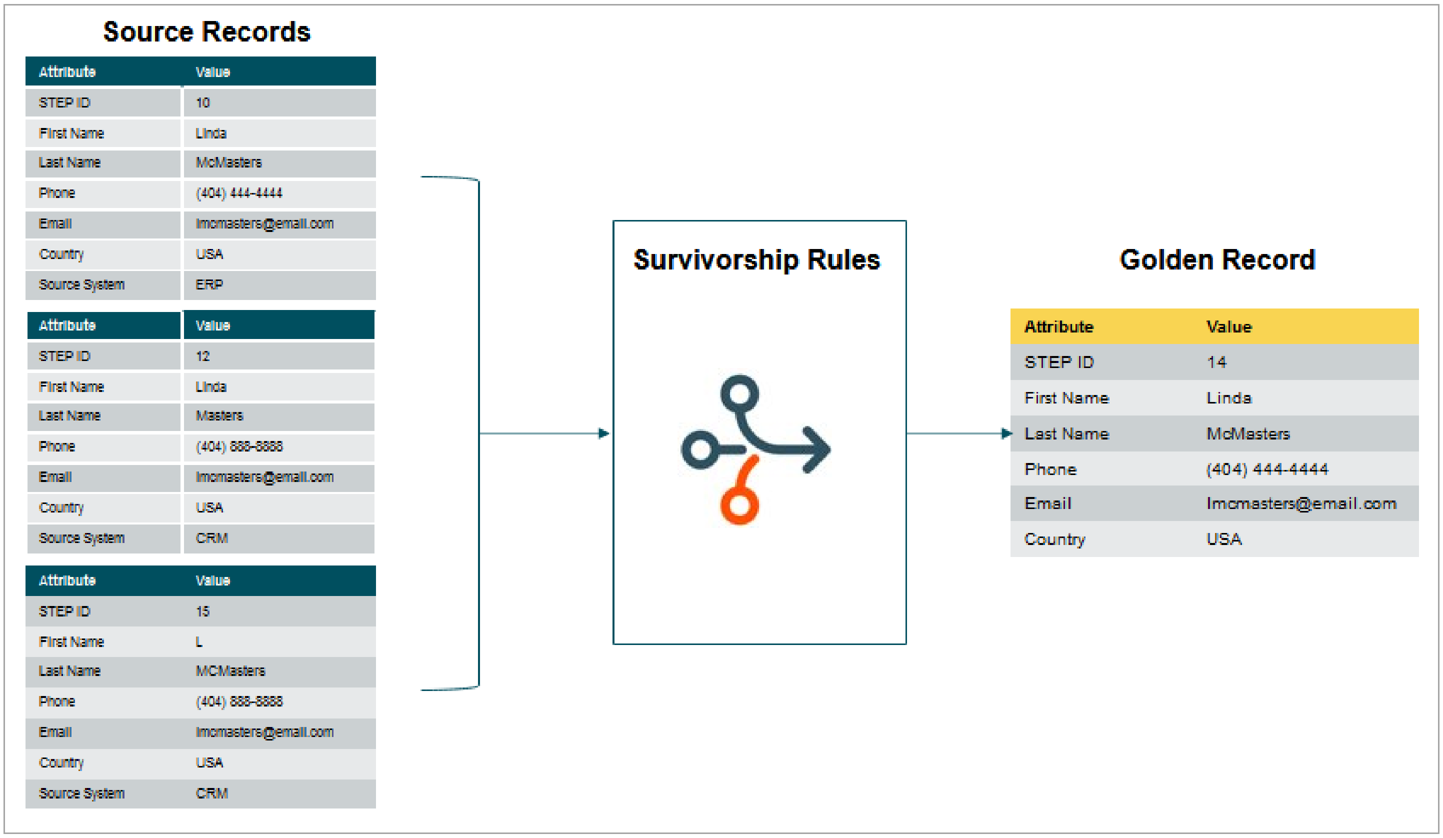
- For the **Member Linked to Golden Record Handler** parameter, the selected business action runs on the source object when a source object link changes from one golden record to a new golden record. The handler runs after the sources have been added, but before survivorship rules run.

6. Click the **Save** button to keep the settings or the **Cancel** button to close the dialog without saving.

Survivorship in Match and Link

In a match and link solution, source records are products or entities that already exist in STEP. The golden record is a new product or entity, created and populated by the survivorship rules.

When survivorship rules run in a match and link solution, the number of sources is unknown; there could be one or many sources. This lack of information is especially important to remember if writing business action survivorship rules.



Source Records

Attribute	Value
STEP ID	10
First Name	Linda
Last Name	McMasters
Phone	(404) 444-4444
Email	lmcmasters@email.com
Country	USA
Source System	ERP

Attribute	Value
STEP ID	12
First Name	Linda
Last Name	Masters
Phone	(404) 888-8888
Email	lmcmasters@email.com
Country	USA
Source System	CRM

Attribute	Value
STEP ID	15
First Name	L
Last Name	MCMasters
Phone	(404) 888-8888
Email	lmcmasters@email.com
Country	USA
Source System	CRM

Golden Record

Attribute	Value
STEP ID	14
First Name	Linda
Last Name	McMasters
Phone	(404) 444-4444
Email	lmcmasters@email.com
Country	USA

Match and link survivorship rules are only ever run in the context of an event processor; they are not used when merging source records.

Golden records should not be merged in a match and link solution as that conflicts with the general rule that the golden record is not to be directly edited.

Trusted Source

To use the trusted source survivorship rule, information about the source, e.g., the object's originating system / supplier, must be available on the source objects. This attribute is defined in the general Matching component model as the 'Data Source Attribute.' Typically, this attribute is a mandatory LOV-based description attribute that does not allow users to add values. For more information, refer to the **Configuring Matching Component Model** topic.

Information from a source outside the list of trusted sources is not copied to the golden record during a trusted source survivorship rule evaluation. Information on a record without a source attribute is not copied to the golden record by trusted source survivorship rules.

For more information, refer to the **Configuring Survivorship Rules** topic.

Most Recent

The 'Most Recent' survivorship rule strategy takes the most recent data from a golden record's source objects.

The most recent can be qualified either by the revision date in STEP or by a 'Last Edited' date attribute. The date attribute option allows promotion of data based on the time of edit in source systems.

For more information, refer to the **Golden Records Survivorship Rules** topic.

Business Action Rule

Solutions commonly include special rules for survivorship that can be implemented via business actions that run as survivorship rules.

Note: A survivorship rule should never update values outside the golden record.

For more information, refer to the **Business Actions** topic in the **Business Rule** documentation.

Configuring the Link Event Processor

An event processor monitors the system for actionable events on specified objects, ensures match codes are regenerated, and runs the matching algorithms in response to any relevant change. For example, consider an object that is subject to a matching algorithm. When the match code assignment or data on that object is approved, the approval can trigger the event processor to regenerate the match code for that object and run the algorithm. Alternatively, events can be passed to the event processor via a republish business rule as part of a workflow or integration.

Event processors write to a background process log so you can identify when events were processed and what actions were taken in response. Additionally, event processor performance measurements are available on the Statistics tab for both matching algorithms and match code configurations.

A match and link match algorithm is run via an event processor configured to trigger the matching algorithm.

Important: While it is possible to use the same match and link matching algorithm across several event processors, that usually results in an optimistic locking and/or unique constraint violation when the two processors conflict. To avoid these issues, ensure that each algorithm on the system is run by a single event processor.

Configuration

To configure an event processor for a matching solution:

1. Create a matching event processor as defined in the **Creating an Event Processor** topic and the **Matching Processing Plugin Parameters and Triggers** topic of the **System Setup** documentation.
2. In System Setup, open your event processor and review the following parameter settings:

   Open the Configuration flipper and click the **Edit Configuration** link to display the wizard.

   - On the Configure Event Processor step, verify the Select Processor parameter is set to 'Matching'
   - On the Configure Processing Plugin step, verify the Event Processing parameter is set to 'Generate/Update Match Code Values and Run Matching Algorithm'
   - On the Configure Processing Plugin step, verify the Matching Algorithms parameter displays the desired matching algorithm(s)
   - On the Schedule Event Processor step, verify the Start parameter shows the desired schedule (Every 1 minute is recommended.)

   Close the wizard and review the event processor editor.

   - On the Event Processor tab, open the Configuration flipper, and verify the Queue Status parameter is set to Read Events

- On the Event Triggering Definitions tab, verify the appropriate event triggering definitions are selected

  For a **match and merge** scenario, based on the selected algorithm, for existing golden records, the event processor performs a merge or initiates a clerical review. Add triggers for the following:

  - references defined by your **Matching component model**: Non-Duplicate Reference Types
  - references defined by your **Matching - Merge Golden Record References component model**: Unmerged-From Relation Reference Types
  - attributes, references, and data containers included in your **Match Criteria**

    > **Important:** For accurate match and merge functionality, the event processor must trigger on updates that can change the outcome of the record comparisons. To accomplish this, the recommendation is to trigger on any attribute, reference, or data container that is used in the match criteria.

  For a match and merge scenario, avoid triggers on the following attributes and reference types as defined by your component models:

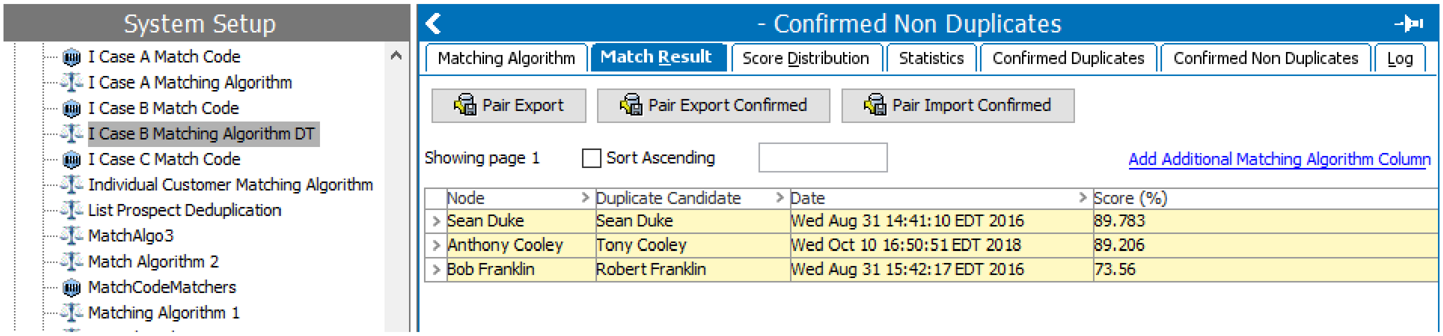
  - Potential Duplicate Reference Type
  - Merged-Into Relation Reference Types
  - Source Relation Reference Type
  - Potential Duplicate Match Algorithm ID Attribute
  - Source Record ID Attribute
  - Deactivated Attribute

3. Enable the matching event processor as defined in the **Enable Event Processor** section of the **Running an Event Processor** topic in the **System Setup** documentation.

For more information, refer to the **Maintaining an Event Processor** topic of the **System Setup** documentation.

Match and Link in Workbench

When the matching algorithm runs, the possible matches can be viewed on the 'Match Result' tab of the matching algorithm. Workbench supports the matching user actions defined below.

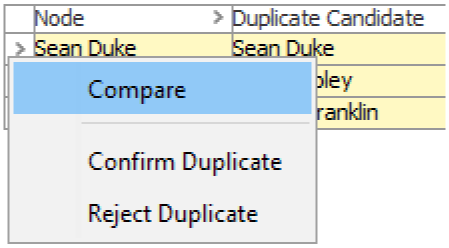


Node	Duplicate Candidate	Date	Score (%)
Sean Duke	Sean Duke	Wed Aug 31 14:41:10 EDT 2016	89.783
Anthony Cooley	Tony Cooley	Wed Oct 10 16:50:51 EDT 2018	89.206
Bob Franklin	Robert Franklin	Wed Aug 31 15:42:17 EDT 2016	73.56

You can merge identified duplicate source records using the Web UI. For more information, refer to **Merging Confirmed Matches** topic.

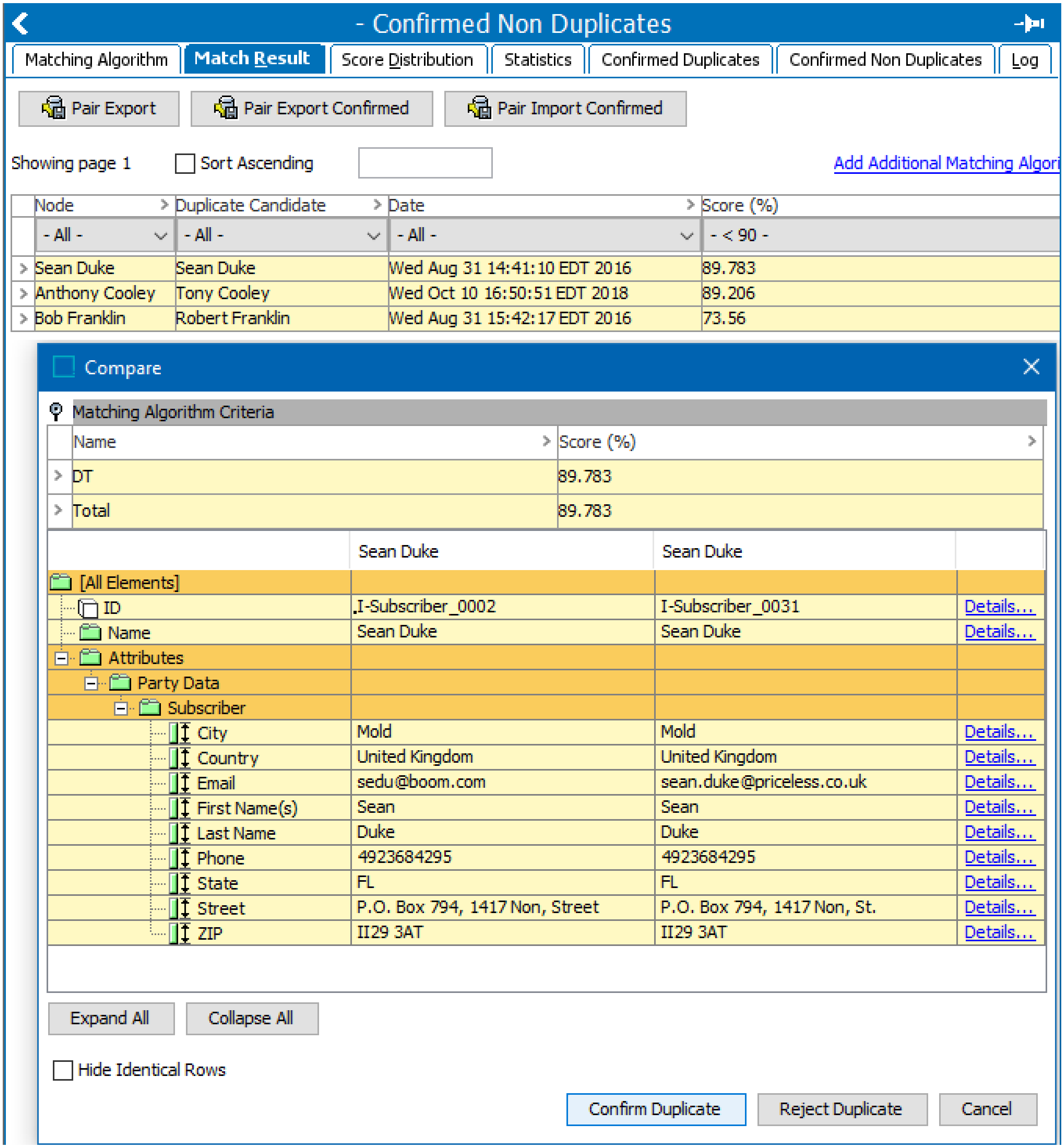
Compare Match Result

To compare an object with its duplicate or non duplicate candidate, on the 'Match Result' or 'Confirmed Non Duplicates' tab, right-click the first column of a row and select the 'Compare' option.



The 'Compare' screen shows the similarities and differences between the paired objects. When accessed via the 'Match Result,' you can confirm or reject duplicates via the 'Confirm Duplicate' and 'Reject Duplicate' buttons.

Right-click a column heading and select 'Filtering enabled' to allow easy navigation and filtering of desired data. Filtering in the following image has been set to include only rows that have a score of less than 90.



Showing page 1 □ Sort Ascending

Node	Duplicate Candidate	Date	Score (%)
- All -	- All -	- All -	- < 90 -
Sean Duke	Sean Duke	Wed Aug 31 14:41:10 EDT 2016	89.783
Anthony Cooley	Tony Cooley	Wed Oct 10 16:50:51 EDT 2018	89.206
Bob Franklin	Robert Franklin	Wed Aug 31 15:42:17 EDT 2016	73.56

Compare

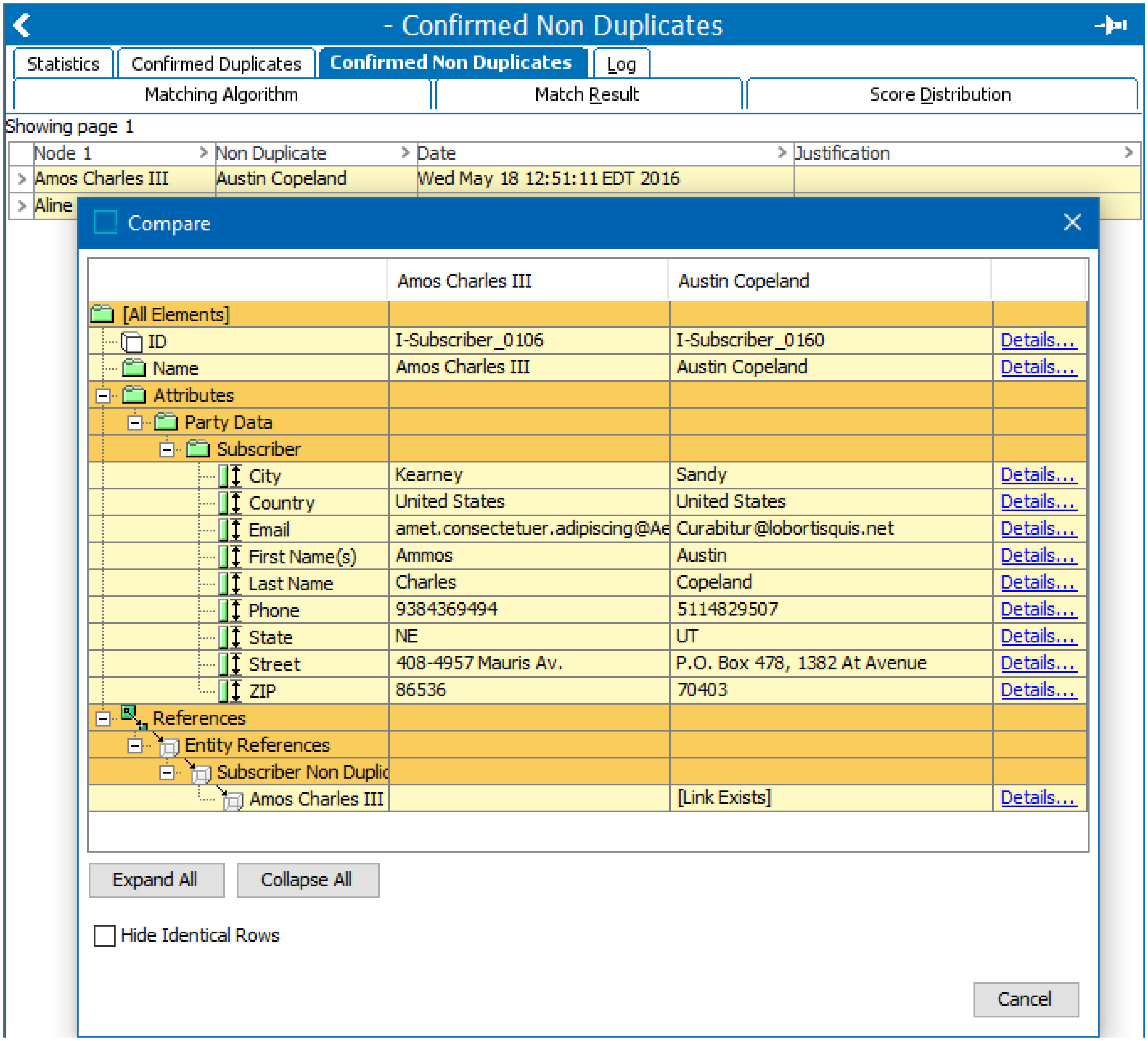
Matching Algorithm Criteria

Name	Score (%)
DT	89.783
Total	89.783

	Sean Duke	Sean Duke	
[All Elements]			
ID	I-Subscriber_0002	I-Subscriber_0031	Details...
Name	Sean Duke	Sean Duke	Details...
Attributes			
Party Data			
Subscriber			
City	Mold	Mold	Details...
Country	United Kingdom	United Kingdom	Details...
Email	sedu@boom.com	sean.duke@priceless.co.uk	Details...
First Name(s)	Sean	Sean	Details...
Last Name	Duke	Duke	Details...
Phone	4923684295	4923684295	Details...
State	FL	FL	Details...
Street	P.O. Box 794, 1417 Non, Street	P.O. Box 794, 1417 Non, St.	Details...
ZIP	II29 3AT	II29 3AT	Details...

□ Hide Identical Rows

When accessed from the 'Confirmed Non Duplicates' tab, you can only view the data, no further actions are available.



Showing page 1

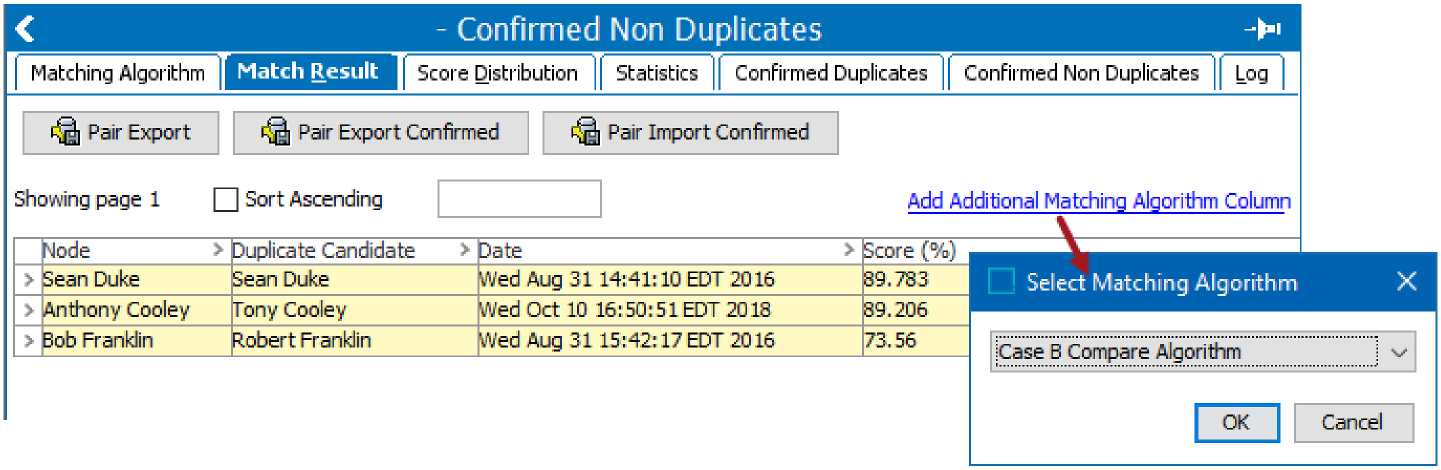
Node 1	Non Duplicate	Date	Justification
Amos Charles III	Austin Copeland	Wed May 18 12:51:11 EDT 2016	
Aline			

	Amos Charles III	Austin Copeland	
[All Elements]			
ID	I-Subscriber_0106	I-Subscriber_0160	Details...
Name	Amos Charles III	Austin Copeland	Details...
Attributes			
Party Data			
Subscriber			
City	Kearney	Sandy	Details...
Country	United States	United States	Details...
Email	amet.consectetuer.adipiscing@Ae	Curabitur@lobortisquis.net	Details...
First Name(s)	Ammos	Austin	Details...
Last Name	Charles	Copeland	Details...
Phone	9384369494	5114829507	Details...
State	NE	UT	Details...
Street	408-4957 Mauris Av.	P.O. Box 478, 1382 At Avenue	Details...
ZIP	86536	70403	Details...
References			
Entity References			
Subscriber Non Duplic			
Amos Charles III		[Link Exists]	Details...

☐ Hide Identical Rows

Adding Additional Matching Algorithm

On the 'Match Result tab, click the **Add Additional Matching Algorithm Column** link to add another matching algorithm to compare the objects. This allows you to review more information about the objects before deciding if they are duplicates or not.



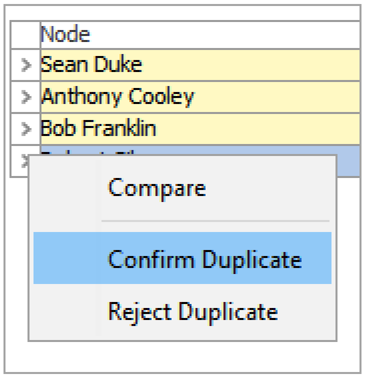
Node	Duplicate Candidate	Date	Score (%)
Sean Duke	Sean Duke	Wed Aug 31 14:41:10 EDT 2016	89.783
Anthony Cooley	Tony Cooley	Wed Oct 10 16:50:51 EDT 2018	89.206
Bob Franklin	Robert Franklin	Wed Aug 31 15:42:17 EDT 2016	73.56

Case B Compare Algorithm

Confirm or Reject a Duplicate

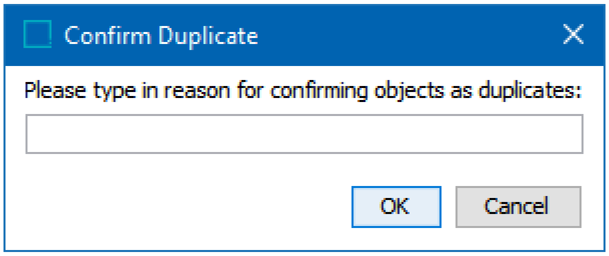
From the 'Match Result' tab, you can compare pairs and mark them as either confirmed duplicates or confirmed non-duplicates.

1. In System Setup, select the relevant matching algorithm, and then click the 'Match Result' tab.
2. Click the row that contains the record being worked, right-click the arrow in the first column and select **Confirm Duplicate** or **Reject Duplicate** from the menu.



Node
Sean Duke
Anthony Cooley
Bob Franklin

3. Provide a reason for the confirmation / rejection and click **OK**. The reason is saved as an attribute value on the corresponding Confirm Duplicate / Confirm Non Duplicate reference.



Please type in reason for confirming objects as duplicates:

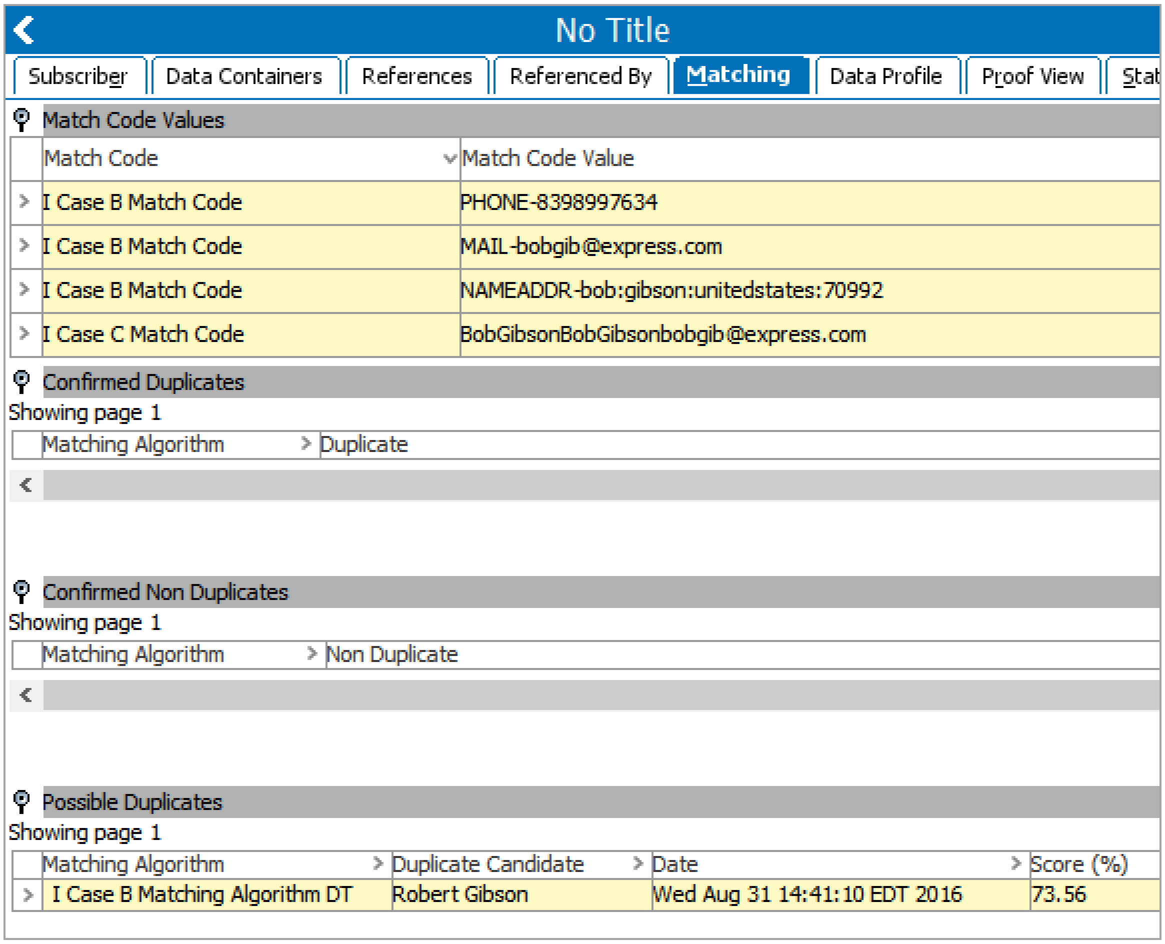
- The **Duplicate** reference type is created between two objects that are manually confirmed as duplicates. This reference means that regardless of how the objects are modified, the matching algorithm always considers them as duplicates.
- The **Non Duplicate** reference type is created between two objects when a duplicate candidate is rejected. This reference means the two objects will never be identified as duplicates by the matching algorithm regardless of how they are modified.
- These references can be manually removed via the 'References' tab of the object in question.

View Matched Objects in Tree

Duplicate information can also be viewed directly on each link golden record source record in the Tree.

Choose a method to view the object:

- In the Tree, select the relevant source record and click the 'Matching' tab.



Match Code Values

Match Code	Match Code Value
I Case B Match Code	PHONE-8398997634
I Case B Match Code	MAIL-bobgib@express.com
I Case B Match Code	NAMEADDR-bob:gibson:unitedstates:70992
I Case C Match Code	BobGibsonBobGibsonbobgib@express.com

Confirmed Duplicates

Showing page 1

Matching Algorithm	Duplicate

Confirmed Non Duplicates

Showing page 1

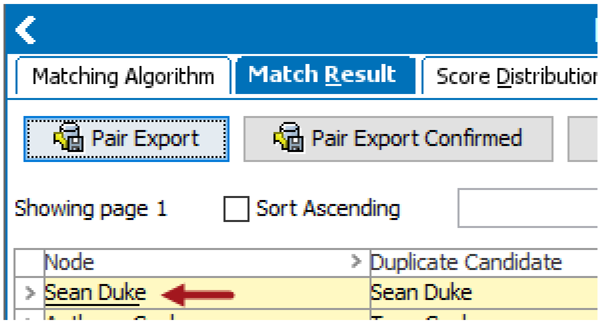
Matching Algorithm	Non Duplicate

Possible Duplicates

Showing page 1

Matching Algorithm	Duplicate Candidate	Date	Score (%)
I Case B Matching Algorithm DT	Robert Gibson	Wed Aug 31 14:41:10 EDT 2016	73.56

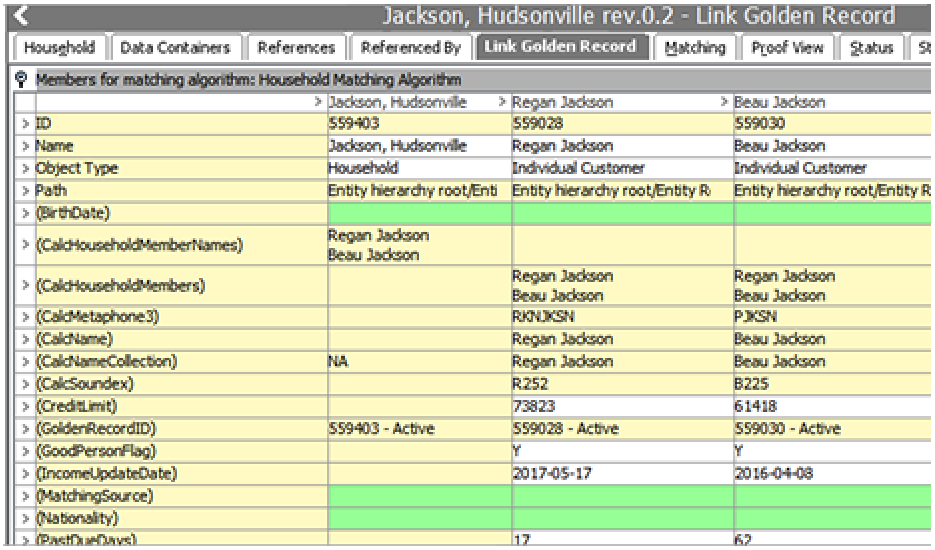
- On the 'Match Result' tab, click the link of the object to open the object editor in Tree.



Showing page 1 □ Sort Ascending

Node	Duplicate Candidate
Sean Duke	Sean Duke

- For the link golden record, the 'Link Golden Record' tab display the source records that are linked to it.



Members for matching algorithm: Household Matching Algorithm

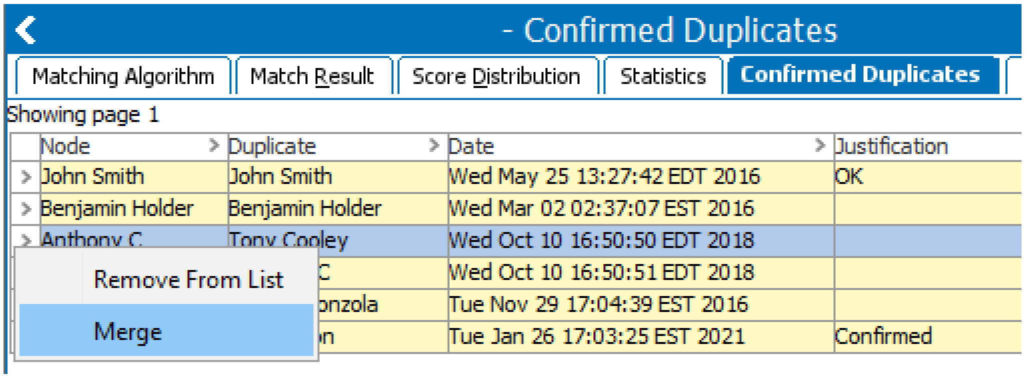
	Jackson, Hudsonville	Regan Jackson	Beau Jackson
ID	559403	559028	559030
Name	Jackson, Hudsonville	Regan Jackson	Beau Jackson
Object Type	Household	Individual Customer	Individual Customer
Path	Entity hierarchy root/Enti	Entity hierarchy root/Entity R	Entity hierarchy root/Entity R
(BirthDate)			
(CalcHouseholdMemberNames)	Regan Jackson Beau Jackson		
(CalcHouseholdMembers)		Regan Jackson Beau Jackson	Regan Jackson Beau Jackson
(CalcMetaphone3)		RKNJKSN	PJKSN
(CalcName)		Regan Jackson	Beau Jackson
(CalcNameCollection)	NA	Regan Jackson	Beau Jackson
(CalcSoundex)		R252	B225
(CreditLimit)		73823	61418
(GoldenRecordID)	559403 - Active	559028 - Active	559030 - Active
(GoodPersonFlag)		Y	Y
(IncomeUpdateDate)		2017-05-17	2016-04-08
(MatchingSource)			
(Nationality)			
(PastDueDays)		17	62

Merge Confirmed Duplicates

The 'Identify Duplicates' or 'Link Golden Record' actions can create two objects that are confirmed duplicates and it is possible to manually merge them into a single object.

Important: Because duplicate source records are deleted during a merge, this should not be used as part of a golden record solution.

1. From the 'Confirmed Duplicates' tab, right-click the first column and choose the **Merge** option.

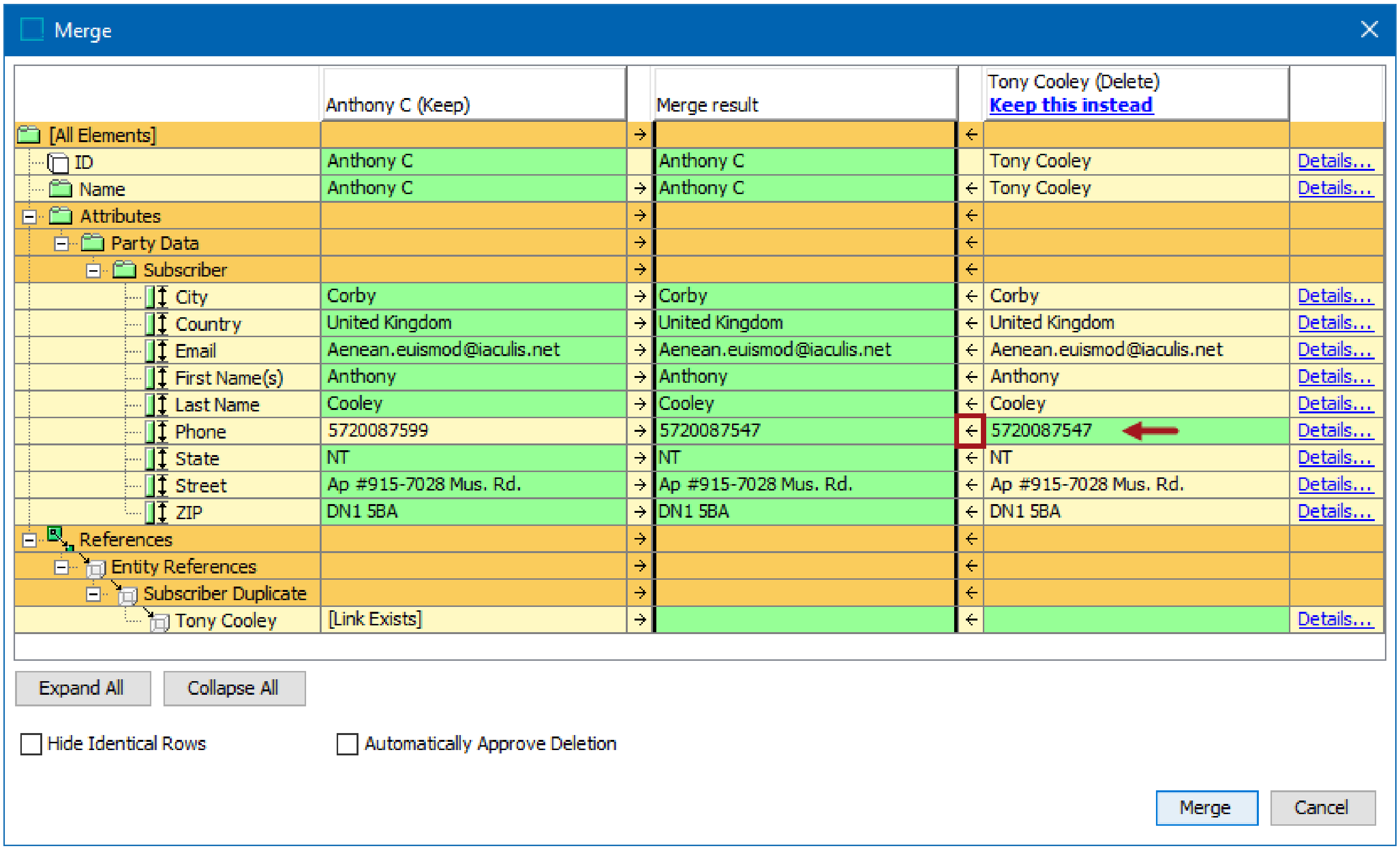


Showing page 1

Node	Duplicate	Date	Justification
John Smith	John Smith	Wed May 25 13:27:42 EDT 2016	OK
Benjamin Holder	Benjamin Holder	Wed Mar 02 02:37:07 EST 2016	
Anthony C	Tony Cooley	Wed Oct 10 16:50:50 EDT 2018	
	C	Wed Oct 10 16:50:51 EDT 2018	
	onzola	Tue Nov 29 17:04:39 EST 2016	
	n	Tue Jan 26 17:03:25 EST 2021	Confirmed

2. On the Merge dialog, review the data and decide which object to keep.

   The first column is the data type. The three data columns are: the '(Keep)' data, the data that will remain after the merge (Merge result), and the '(Delete)' data. The green cell background color indicates where data is taken from.



	Anthony C (Keep)		Merge result		Tony Cooley (Delete) Keep this instead	
[All Elements]		→		←		
ID	Anthony C		Anthony C		Tony Cooley	Details...
Name	Anthony C	→	Anthony C	←	Tony Cooley	Details...
Attributes		→		←		
Party Data		→		←		
Subscriber		→		←		
City	Corby	→	Corby	←	Corby	Details...
Country	United Kingdom	→	United Kingdom	←	United Kingdom	Details...
Email	Aenean.euismod@iaculis.net	→	Aenean.euismod@iaculis.net	←	Aenean.euismod@iaculis.net	Details...
First Name(s)	Anthony	→	Anthony	←	Anthony	Details...
Last Name	Cooley	→	Cooley	←	Cooley	Details...
Phone	5720087599	→	5720087547	←	5720087547	Details...
State	NT	→	NT	←	NT	Details...
Street	Ap #915-7028 Mus. Rd.	→	Ap #915-7028 Mus. Rd.	←	Ap #915-7028 Mus. Rd.	Details...
ZIP	DN1 5BA	→	DN1 5BA	←	DN1 5BA	Details...
References		→		←		
Entity References		→		←		
Subscriber Duplicate		→		←		
Tony Cooley	[Link Exists]	→		←		Details...

☐ Hide Identical Rows

☐ Automatically Approve Deletion

- Click the **Details...** link to open a large display of the data on the selected row.
- Click the **Expand All** or **Collapse All** buttons to show or hide the detailed data.
- Check the **Hide Identical Rows** checkbox to show only the rows with different data.
- Check the **Automatically Approve Deletion** checkbox to approve deletion of objects in the 'Delete' column during the merge process and avoid having to manually delete the duplicate record.
- Click the **Keep this instead** link to move all data from the (Delete) column into the Merge result column.
- Click the arrow on an individual row to move only the data from that cell to the Merge result column, as shown for the Phone row.
- When the data in the Merge result column is the record you want to keep, click the **Merge** button to perform the merge and keep a single record.

Merge Considerations

If the object that remains contains no data in any context, the data is taken from the deleted object and merged into the remaining object. Data is defined as:

- Attributes
- Object name
- Reference types
- Object to classification link types
- Table types
- Object to attribute links

Reference and link types do not accumulate. If the reference or link type is already populated in any context nothing is merged from the object that is deleted.

During the merge process, all references to the deleted object are modified to point to the object that remains in the database. This means that the source objects of these references will be modified. 'Automatically Approve Deletion' only approves the deletion of objects and changes to objects due to references that are pointed to another target are not approved.

Match and Link in Web UI

The Web UI supports the user actions described under the **User Actions** section of the **Match and Link** topic.

Users must add a clerical review widget to the homepage. For more information, refer to the **Adding Widgets to a Homepage** topic in the **Web User Interfaces** documentation.

The following topics are relevant to configuring a Link Golden Record solution in Web UI:

- Configuring a Deduplication Clerical Review
- Golden Record Linked Members Component
- Potential Duplicates List
- Merging Confirmed Matches
- Confirmed Matches Component.

Configuring a Deduplication Clerical Review

A clerical review is the process of manually examining pairs that the algorithm did not identify as duplicates or non-duplicates.

During matching, objects that score between the clerical review threshold and the auto threshold are placed in a clerical review workflow. The potential duplicates from the clerical workflow are then displayed in a Web UI where a user reviews them manually.

Prerequisites

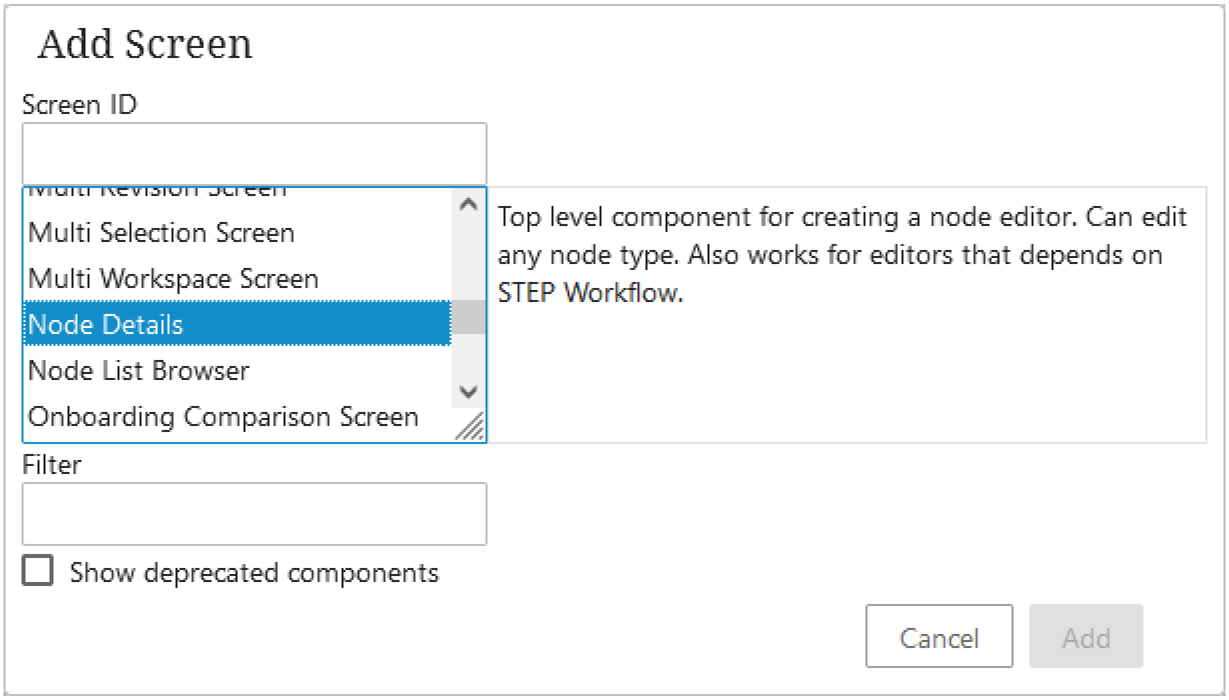
1. This documentation assumes that you are familiar with STEP Web UI design. If you are new to designing Web UIs, it is recommended that you review the **Web UI Getting Started** topics.
2. For more information about creating a workflow, refer to the **Workflows** documentation.
3. For details on configuring the Web UI for Merge Golden Record clerical reviews, refer to the **Merging Confirmed Matches** topic in the **Matching, Linking, and Merging** documentation.

Create and Configure the Deduplication List

You must place the deduplication list inside a tab page.

1. Log in to the Web UI and click the gear wheel icon (⚙) to enter design mode.

2. Click **New**, and then select the **Node Details** screen type.



Add Screen

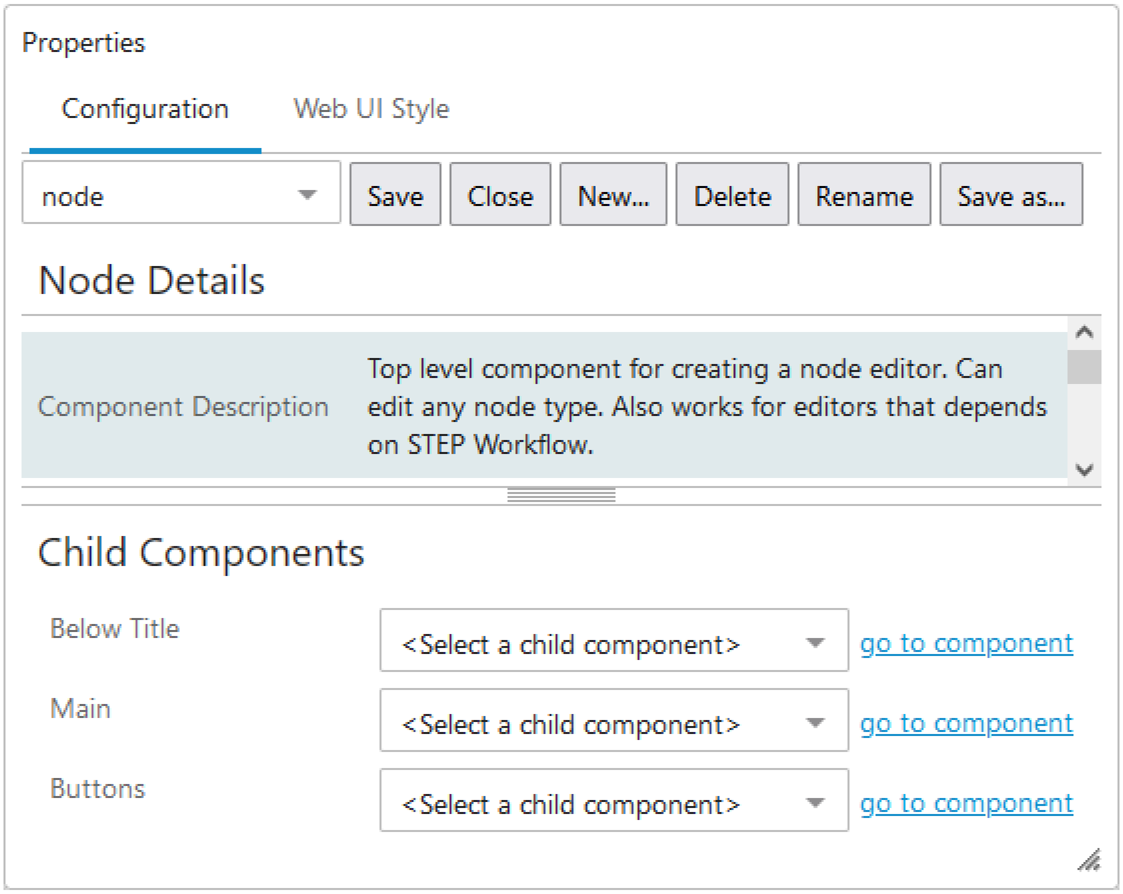
Screen ID

Top level component for creating a node editor. Can edit any node type. Also works for editors that depends on STEP Workflow.

Filter

□ Show deprecated components

3. Enter a **Screen ID** and click **OK**.
4. In the **Child components** area, in the **Main** dropdown, select **Tab Control**.



Properties

Node Details

Component Description Top level component for creating a node editor. Can edit any node type. Also works for editors that depends on STEP Workflow.

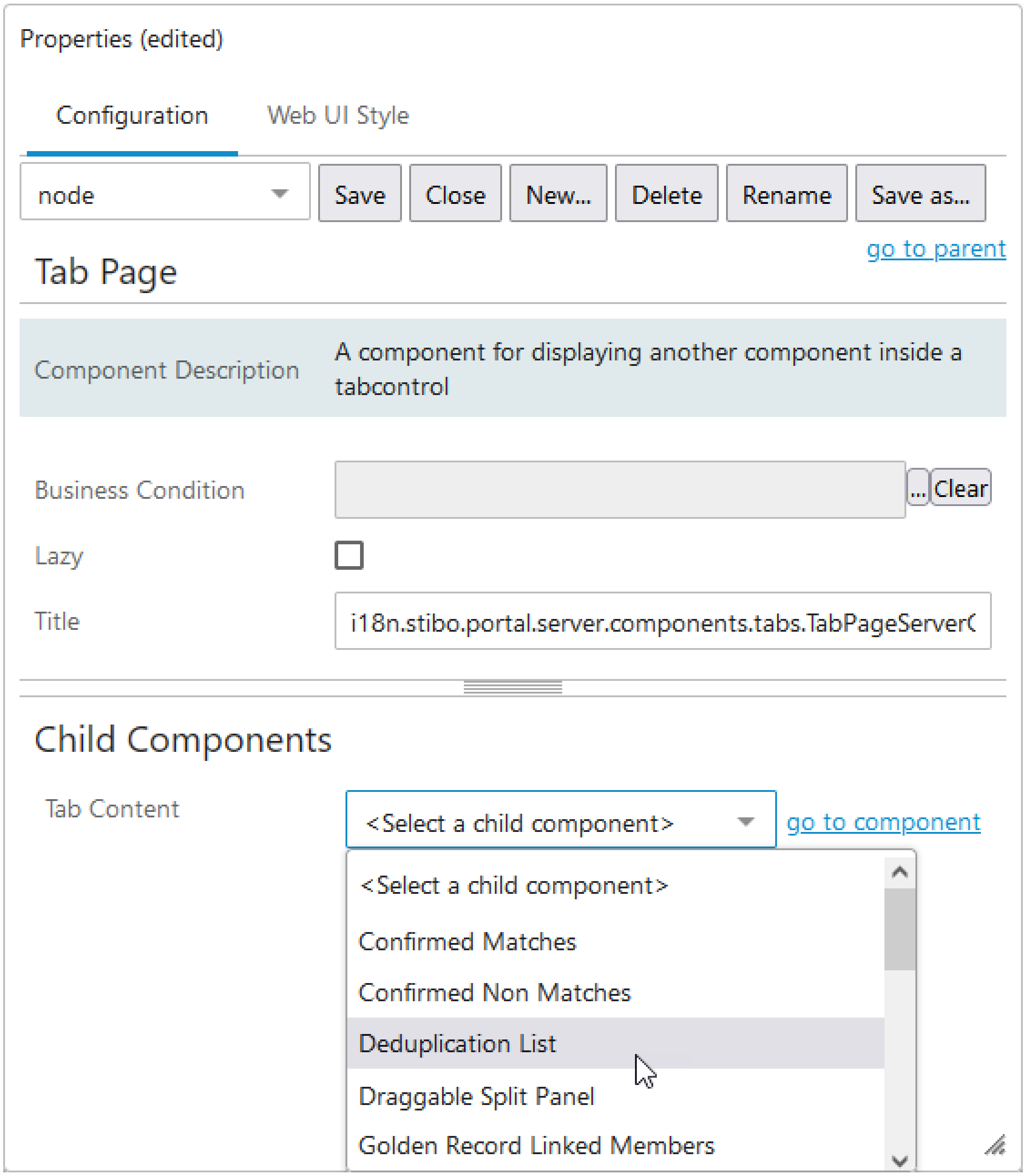
Child Components

Below Title <Select a child component> go to component

Main <Select a child component> go to component

Buttons <Select a child component> go to component

5. Click **go to component** to configure the Tab Control component.
6. Add a **Tab Page** to the Tab Control.
7. On the **Tab Page**, set **Tab Content** to **Deduplication List**.



Properties (edited)

Component Description: A component for displaying another component inside a tabcontrol

Business Condition:

Lazy: □

Title: i18n.stibo.portal.server.components.tabs.TabPageServer(

Tab Content: <Select a child component>

8. Click on the 'go to component' link to configure Deduplication List Properties.
9. In **Headers**, click **Add**, and then select the attribute headers you want to use for the list. Choose meaningful headers that will assist the user with confirming or rejecting potential duplicates.
10. Select **Auto Submit** if you want the task to be automatically submitted when all duplicate candidates have been confirmed or rejected.
11. If you selected **Auto Submit**, in the **Event** field, specify the workflow event to use after auto submit.
12. Select the Hide Selection Buttons option to hide selection buttons such as, ‘Select All’, ‘Show Details,’ etc.

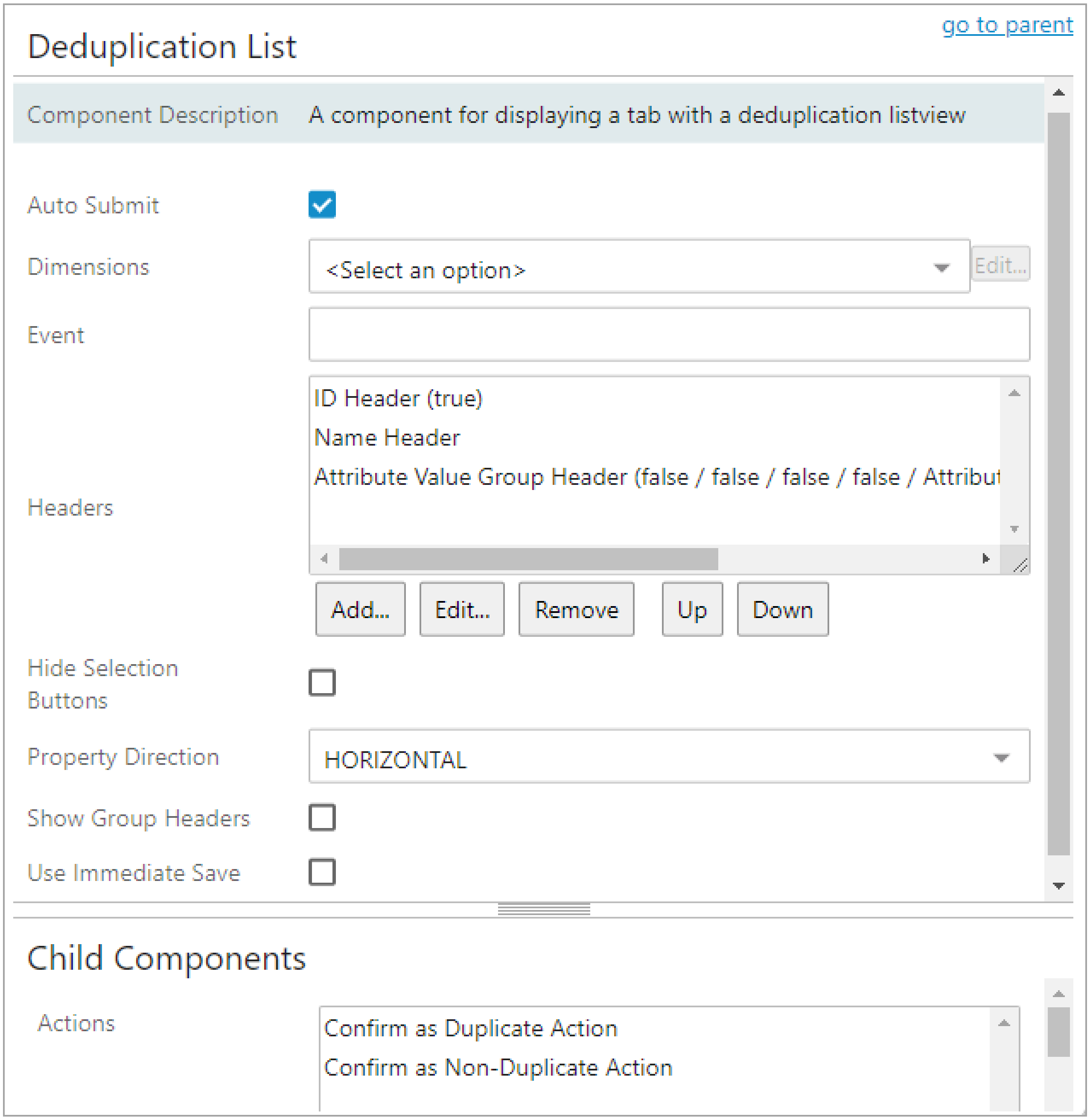
13. Select the Property Direction option if you want to display the data based on the selection from the dropdown. If Horizontal is selected, then columns will be displayed horizontally. If Vertical is selected, then columns will be displayed vertically. If no option is selected, then by default it will be displayed as horizontal.

14. Select 'Show Group Headers' if the attribute group headers should be displayed.

15. Select 'Use Immediate Save' option if you want to save the entered / changed data automatically without clicking the 'Save' button.

16. In the **Child** components area, click **Add**, and then select the actions **Confirm as Duplicate** and **Confirm as Non-Duplicate**.

17. Click **Save** to save the changes.



Deduplication List

Component Description: A component for displaying a tab with a deduplication listview

Auto Submit ☑

Dimensions: <Select an option>

Event:

Headers:
ID Header (true)
Name Header
Attribute Value Group Header (false / false / false / false / Attribut

Hide Selection Buttons □

Property Direction: HORIZONTAL

Show Group Headers □

Use Immediate Save □

Child Components

Actions:
Confirm as Duplicate Action
Confirm as Non-Duplicate Action

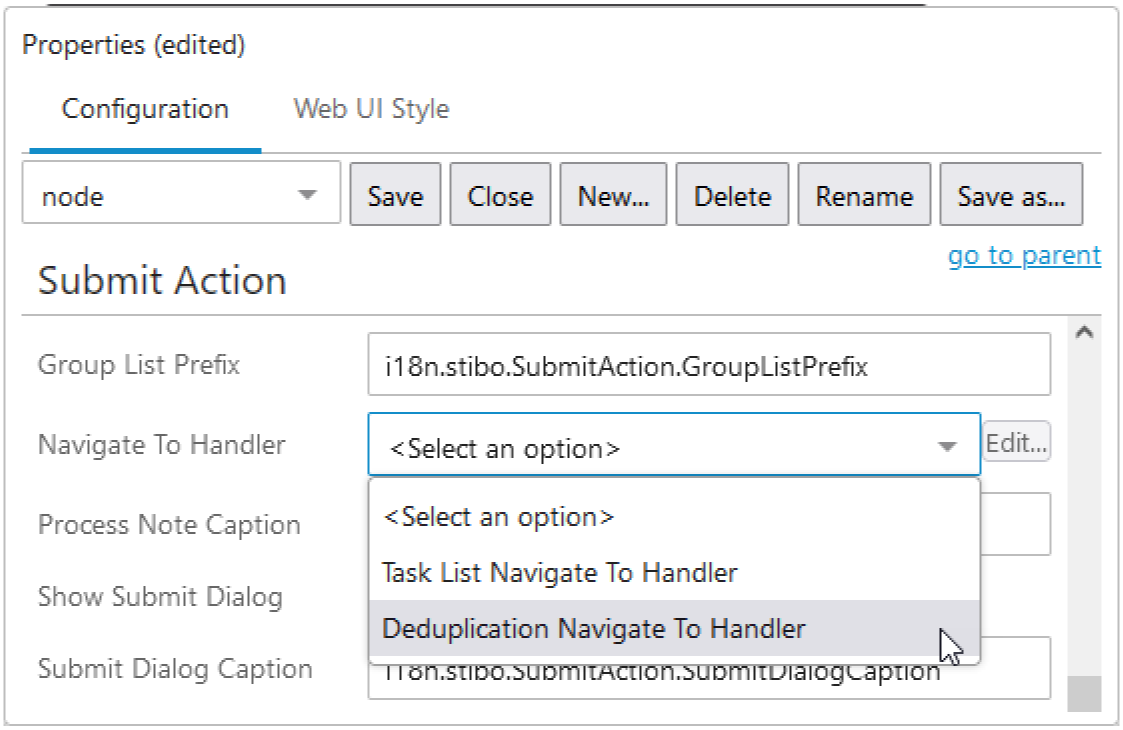
The 'Hide Equal' and 'Mark Different' actions will automatically appear alongside any other actions configured on the Deduplication List. For more information, refer to the **Comparing Data Using Hide Equal and Mark Different** section of the **Web User Interfaces** documentation.

Specify Node Details Buttons

1. Select the **Node Details** screen you just created.
2. In the **Child component** area, from the **Buttons** list, select **Buttons**.
3. Click **go to component**.
4. In the **Child component** area, click **Add**, and then select the **Submit Action**.
5. Double-click the **Submit** action. The **Submit Action Properties** window appears.

   In this window, you can specify where the Web UI navigates to after the **Submit** action. You do this by setting up a **Navigate To Handler**. If you cannot refer to the **Navigate To Handler** list, drag the sizing handle to display all properties.

6. In the **Navigate To Handler** list, select **Deduplication Navigate To Handler**.



Submit Action

Group List Prefix: i18n.stibo.SubmitAction.GroupListPrefix

Navigate To Handler: <Select an option>

Process Note Caption

Show Submit Dialog

Submit Dialog Caption: i18n.stibo.SubmitAction.SubmitDialogCaption

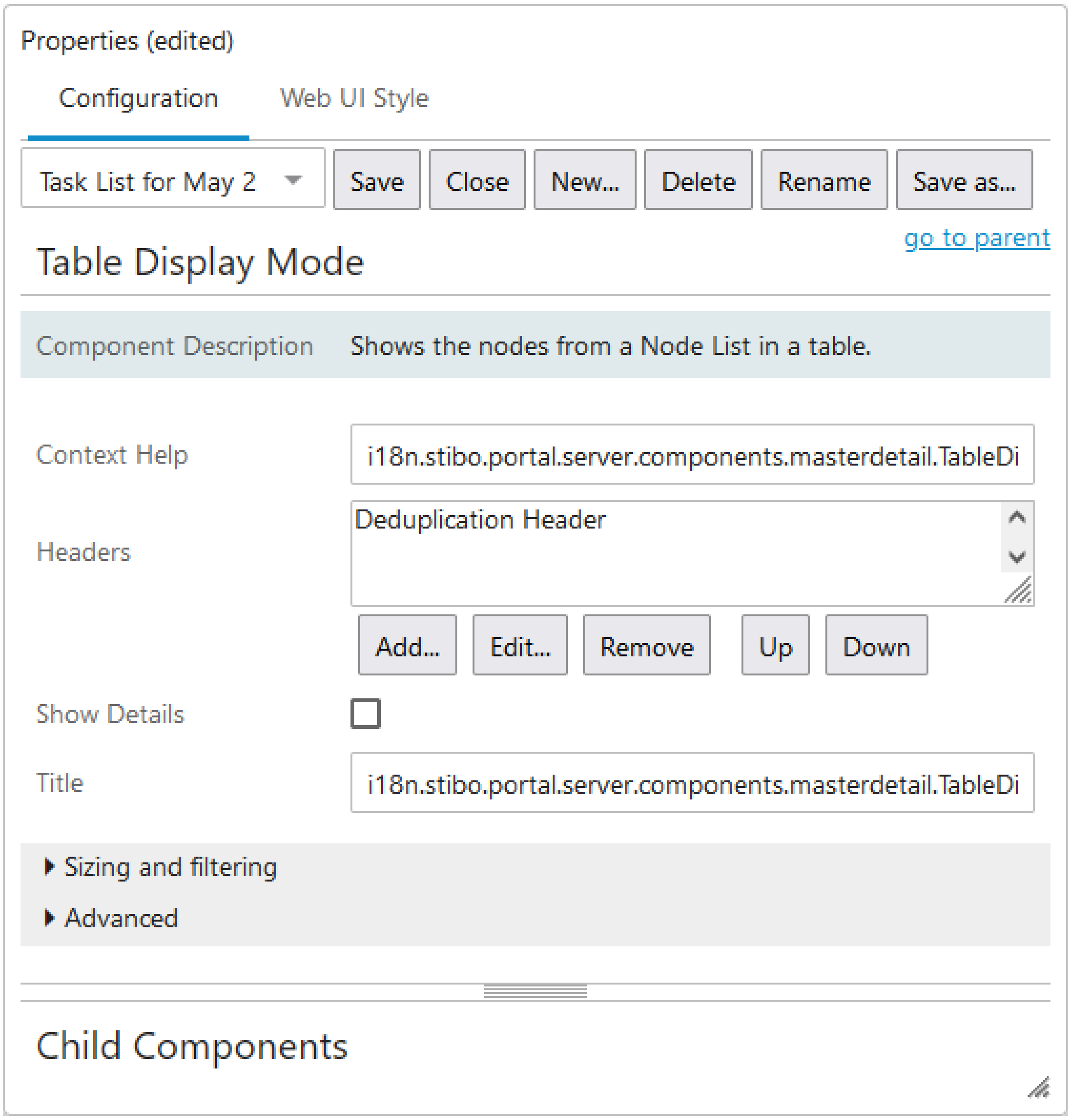
7. Deduplication Navigate To Handler Properties screen will be displayed.
8. In the **Matching Algorithm ID** field, specify the ID of the relevant matching algorithm.
9. In **State ID**, enter the review state of the clerical review workflow, and then click **Save**.

For more information about Submit Action, refer to the **Submit Action** topic.

Create a Task List

The next step is to create a screen to hold the Clerical Review task.

1. Click **New**, in the **Add Screen** window, select **Task List**, and then click **Add**. The **Task List Properties** window appears.
2. In the **Child component** area, in the **Node List**, choose **Node List** and click **go to component**.
3. In the **Node List Properties** window, in the **ID** field, enter an ID for the Node List, and then click **Add**.
4. In the **Task List Properties** window, in the **Child components** area, click **go to component**. The **Node List Properties** window appears.
5. In the **Child components** area, in **Display Modes**, click **Add**.
    - In the **Add component** window, choose **Table Display Mode**, and click **Add**.
    - Double-click **Table Display Mode** and edit the table properties. For **Headers**, click **Add**, and select **Deduplication Header** to generate the link to the deduplication screen. Add any other headers that are meaningful to the users of the list.



Properties (edited)

Table Display Mode

Component Description: Shows the nodes from a Node List in a table.

Context Help: i18n.stibo.portal.server.components.masterdetail.TableDi

Headers: Deduplication Header

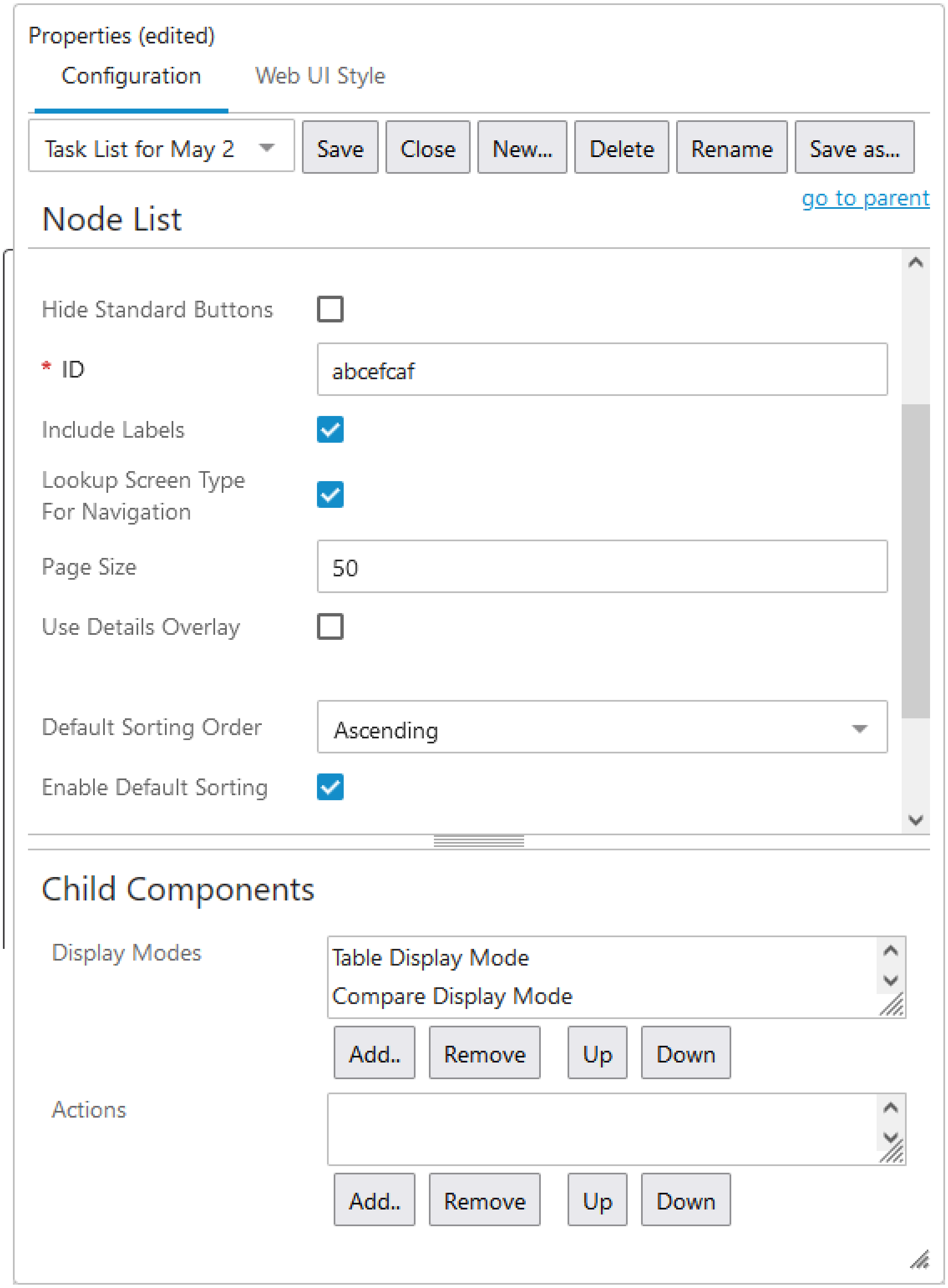
Show Details: □

Title: i18n.stibo.portal.server.components.masterdetail.TableDi

Child Components

- Click the **go to parent** link. The **Node List Properties** dialog displays again.
- If necessary, repeat the bullets in this step to add additional Display Modes (e.g., Compare Display Mode).

6. Back in Node List Properties, enable **Use Details Overlay** to make processing Clerical Review tasks a quicker process. Enabling this parameter means that when after a user clicks either the Confirm Duplicate or Reject Duplicate buttons from the deduplication screen, the user is returned to the clerical review task list after each update, instead of returning to the homepage. Additionally, the clerical review task list is refreshed after each update for confirm or reject action, removing items that have been addressed.



Hide Standard Buttons	□
* ID	abcefcaf
Include Labels	☑
Lookup Screen Type For Navigation	☑
Page Size	50
Use Details Overlay	□
Default Sorting Order	Ascending
Enable Default Sorting	☑

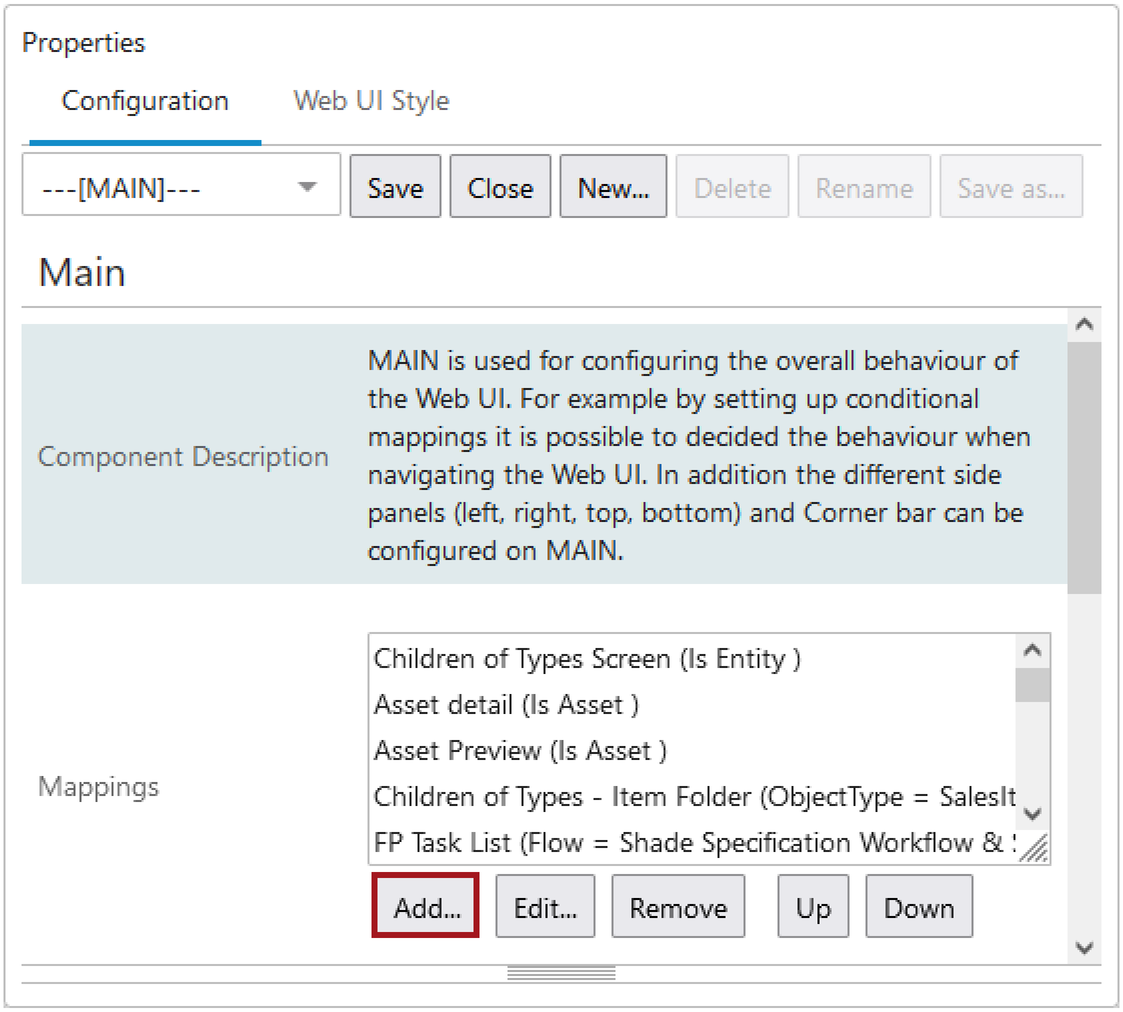
Display Modes: Table Display Mode, Compare Display Mode

7. Click **Save**.

For information about individual table properties, refer to 'Tables and Lists' in the 'Web UI component configuration reference' available at [system]/webui/docs for more information.

Specify Mappings

1. From the screen list, select **Main** to go to the main screen of the designer.
2. In **Mappings**, click **Add**.



Properties

Configuration | Web UI Style

---[MAIN]---

Main

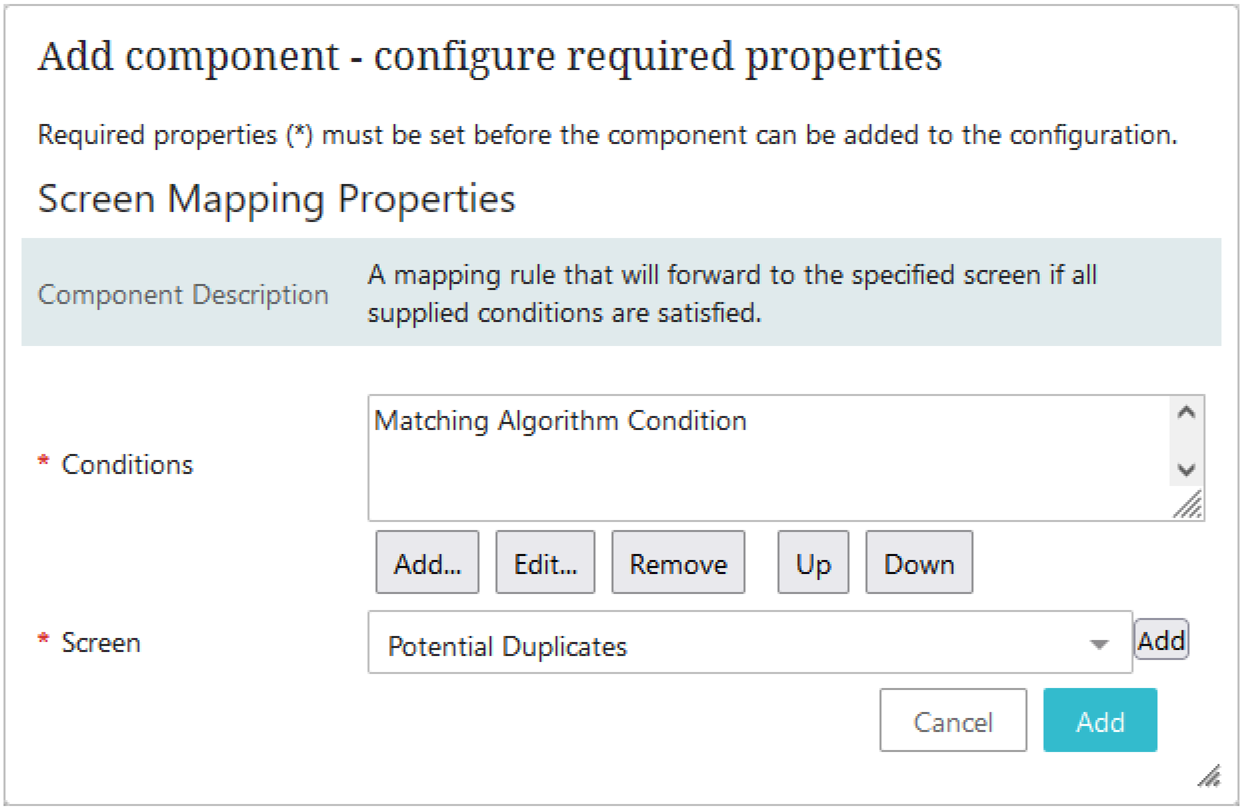
Component Description: MAIN is used for configuring the overall behaviour of the Web UI. For example by setting up conditional mappings it is possible to decided the behaviour when navigating the Web UI. In addition the different side panels (left, right, top, bottom) and Corner bar can be configured on MAIN.

Mappings:
Children of Types Screen (Is Entity)
Asset detail (Is Asset)
Asset Preview (Is Asset)
Children of Types - Item Folder (ObjectType = SalesIt
FP Task List (Flow = Shade Specification Workflow & !

3. In the **Screen Mapping properties** window, in **Conditions**, click **Add**.
4. Select the **Matching Algorithm** condition, and then click **Add**.
5. Double click on the Matching Algorithm to configure the Matching Algorithm Condition Properties.

> **Note:** Matching Algorithm must exist to select in Matching Algorithm Condition Properties screen.

6. In the **Screen** list, select the screen you created for deduplication, click **Add**, and then click **Save**.



Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

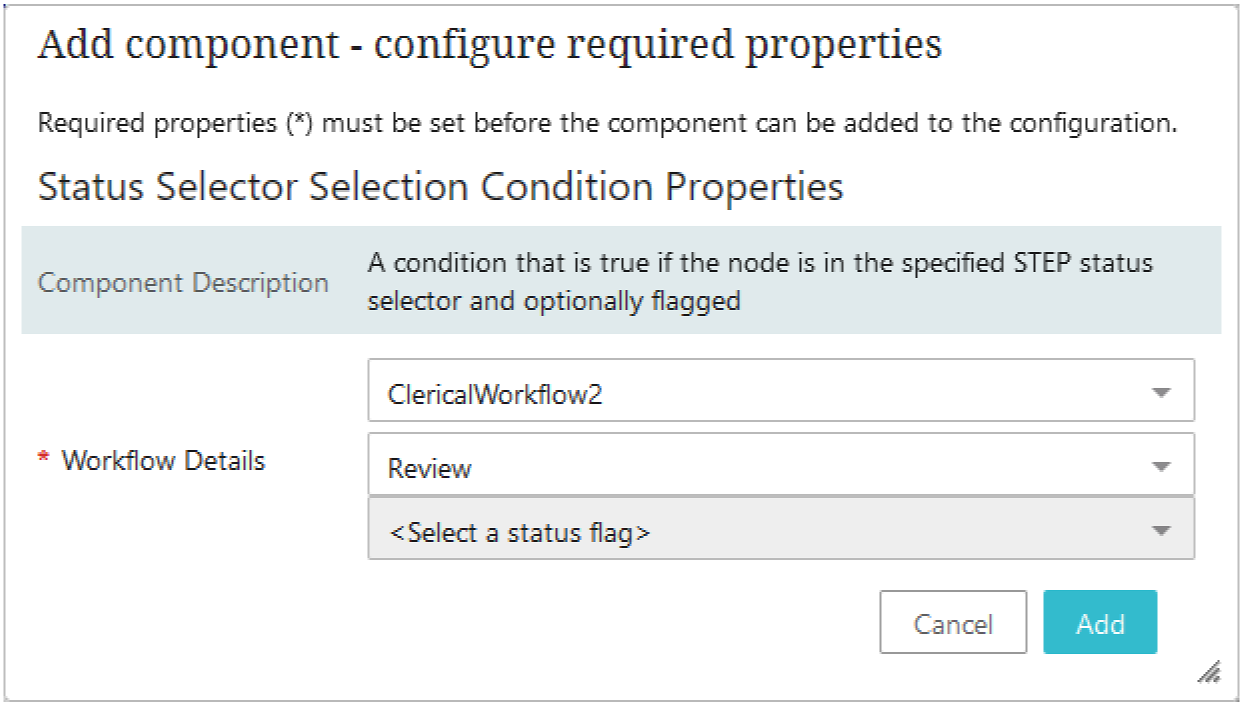
Screen Mapping Properties

Component Description: A mapping rule that will forward to the specified screen if all supplied conditions are satisfied.

* Conditions: Matching Algorithm Condition

* Screen: Potential Duplicates

7. In the **Mappings** list, select the mapping you just created, and then click **Up** to move the screen mapping higher up in the list. Next you will create another mapping.
8. In **Mappings**, click **Add**.
9. In the **Screen Mapping properties** window, in **Conditions**, click **Add**.
10. Select the **Status Selector Selection** condition, and then click **Add**. The **Status Selector Selection Condition properties** window opens.
11. In the **Workflow** list, select the clerical review workflow.
12. In the **Select a state list**, select the start state of the clerical review workflow, click **OK**, and then click **Add**.



Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Status Selector Selection Condition Properties

Component Description

A condition that is true if the node is in the specified STEP status selector and optionally flagged

* Workflow Details

ClericalWorkflow2

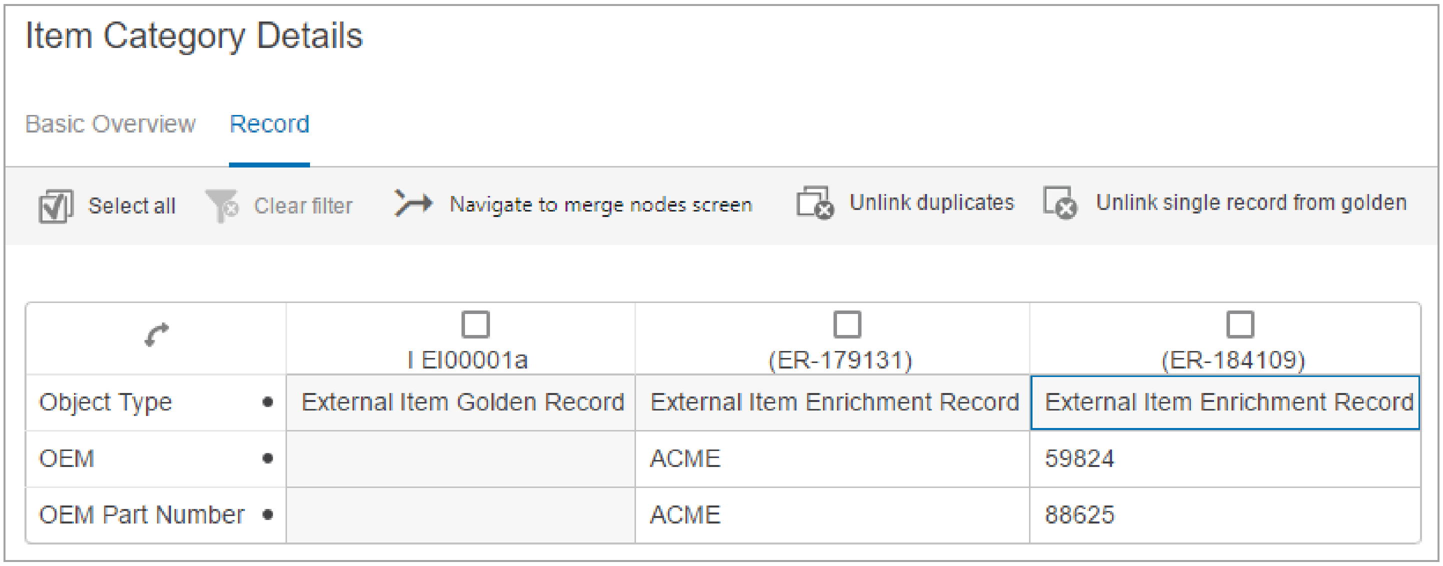
Review

<Select a status flag>

13. In the **Screen Mapping properties** window, in the **Screen** dropdown list, select the Task List screen you just created, and then click **Save**.
14. In the **Mappings** list, select the mapping you just created, and then click **Up** to move the screen mapping higher up in the list.

Golden Record Linked Members Component

In a Match and Link solution, the Golden Record Linked Members component screen allows users to view a golden record node alongside its source records. Attribute headers can be configured for comparing the records and to identify where each inherited value originated. A matching algorithm and corresponding action button(s) are required, and users can customize the table formatting.



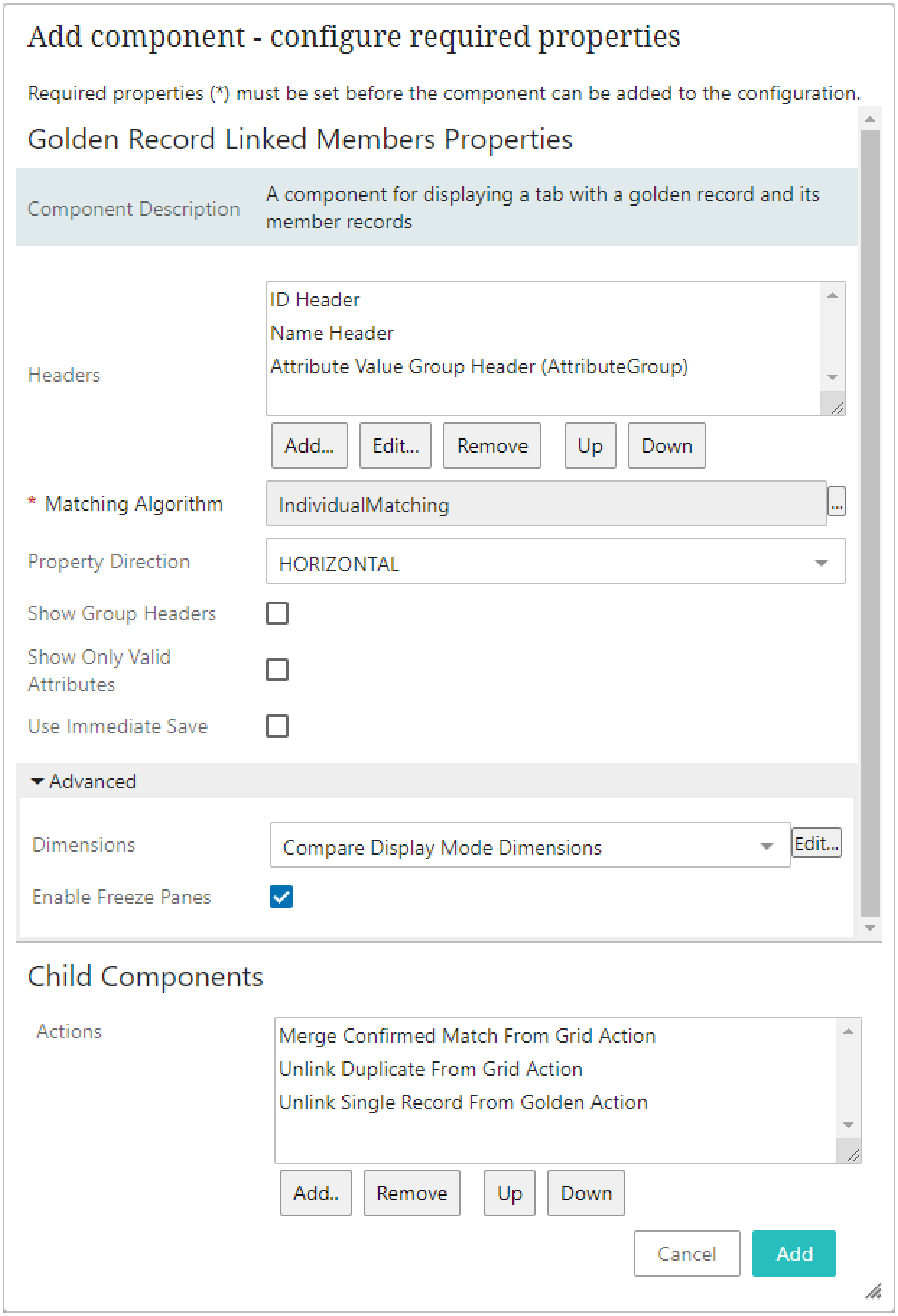
	□ I EI00001a	□ (ER-179131)	□ (ER-184109)
Object Type ●	External Item Golden Record	External Item Enrichment Record	External Item Enrichment Record
OEM ●		ACME	59824
OEM Part Number ●		ACME	88625

Prerequisites

It is expected that anyone configuring the Golden Record Linked Members component is familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** topic of the **Web User Interfaces** documentation.

Configuration

Configure this component on a node details screen.



Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Golden Record Linked Members Properties

Component Description: A component for displaying a tab with a golden record and its member records

Headers:
- ID Header
- Name Header
- Attribute Value Group Header (AttributeGroup)

* Matching Algorithm: IndividualMatching

Property Direction: HORIZONTAL

Show Group Headers: □

Show Only Valid Attributes: □

Use Immediate Save: □

Advanced

Dimensions: Compare Display Mode Dimensions

Enable Freeze Panes: ☑

Child Components

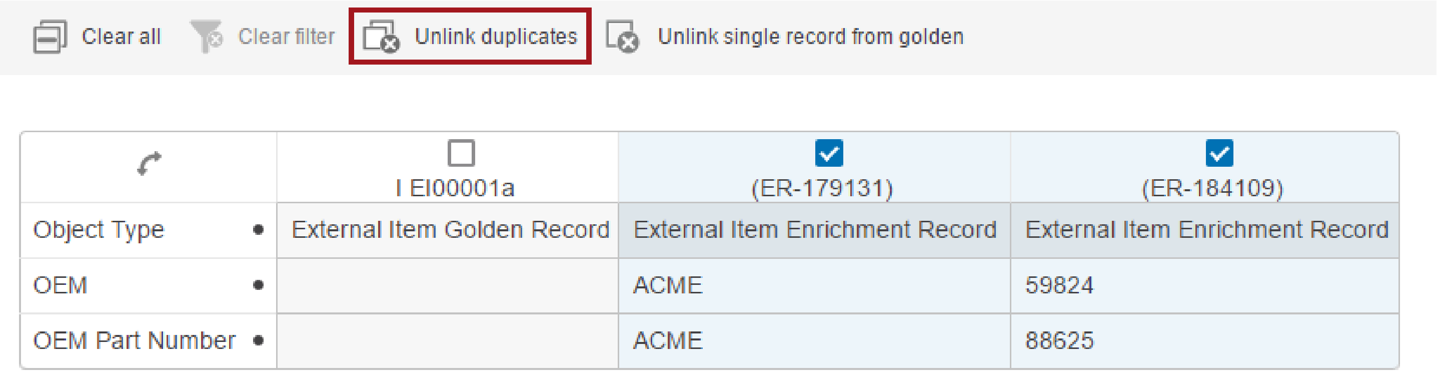
Actions:
- Merge Confirmed Match From Grid Action
- Unlink Duplicate From Grid Action
- Unlink Single Record From Golden Action

1. For the **Headers** parameter, specify the headers to display on the table by clicking **Add...** and selecting the desired header. Depending on the header selected, additional configuration steps may be required. Ideally, specify attributes most relevant to comparing records.
2. For the **Matching Algorithm** parameter (required), click the ellipsis button (**...**) and select the relevant matching algorithm.
3. For the **Property Direction** parameter, determine whether to display the data in a horizontally or vertically aligned list via the dropdown.
4. For the **Show Group Headers** parameter, when checked, display attribute group headers.
5. For the **Show Only Valid Attributes** parameter, when checked, display valid attributes only.
6. For the **Use Immediate Save** parameter, when checked, every edit prompts an immediate save.
7. For the **Dimensions** parameter, to change the standard dimensions of the grid, select 'Compare Display Mode Dimensions' from the dropdown and click the **Edit...** button. In the dialog, specify the height and width (in pixels) of the columns and rows.
8. For the **Enable Freeze Panes** parameter, when checked, the **Freeze panes** action button in the toolbar is enabled.
9. For the **Actions** parameter (required), click the **Add...** button and select 'Unlink Duplicates From Grid Action' to add an **Unlink Duplicates** action button.
10. For the **Actions** parameter (required), click the **Add...** button and select 'Unlink Single Record From Golden Action' to add an **Unlink single record from golden** action button.
11. For the **Actions** parameter (required), click the **Add...** button and select 'Merge Confirmed Match From Grid Action' to add a **Navigate to merge node screen** action button.

Using Action Buttons

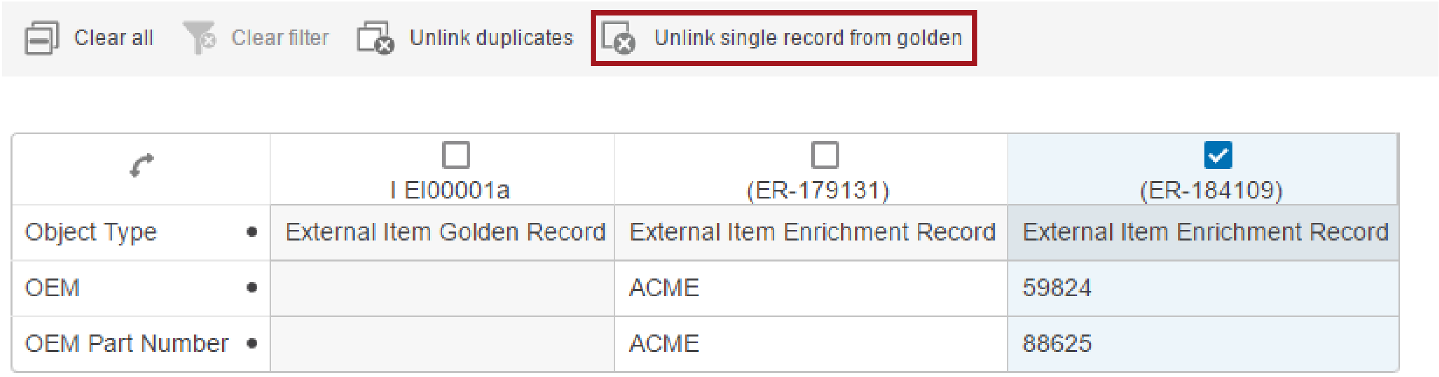
Once configured, the following explains the conditions required and expected outcome for each action.

1. The **Unlink duplicates** button requires that two source records are selected. Click the button to unlink the two records and mark them as confirmed non-duplicates. Only one record survives and remains linked to the golden record.



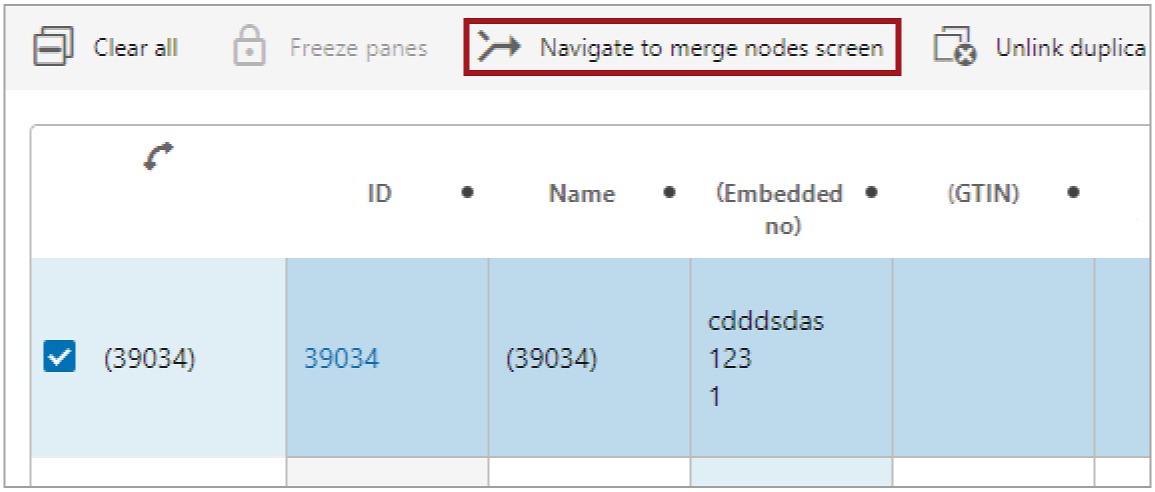
	□ I EI00001a	☑ (ER-179131)	☑ (ER-184109)
Object Type ●	External Item Golden Record	External Item Enrichment Record	External Item Enrichment Record
OEM ●		ACME	59824
OEM Part Number ●		ACME	88625

2. The **Unlink single record from golden** requires one source record to be selected. Click the button to remove the source record reference from the golden record.



	☐ I EI00001a	☐ (ER-179131)	☑ (ER-184109)
Object Type ●	External Item Golden Record	External Item Enrichment Record	External Item Enrichment Record
OEM ●		ACME	59824
OEM Part Number ●		ACME	88625

3. The **Navigate to merge node screen** button requires one source record to be selected. Click the button to proceed to the merge screen for the golden record.



	ID ●	Name ●	(Embedded no) ●	(GTIN) ●
☑ (39034)	39034	(39034)	cdddsdas 123 1	

Merging Confirmed Matches

With the Match and Link solution, two objects that have been confirmed as duplicates can be viewed on a Confirmed Matches component screen. The user decides whether to merge the duplicates or reject the confirmation and revert the objects back to potential duplicates.

Merging matched objects invokes a Merge Nodes screen where users choose the object to survive the merge, thereby assigning an object to be deleted, in addition to selecting the specific attribute values and outgoing references to be applied to the surviving record.

For more information, refer to the **Potential Duplicates List** topic.

For more information about how to merge confirmed matches via workbench, refer to the **Match and Link in Workbench** topic.

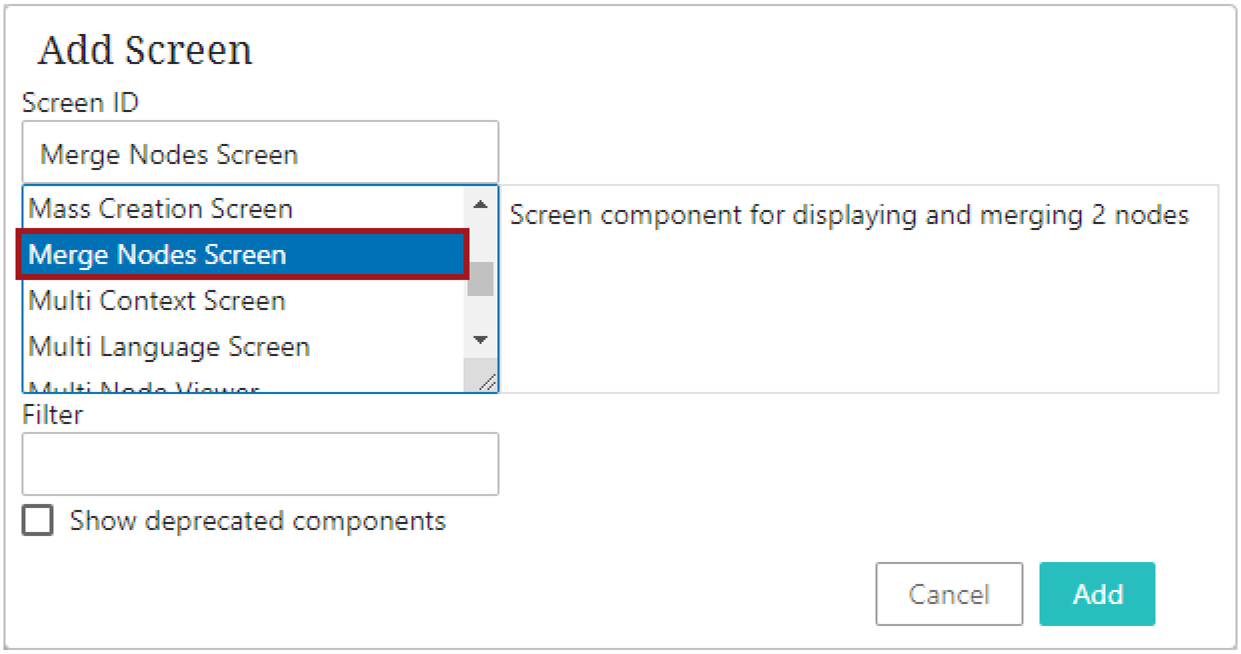
Note: Although the Merge Nodes Screen can be used to merge any nodes of the same super type, it is intended for use with matching algorithms and the Match Action for Identify Duplicates in conjunction with the Merge Confirmed Match From Grid action in the Confirmed Matches component.

Configuration

The following setup is required to enable the merge confirmed matches functionality in Web UI.

Add Merge Nodes Screen

1. In design mode, click the **New** button.
2. In the 'Add screen' window, select 'Merge Nodes Screen', enter a Screen ID, and click **Add**.



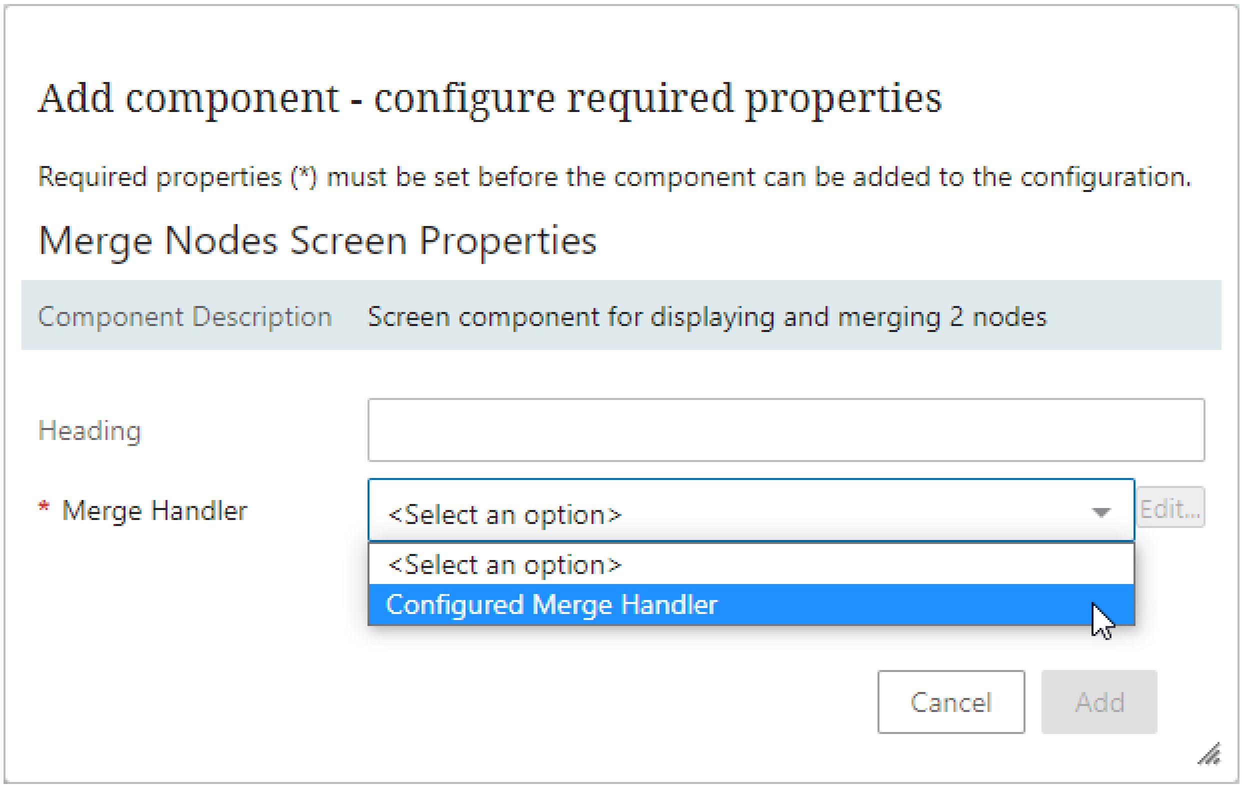
Add Screen

Screen ID: Merge Nodes Screen

Screen component for displaying and merging 2 nodes

Filter:

3. In the 'Merge Nodes Screen Properties' dialog, from the 'Merge Handler' dropdown select 'Configured Merge Handler' and click the **Add** button.



Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Merge Nodes Screen Properties

Component Description: Screen component for displaying and merging 2 nodes

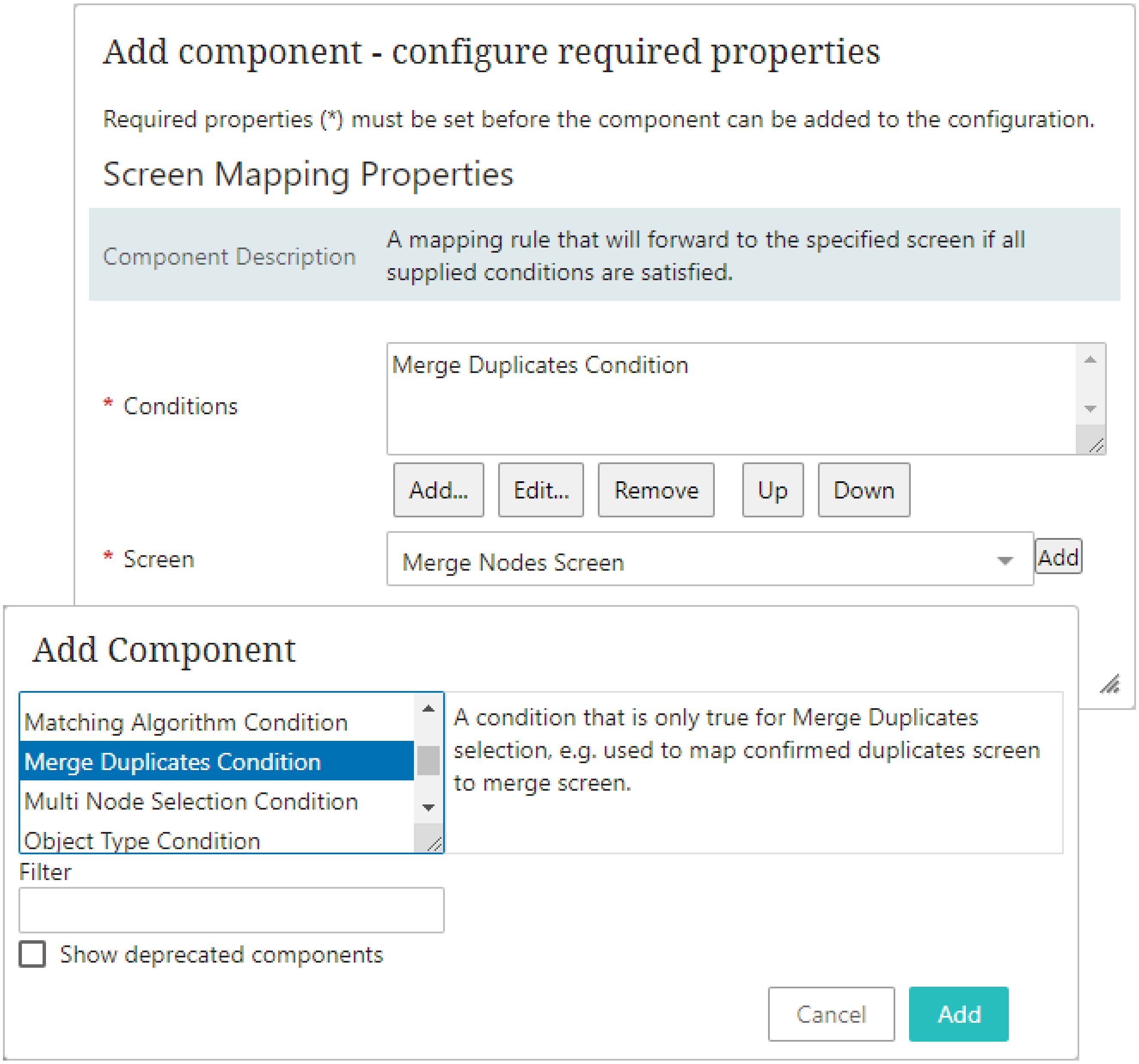
Heading:

* Merge Handler: <Select an option>

4. Click the **Save** button.

Map Merge Node Screen

1. In design mode, navigate to [MAIN].
2. Under 'Mappings', click **Add**.
3. On the Screen Mapping dialog, under 'Conditions', click **Add**, select 'Merge Duplicate Condition', and click **Add**.
4. From the 'Screen' dropdown, select the merge nodes screen configured in the previous section.
5. Click the **Add** button.



Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Screen Mapping Properties

Component Description: A mapping rule that will forward to the specified screen if all supplied conditions are satisfied.

* Conditions: Merge Duplicates Condition

* Screen: Merge Nodes Screen

Add Component

A condition that is only true for Merge Duplicates selection, e.g. used to map confirmed duplicates screen to merge screen.

Add...
Edit...
Remove
Up
Down
Add
Matching Algorithm Condition
Merge Duplicates Condition
Multi Node Selection Condition
Object Type Condition
Filter
□ Show deprecated components
Cancel
Add

6. Adjust the priority of the screen as needed. For more information, refer to the **Mappings** topic in the **Web User Interfaces** documentation.

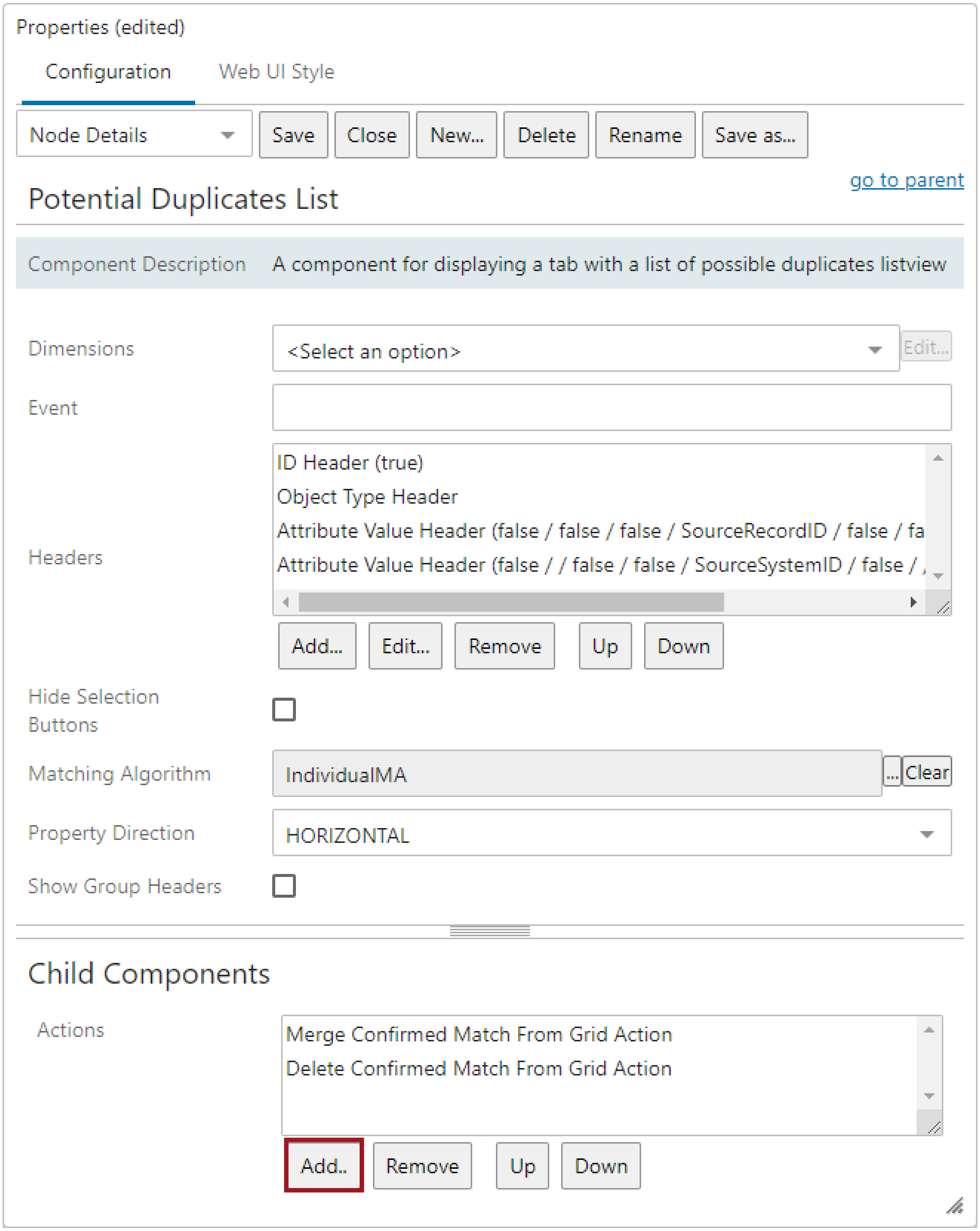
7. Click the **Save** button.

Add Merge Confirmed Match From Grid Action

Note: The Merge Confirmed Match action cannot be used on a Deduplication List screen.

1. In design mode, select a 'Potential Duplicates' component screen. If one does not exist, configure it on any Tab Page or Node Details component as defined in the **Potential Duplicates List** topic.

2. Under 'Child Components', in the 'Actions' parameter, click **Add**.



Potential Duplicates List

Component Description: A component for displaying a tab with a list of possible duplicates listview

Dimensions: <Select an option>

Event:

Headers:
ID Header (true)
Object Type Header
Attribute Value Header (false / false / false / SourceRecordID / false / fa
Attribute Value Header (false / / false / false / SourceSystemID / false / ,

Hide Selection Buttons: □

Matching Algorithm: IndividualMA

Property Direction: HORIZONTAL

Show Group Headers: □

Child Components

Actions:
Merge Confirmed Match From Grid Action
Delete Confirmed Match From Grid Action

3. In the Add Component window, select 'Merge Confirmed Match From Grid Action' and click **Add**.

4. Click the **Save** button.

Performing a Merge

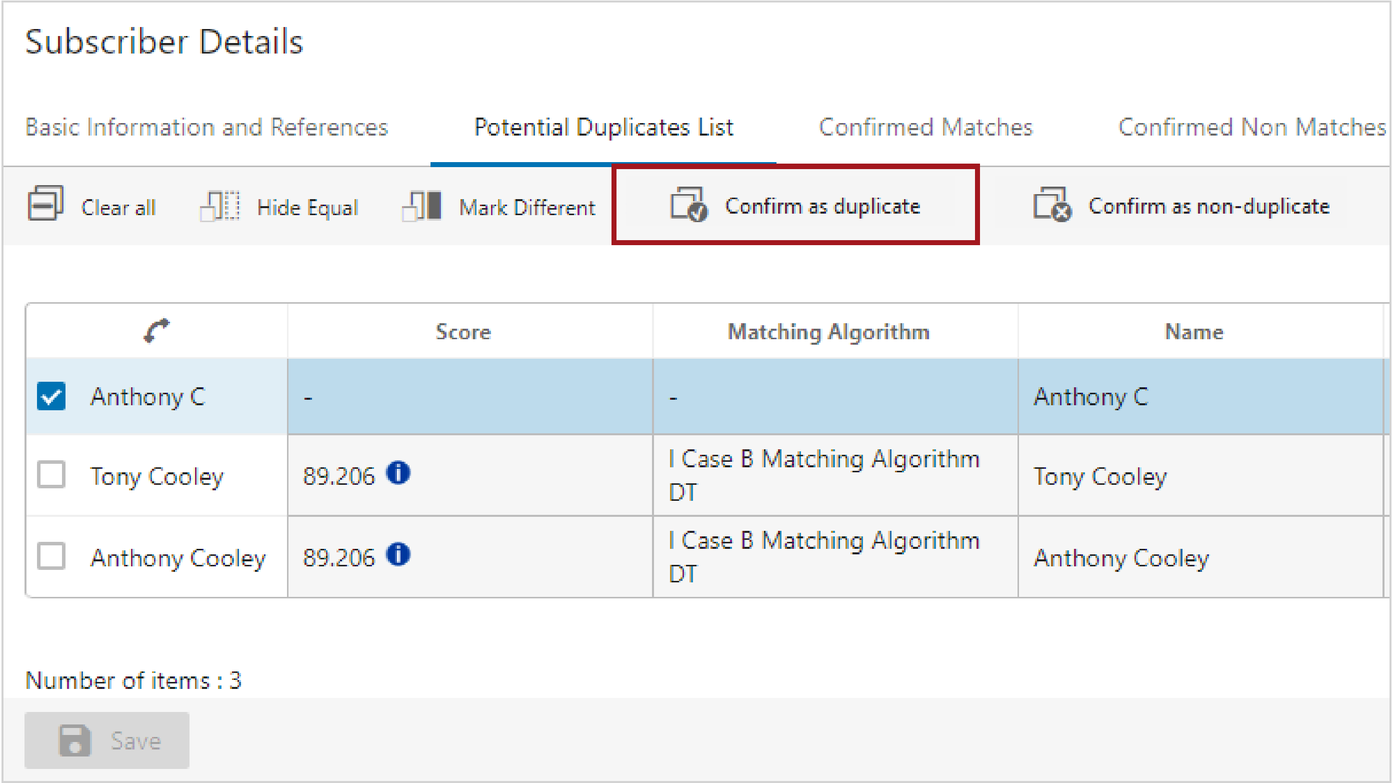
Important: Before beginning the merge process, review the following considerations:

- Metadata attributes, inherited attributes, and inbound references are not merged.
- If the object that remains after the merge contains no data in any context for a given attribute or reference, the data is taken from the deleted object and merged into the remaining object.
- All attributes and references eligible for merging are displayed in the table.

Use the following steps to perform a merge.

1. On a 'Potential Duplicates List' screen, choose the object to merge with the currently selected node.

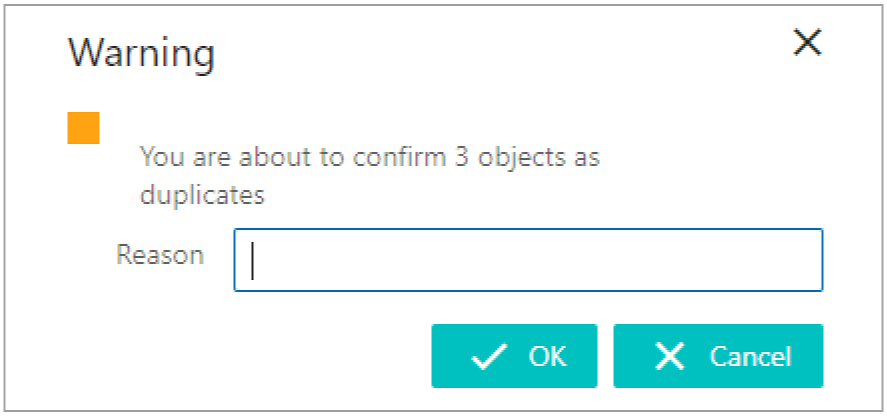
2. Click the 'Confirm as duplicate' button.



	Score	Matching Algorithm	Name
☑ Anthony C	-	-	Anthony C
☐ Tony Cooley	89.206 ℹ	I Case B Matching Algorithm DT	Tony Cooley
☐ Anthony Cooley	89.206 ℹ	I Case B Matching Algorithm DT	Anthony Cooley

Number of items : 3

3. Explain why the objects are duplicates and click **OK**.



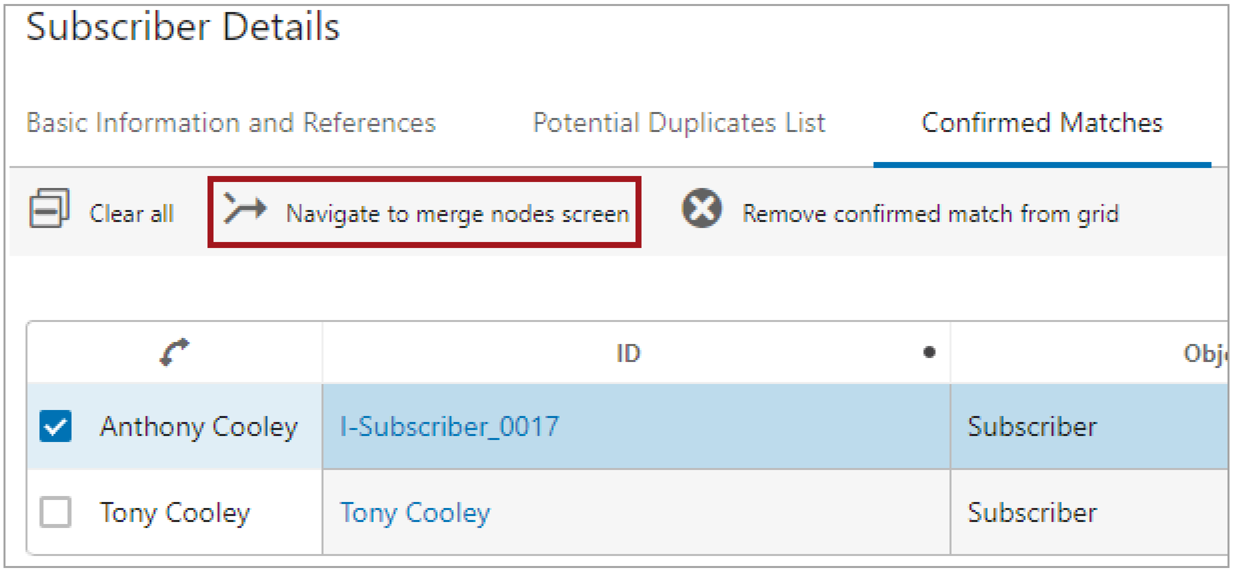
Warning

You are about to confirm 3 objects as duplicates

Reason

4. Navigate to the 'Confirmed Matches' component screen and select the object to be merged with the currently selected node.

5. Click the 'Navigate to merge nodes screen' button.



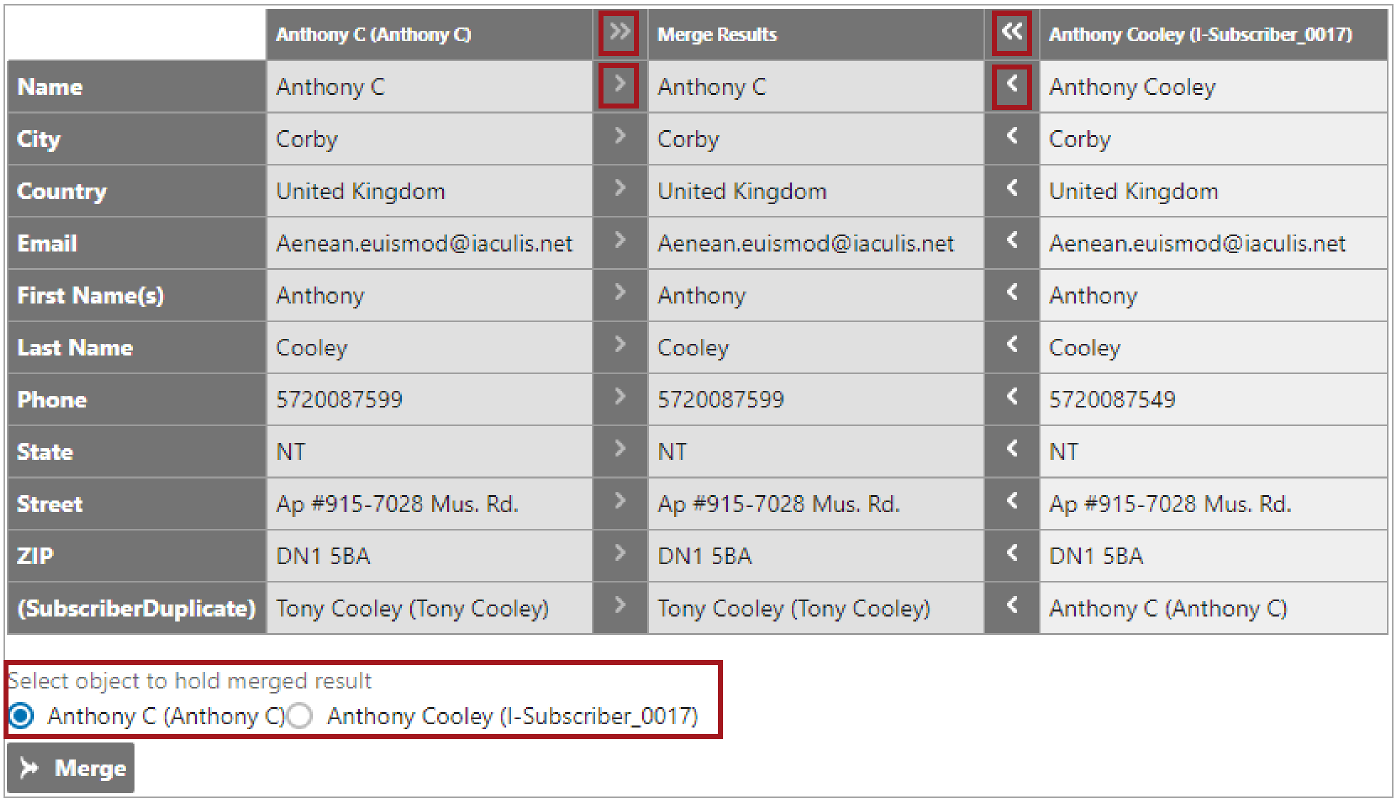
Subscriber Details

Basic Information and References
Potential Duplicates List
Confirmed Matches
Clear all
Navigate to merge nodes screen
Remove confirmed match from grid

	ID	Obj
☑ Anthony Cooley	I-Subscriber_0017	Subscriber
□ Tony Cooley	Tony Cooley	Subscriber

6. On the merge screen, click the arrow next to an element to choose the values for attributes and references to survive the merge.

> **Note:** Surviving attributes / references appear in the center 'Merge Results' column. Select the surviving node via the radio buttons that appear above the **Merge** button.



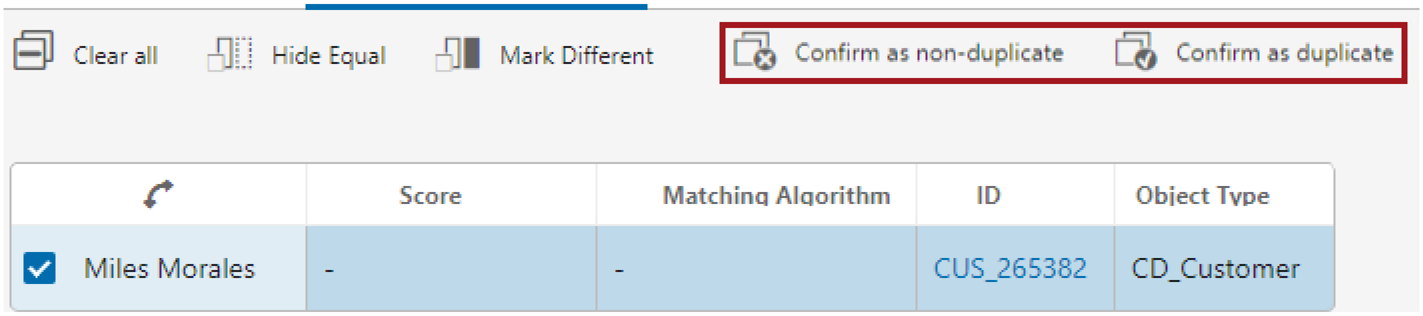
	Anthony C (Anthony C)	»	Merge Results	«	Anthony Cooley (I-Subscriber_0017)
Name	Anthony C	›	Anthony C	‹	Anthony Cooley
City	Corby	›	Corby	‹	Corby
Country	United Kingdom	›	United Kingdom	‹	United Kingdom
Email	Aenean.euismod@iaculis.net	›	Aenean.euismod@iaculis.net	‹	Aenean.euismod@iaculis.net
First Name(s)	Anthony	›	Anthony	‹	Anthony
Last Name	Cooley	›	Cooley	‹	Cooley
Phone	5720087599	›	5720087599	‹	5720087549
State	NT	›	NT	‹	NT
Street	Ap #915-7028 Mus. Rd.	›	Ap #915-7028 Mus. Rd.	‹	Ap #915-7028 Mus. Rd.
ZIP	DN1 5BA	›	DN1 5BA	‹	DN1 5BA
(SubscriberDuplicate)	Tony Cooley (Tony Cooley)	›	Tony Cooley (Tony Cooley)	‹	Anthony C (Anthony C)

Select object to hold merged result
● Anthony C (Anthony C) ○ Anthony Cooley (I-Subscriber_0017)

7. Click the **Merge** button to merge the two objects. The object that was not picked to survive is deleted but the deletion is not automatically approved.

Potential Duplicates List

In a match and link solution for deduplication, the matching threshold value can be set so that matches higher than the threshold are considered potential duplicates. With this configuration, Web UI users can work a list of potential duplicates and then confirm or reject each object as a duplicate. This is like the functionality offered in the workbench on the Matching tab for an object where a matching algorithm has been run.



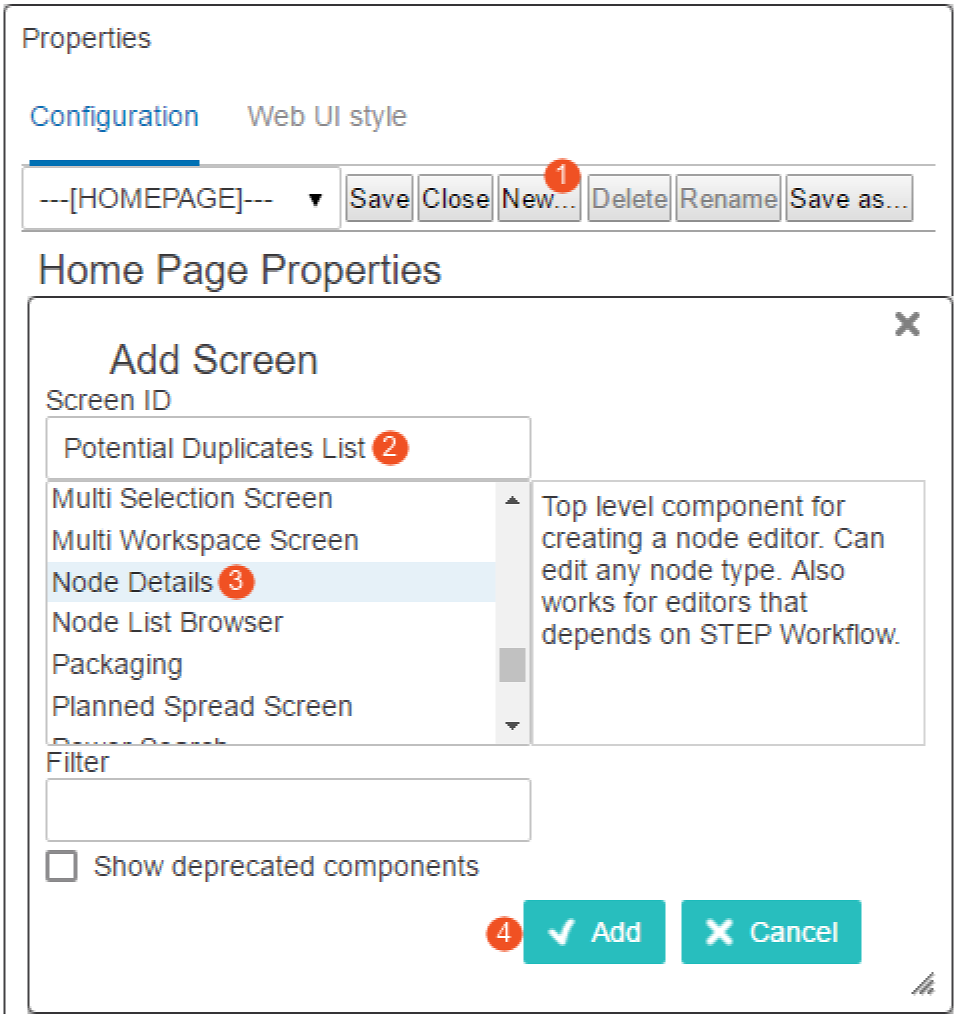
	Score	Matching Algorithm	ID	Object Type
☑ Miles Morales	-	-	CUS_265382	CD_Customer

Note: The Potential Duplicates List component uses a Match Score and Algorithm to identify potential duplicates. Create and configure these in the STEP Workbench before continuing with the configuration below. For more information, refer to the **Match and Link** documentation.

Configuring the Deduplication Table

The Potential Duplicates List component can be added to any Node Details or Tab Control / Tab Page component. Below are steps to configure the component using a Node Details screen.

1. In the Web UI designer, create a new screen, assign a Screen ID ('Potential Duplicates List' in this example), select the Node Details screen type, and click **Add**.



Properties

Configuration Web UI style

---[HOMEPAGE]--- ▼ Save Close New... 1 Delete Rename Save as...

Home Page Properties

Add Screen

Screen ID

Potential Duplicates List 2

Multi Selection Screen
Multi Workspace Screen
Node Details 3
Node List Browser
Packaging
Planned Spread Screen

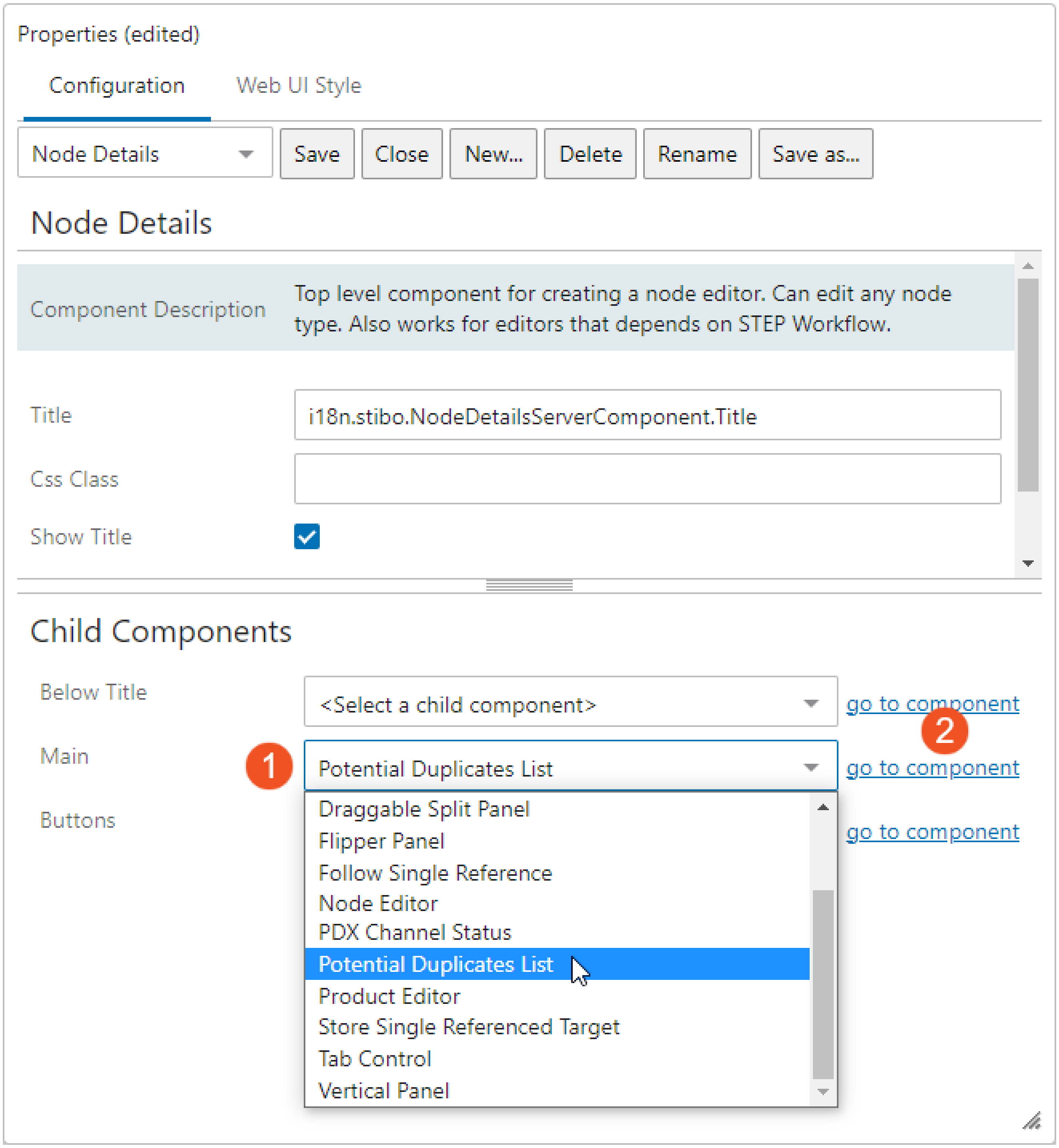
Top level component for creating a node editor. Can edit any node type. Also works for editors that depends on STEP Workflow.

Filter

□ Show deprecated components

4 ✓ Add ✕ Cancel

2. On the newly created screen, in the **Child Components** section click the **Main** dropdown menu, select **Potential Duplicates List**, and click the **go to component** link.



Node Details

Component Description: Top level component for creating a node editor. Can edit any node type. Also works for editors that depends on STEP Workflow.

Title: i18n.stibo.NodeDetailsServerComponent.Title

Css Class:

Show Title: ☑

Child Components

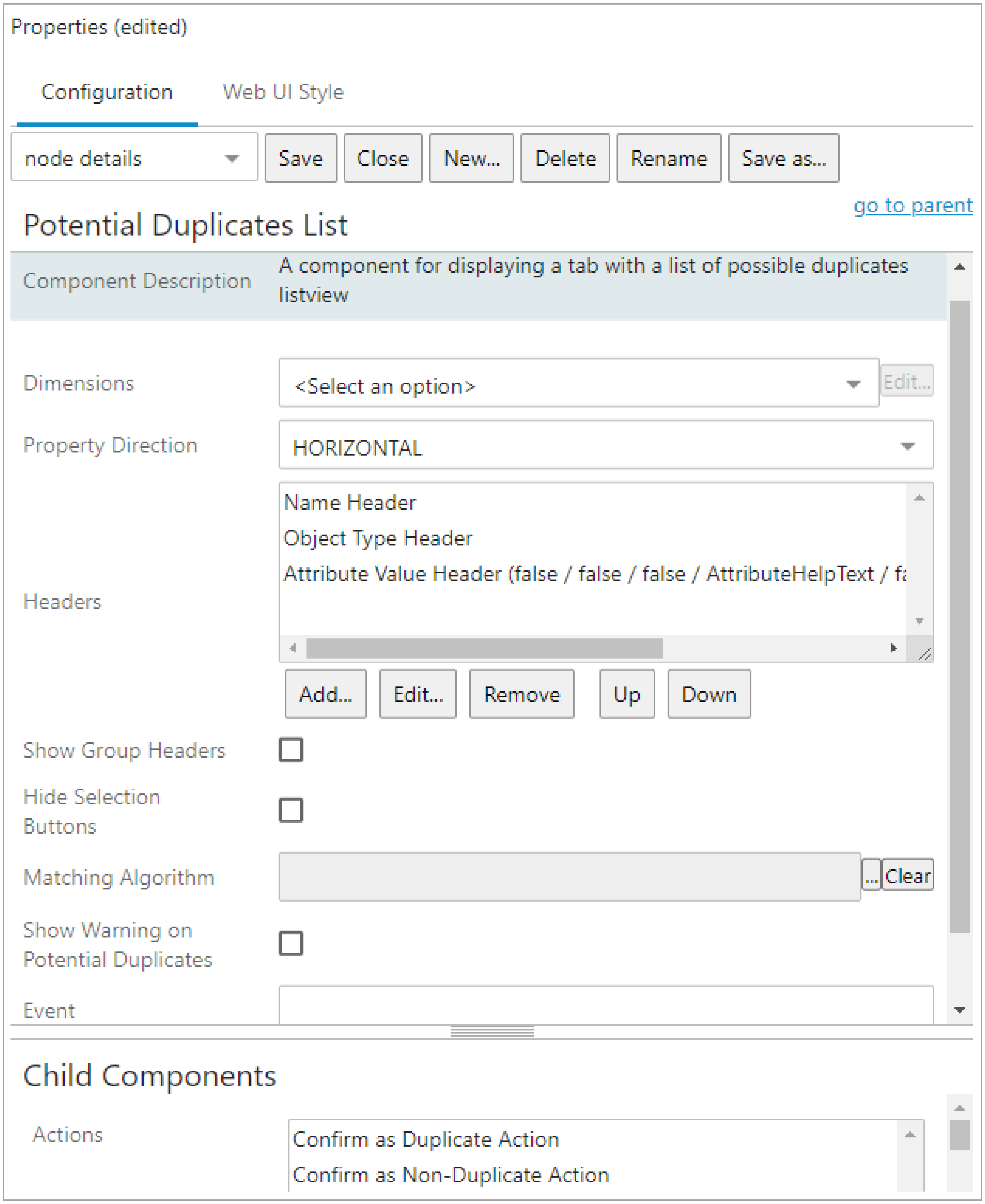
Below Title: <Select a child component> go to component

Main: Potential Duplicates List go to component

Buttons: go to component

3. On the Potential Duplicates List properties dialog, for the **Dimensions** parameter, optionally select Compare Display Mode Dimensions and click the **Edit** button to define height and width for the page. Leave this parameter at the default for automatic sizing.
4. If using auto-submit in the Clerical Review Task List screen (as defined in the **Configuring a Deduplication Clerical Review** topic), for the **Event** parameter, add the workflow event type to use after submission.

5. For the **Headers** parameters, click the **Add** button to select the information to be included in the table, like Name an Object Type. By default, the table includes the Score and Matching Algorithm headers followed by the other headers added manually.
6. For the **Matching Algorithm** parameter, optionally select an algorithm that determines how potential duplicates are identified. If no selection is made, results from all relevant algorithms are shown.



Potential Duplicates List

Component Description: A component for displaying a tab with a list of possible duplicates listview

Dimensions: <Select an option>

Property Direction: HORIZONTAL

Headers:
Name Header
Object Type Header
Attribute Value Header (false / false / false / AttributeHelpText / fa

Show Group Headers: □

Hide Selection Buttons: □

Matching Algorithm:

Show Warning on Potential Duplicates: □

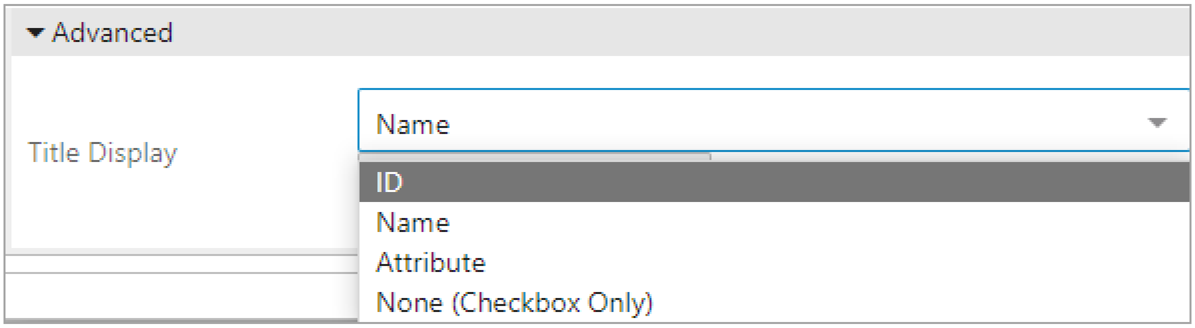
Event:

Child Components

Actions:
Confirm as Duplicate Action
Confirm as Non-Duplicate Action

7. For the **Show Warning on Potential Duplicates** parameter, optionally select if a yellow warning icon should display on the tab that shows potential duplicates which must be reviewed before submitting.

8. Under the **Advanced** flipper, in the **Title Display** dropdown menu, optionally select if the screen should display 'ID', 'Name', 'Attribute' or only a checkbox with no text added to it.



9. In the **Child Components** section, for the Actions parameter, add the **Confirm as Duplicate** and **Confirm as Non-Duplicate** actions.

   The 'Hide Equal' and 'Mark Different' actions automatically display before other manually configured actions. For more information, refer to the **Comparing Data Using Hide Equal and Mark Different** section of the **Web User Interfaces** documentation.

10. Map the screen to display as needed via the 'Merge Duplicate Condition' for the Node Details screen configured with the Potential Duplicate List (or other conditions as required). For more information, refer to the **Mappings** topic in the **Web User Interfaces** documentation.
11. Save and close the Web UI Design Mode.

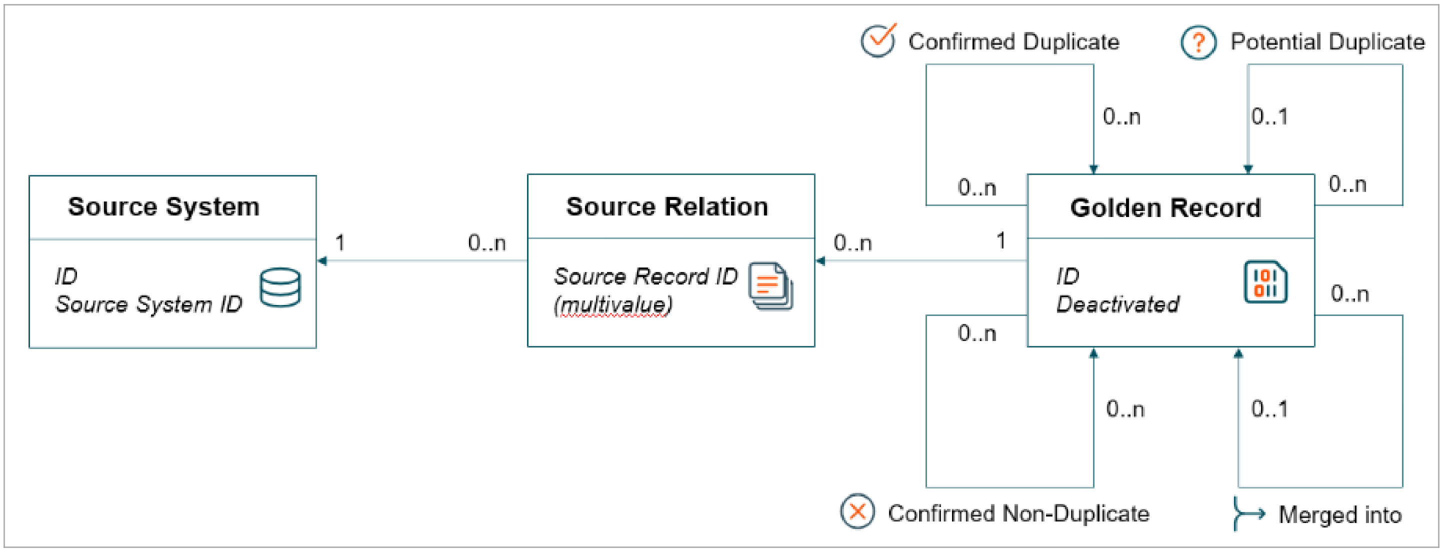
Match and Merge

A match and merge solution takes ownership over the data and is well suited to data hub implementations with any degree of centralized or decentralized management of data.

For details on configuration, refer to the **Match and Merge Traceability** topic and the **Configuring Match and Merge** topic.

In the following sections, an example of maintaining customer records in a match and merge solution is used to explain the match and merge data functionality.

Data Model



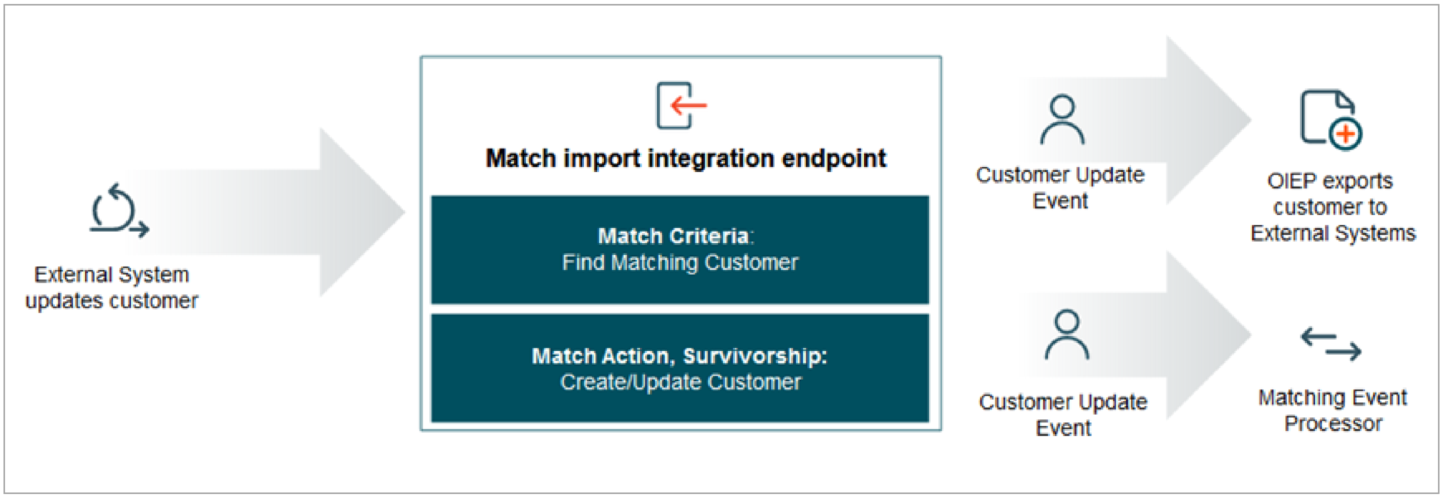
Unlike the Match and Link Match Action, in the Match and Merge Match Action the source record and golden record do not use separate object types. The source system is registered as an entity and the source relation is modeled as a reference from the golden record to that source system.

> **Note:** When consolidating data, you must use the match and merge solution, not the match and link solution.

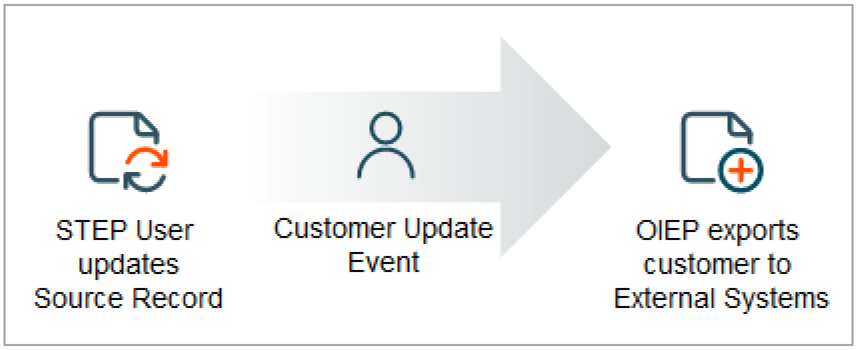
Information Flow

When a customer record is created or updated in an external system, the update is delivered to STEP via either a web service endpoint or an IIEP.

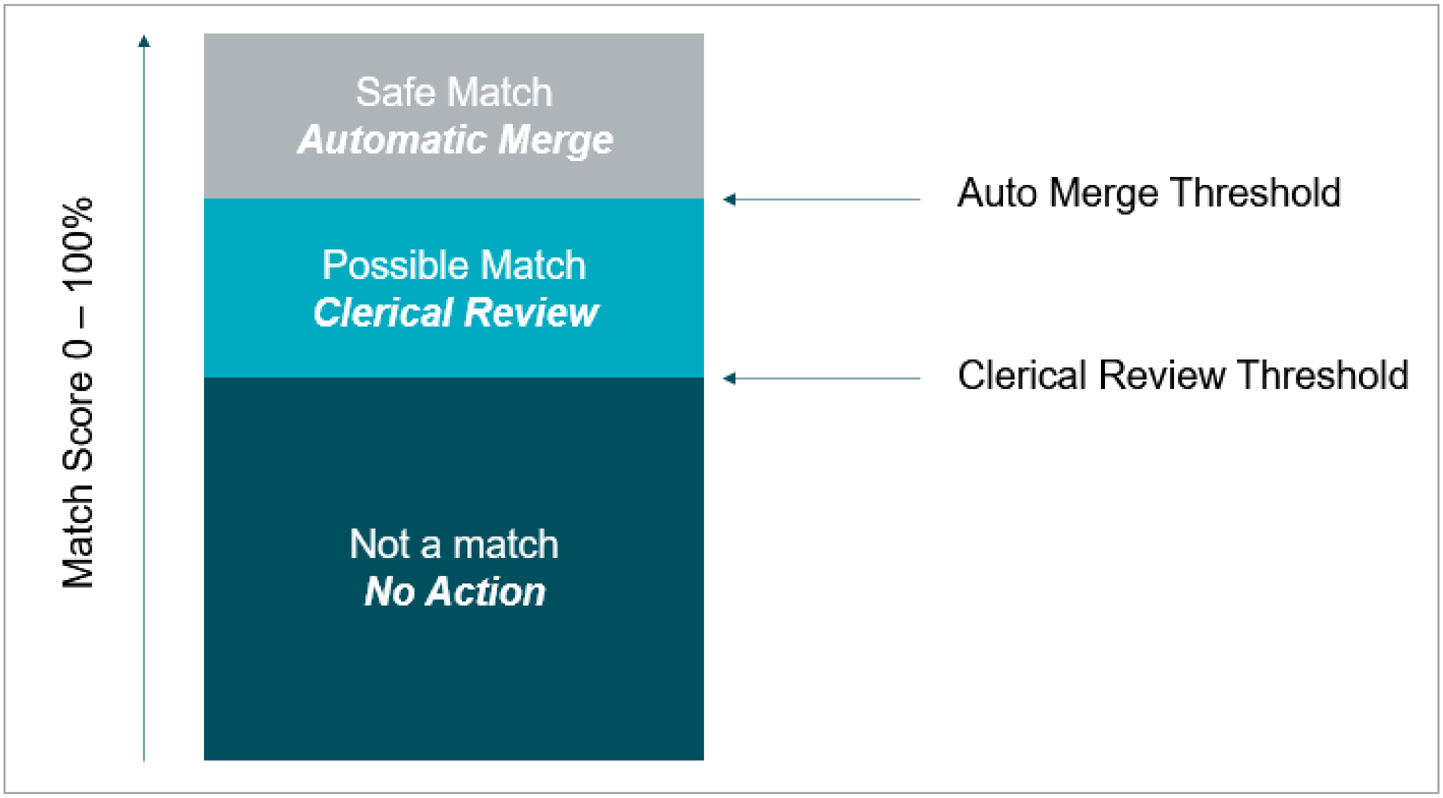
In both cases, the incoming source record is matched against the existing golden records, and if a match is found, the information from the source record is merged into the relevant golden record using survivorship rules. If this results in updated information, the customer record can be exported back to all external systems. In this way, an update to the customer record in any system can be automatically managed for trust and timeliness. This ensures the best possible view of the customer record is reflected across the entire ecosystem.


External System updates customer
Match import integration endpoint
Match Criteria: Find Matching Customer
Match Action, Survivorship: Create/Update Customer
Customer Update Event
OIEP exports customer to External Systems
Customer Update Event
Matching Event Processor

When a user updates the customer record in STEP, the update takes place on the golden record itself, and the new trusted record can be exported in the same way as before.

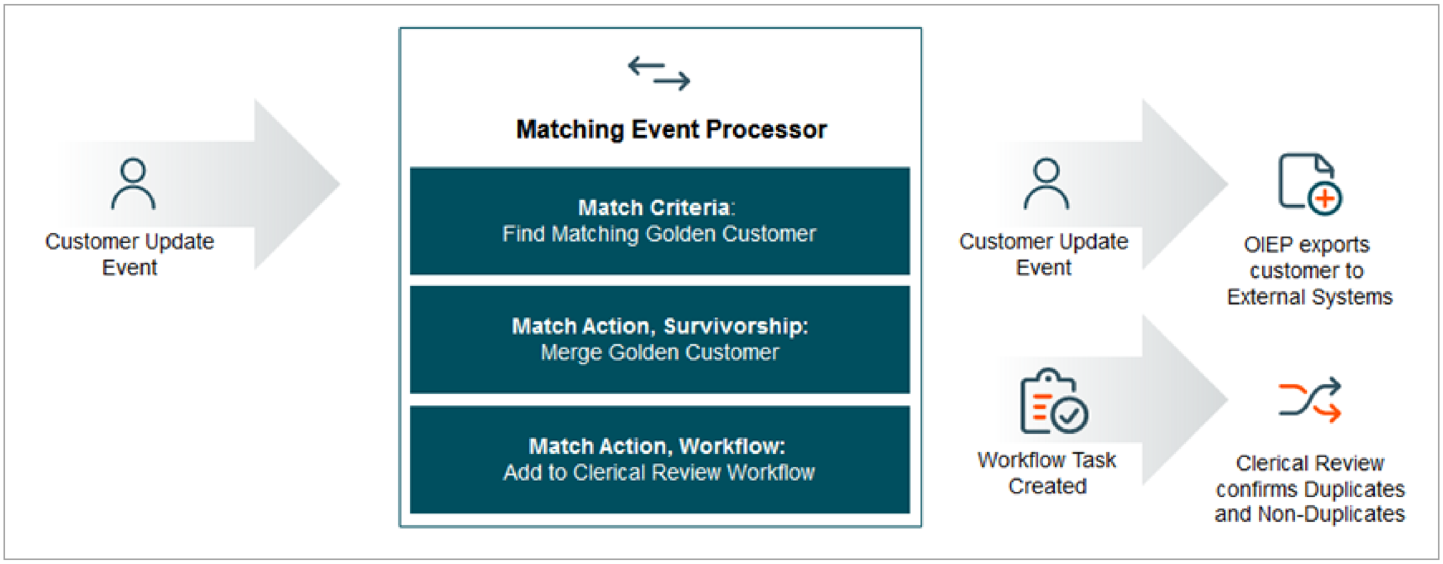

STEP User updates Source Record
Customer Update Event
OIEP exports customer to External Systems

The matching process uses a 'match score' within three groups separated by thresholds to indicate the likelihood of a match. For more information on match scores, refer to the **Match Scores** topic.

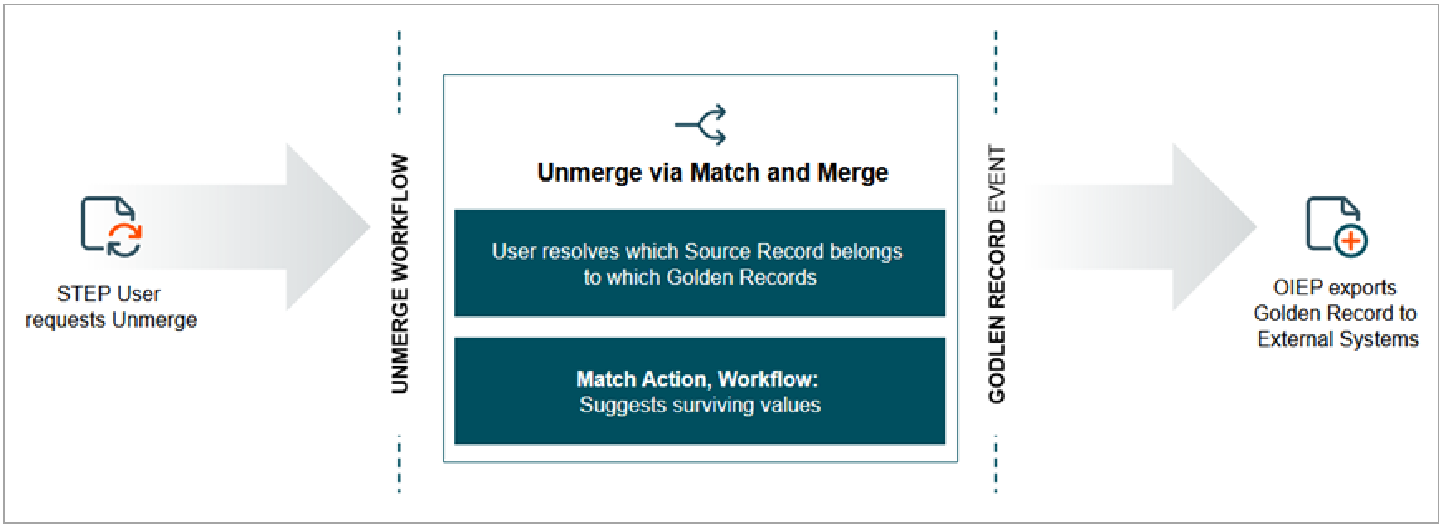


- A match score above the auto merge threshold (the highest threshold) is considered a match and the system automatically merges the data. During import, this results in the incoming data being merged directly into the existing golden record. If updates make two existing records match above the auto merge threshold, the matching algorithm declares one of the records as the 'survivor' and deactivates the other record. Information from the incoming or deactivated record is merged into the surviving record based on the survivorship rules set on the matching algorithm.
- A match score between the clerical review threshold and the auto threshold indicates a possible match. The two records are sent to the clerical review workflow so a user can determine if there is a match or not. The data steward manually confirms the two records are duplicates and merges them or confirms they are not duplicates and should be kept separate going forward.
- A match score below the clerical review threshold (the lowest threshold) is considered a non-match.

As golden records are created or updated, the matching event processor continuously compares the golden record to other golden records in the system.


Customer Update Event
Matching Event Processor
Match Criteria: Find Matching Golden Customer
Match Action, Survivorship: Merge Golden Customer
Match Action, Workflow: Add to Clerical Review Workflow
Customer Update Event
OIEP exports customer to External Systems
Workflow Task Created
Clerical Review confirms Duplicates and Non-Duplicates

Even in the best organizations, accidents happen. When two records are merged accidentally, STEP has tools to help resolve the issue. In a data hub that is closely integrated with a multitude of source systems, the process of unmerge may require a range of activities in the workflow in addition to the actual unmerge Web UI. The Web UI unmerge uses both original source records from source systems, revision history, and the match algorithm survivorship rules to help the user determine which values belong to which records during an unmerge.


STEP User requests Unmerge
UNMERGE WORKFLOW
Unmerge via Match and Merge
User resolves which Source Record belongs to which Golden Records
Match Action, Workflow: Suggests surviving values
GODLEN RECORD EVENT
OIEP exports Golden Record to External Systems

For detailed charts and explanations of how information flows in a match and merge solution, refer to the **Match and Merge Flow Details** topic.

Match and Merge Traceability

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

Match and merge is designed for the data hub, and as such, how records are identified by the source systems is important. If traceability is not configured, match and merge imports update data directly into golden records and discard the non-surviving data, making unmerging less effective. Configuring traceability retains source record information for better revision history and improved unmerge capabilities.

This topic includes how to:

- Configure source record data in Web UI (in the **Configure Traceability in Web UI** section)
- View source record data in workbench (in the **View Traceability in Workbench** section)

Additional traceability configuration is required as defined in the following topics:

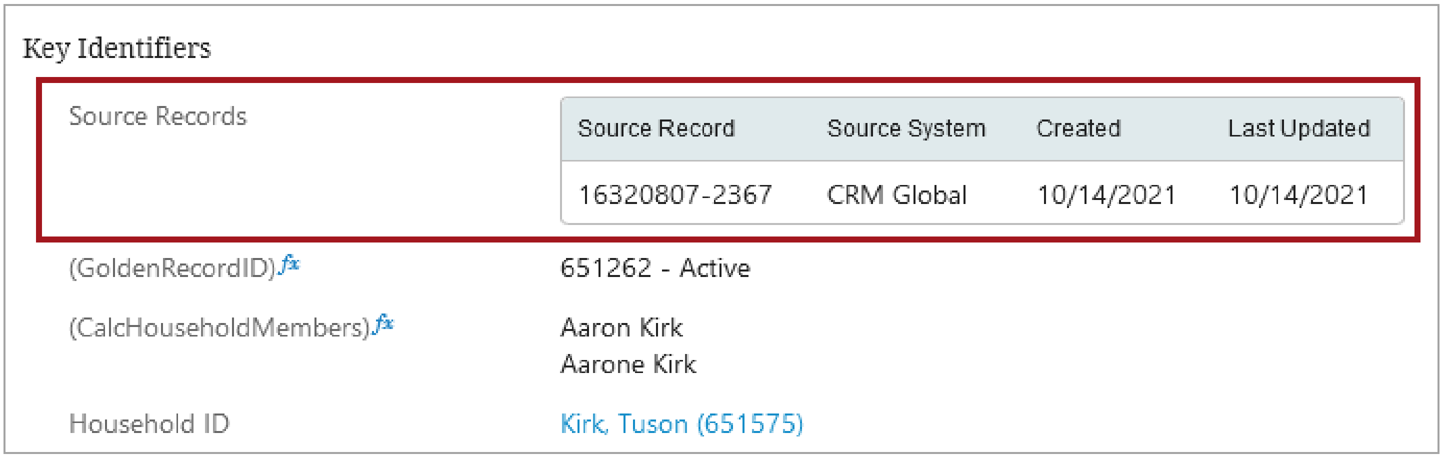
- **Storing Source Records for Golden Records**
- **Golden Record Source Traceability Screen**

Configure Traceability in Web UI

Configure the following component and screen to view traceability information in Web UI.

Golden Record Source Information

The 'Golden Record Source Information' component offers an overview of the golden record's history and the systems from which data was received. The default component label is Source Records but can be modified if desired. Once added as a child component on a node editor screen, no further configuration is required.



Key Identifiers

Source Records

Source Record	Source System	Created	Last Updated
16320807-2367	CRM Global	10/14/2021	10/14/2021

(GoldenRecordID)	651262 - Active
(CalcHouseholdMembers)	Aaron Kirk Aarone Kirk
Household ID	Kirk, Tuson (651575)

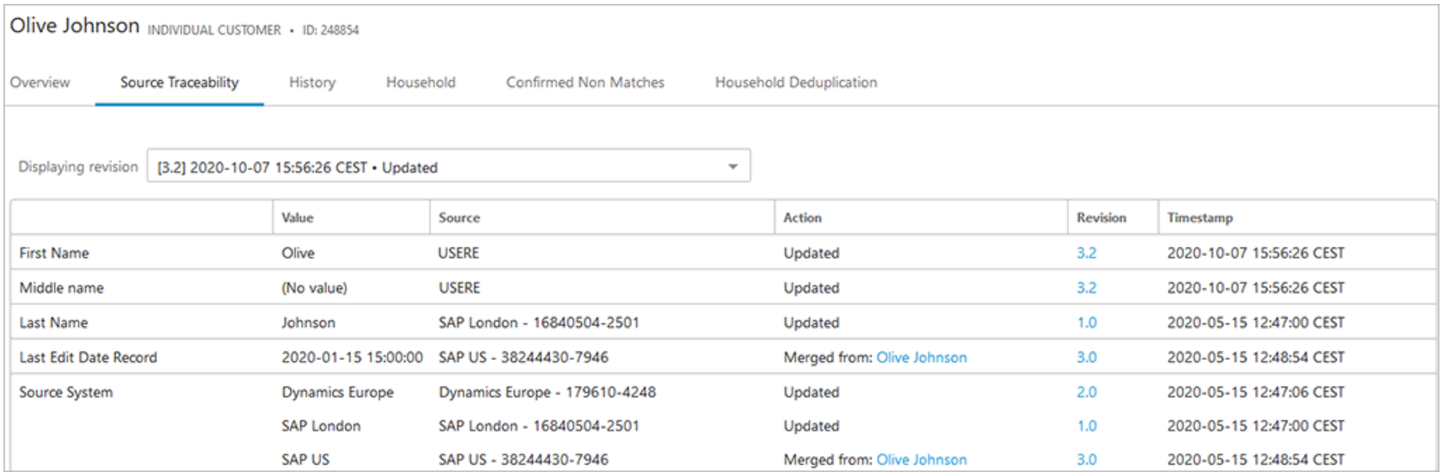
On a node editor screen for an entity, the child component displays:

- Source Record - the ID of the source record.
- Source System - the name of the source system from which the record originated.
- Created - the date the source record was created.
- Last Updated - the date the source record was last updated.

For more information, refer to the **Node Details Screen** topic of the **Web User Interfaces** documentation.

Golden Record Source Traceability Screen

The 'Golden Record Source Traceability Screen' offers a more comprehensive look at a golden record's revision history. It can be configured with header rows to display the values of attributes, attribute groups, data container attributes, and reference types. This allows the user to track changes to individual aspects of a golden record, it displays the system from which the new values originated, and it records when the changes were made.



Olive Johnson INDIVIDUAL CUSTOMER • ID: 248854

Displaying revision [3.2] 2020-10-07 15:56:26 CEST • Updated

	Value	Source	Action	Revision	Timestamp
First Name	Olive	USERE	Updated	3.2	2020-10-07 15:56:26 CEST
Middle name	(No value)	USERE	Updated	3.2	2020-10-07 15:56:26 CEST
Last Name	Johnson	SAP London - 16840504-2501	Updated	1.0	2020-05-15 12:47:00 CEST
Last Edit Date Record	2020-01-15 15:00:00	SAP US - 38244430-7946	Merged from: Olive Johnson	3.0	2020-05-15 12:48:54 CEST
Source System	Dynamics Europe	Dynamics Europe - 179610-4248	Updated	2.0	2020-05-15 12:47:06 CEST
	SAP London	SAP London - 16840504-2501	Updated	1.0	2020-05-15 12:47:00 CEST
	SAP US	SAP US - 38244430-7946	Merged from: Olive Johnson	3.0	2020-05-15 12:48:54 CEST

For more information, refer to the **Golden Record Source Traceability Screen** topic.

View Traceability in Workbench

Once the 'Matching - Merge Golden Record' component model configuration is complete, no additional configuration is required to display traceability in workbench.

On the revisions of the individual records, all merge and unmerge information is displayed in the 'Comment' parameter along with the Source System ID and Source Record ID.

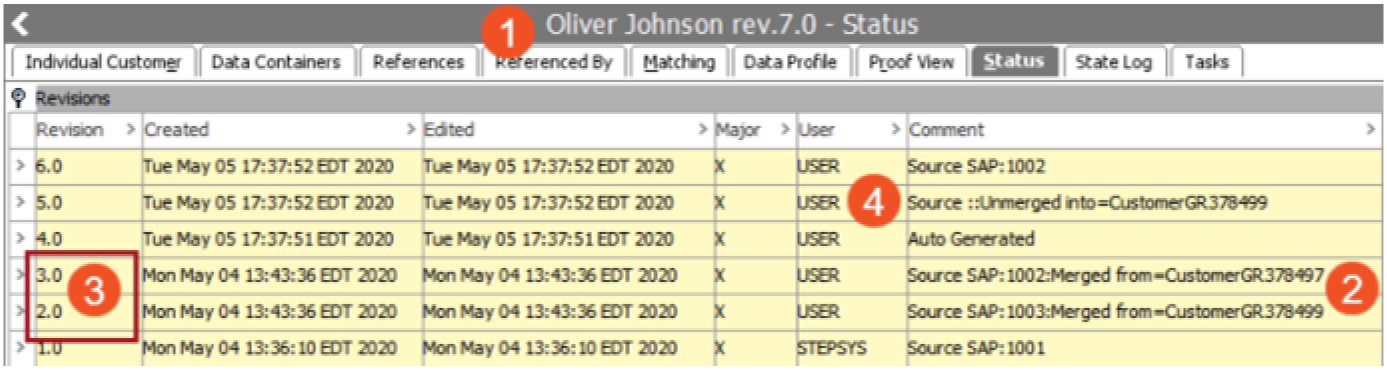
When merging, the surviving golden record has the ‘Merged into’ information with the object ID that was merged into this golden record.

When unmerging, the IDs of the reactivated or new golden records are listed in the ‘Unmerged into’ parameter.

Examples

Removing a Record from a Golden Record - This example shows a golden record with a record mistakenly merged into it and then unmerged.

1. **Golden Record Name -** Oliver Johnson
2. **Deactivated Golden Records -** CustomerGR378497 and CustomerGR378499
3. In Revision 2.0 and Revision 3.0, these two deactivated golden records are merged into the Oliver Johnson golden record, leaving the 'Merged from' traceability information.
4. In Revision 5.0, the CustomerGR378499 golden record is unmerged from the Oliver Johnson record and reactivated, leaving the 'Unmerged into' traceability information.

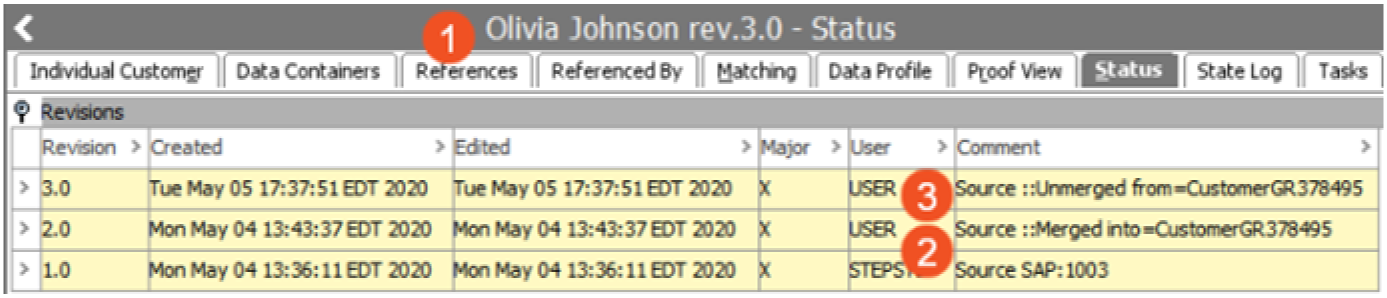


Revision	Created	Edited	Major	User	Comment
6.0	Tue May 05 17:37:52 EDT 2020	Tue May 05 17:37:52 EDT 2020	X	USER	Source SAP:1002
5.0	Tue May 05 17:37:52 EDT 2020	Tue May 05 17:37:52 EDT 2020	X	USER	Source ::Unmerged into=CustomerGR378499
4.0	Tue May 05 17:37:51 EDT 2020	Tue May 05 17:37:51 EDT 2020	X	USER	Auto Generated
3.0	Mon May 04 13:43:36 EDT 2020	Mon May 04 13:43:36 EDT 2020	X	USER	Source SAP:1002:Merged from=CustomerGR378497
2.0	Mon May 04 13:43:36 EDT 2020	Mon May 04 13:43:36 EDT 2020	X	USER	Source SAP:1003:Merged from=CustomerGR378499
1.0	Mon May 04 13:36:10 EDT 2020	Mon May 04 13:36:10 EDT 2020	X	STEPSYS	Source SAP:1001

On an active golden record, the 'Merged from' information is stored with the object ID of the golden record into which it was merged. When the golden record is reactivated in an unmerge operation, the 'Unmerged into' information is stored as a reference.

A Record's Removal from a Golden Record - This example shows how the removed record traces unmerging.

1. **Golden Record Name -** Olivia Johnson, CustomerGR378499
2. Olivia Johnson is merged into the active golden record - Oliver Johnson, CustomerGR378495.
3. In Revision 3.0, this merging was reversed and unmerged from the Oliver Johnson record which re-activates the Olivia Johnson golden record.



Revision	Created	Edited	Major	User	Comment
3.0	Tue May 05 17:37:51 EDT 2020	Tue May 05 17:37:51 EDT 2020	X	USER	Source ::Unmerged from=CustomerGR378495
2.0	Mon May 04 13:43:37 EDT 2020	Mon May 04 13:43:37 EDT 2020	X	USER	Source ::Merged into=CustomerGR378495
1.0	Mon May 04 13:36:11 EDT 2020	Mon May 04 13:36:11 EDT 2020	X	STEPSYS	Source SAP:1003

On the Olivia Johnson golden record, the Oliver Johnson record is stored as a reference of the 'Unmerged From' reference type.

Storing Source Records for Golden Records

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

During a Match and Merge operation, the imported data is often merged directly into golden records. Without configuring the 'Keep Source Records' option, this automated process discards data from different source systems. Once data is discarded during merge, unmerging is impossible because the new records created by the process are missing data.

Storage of source data is only supported on object types identified by the 'Matching - Merge Golden Record' component model.

The Matching - Merge Golden Record component model uses the following aspects to store source records imported with the source record ID and provide the unmerge functionality:

- Keep Source Records for Golden Record Object Types
- Source Record ID Attribute
- Source System ID Attribute

Removing a golden record object type from the component model does not delete source data that is already stored in the system. When an object type is deleted from the component model the system stops storing source data.

Source data includes a revision history and provides data lineage functionality. For more information, refer to the **Match and Merge Traceability** topic.

Considerations

Review the following when planning your configuration:

- Ensure all data container keys are defined satisfactorily. Changing key definitions later impacts the validity of the existing stored source data and creates issues because modified data container keys are incomplete or data container instances are duplicates. There is no method to identify data container source data as there is with golden records.
- After enabling the storage of source data, perform a full import of the source data either via IIEP or web service. Otherwise the source data in the system is incomplete and future partial updates will complicate the unmerge process. Without a full import, the system does not have a full dataset from each source.
- Storing source data increases the disk space used by the underlying storage system. The extent of the increase depends on the frequency of source record updates. The 'Source Record Data Management – Historical Values Cleanup' event processor (discussed below) works to limit the space used.

Storage Functionality

Source data storage include the following functionality:

- Source data is stored persistently in the system database and therefore it is included in standard backup procedures.
- Source data is excluded from In-Memory implementations.
- Source data storage is accumulative, meaning a source record can be updated by only sending part of the complete source dataset.
- Send an empty tag in STEPXML to delete an attribute value.
- No two records of the same object type in STEP should ever share the same Source Record ID for the same Source System.
- Source Systems may have several IDs on a single record in STEP.
- Different source systems are expected to assign different IDs to the same customer.

The following sections describe storage functionality upon import for multi-valued data containers and multi-valued references.

Multi-Valued Data Container without a defined Data Container Key

- All instances must be imported every time because existing instances are always replaced.
- Existing instances that are not part of the update are deleted.
- Applies only if that data container type is part of the import. If not, the existing instances are left unchanged.

Multi-Valued Data Containers with a defined Data Container Key

- Only instances with a matching key are updated.
- If no matching keys are found, a new instance is created.
- Existing data container instances cannot be deleted.

Multi-Valued References

- Instances of the reference are updated with respect to reference target.
- If no matching target is found, a new instance is created.
- Existing reference instances cannot be deleted.

Configuration

Complete the following workbench configuration:

1. In System Setup, open the Component Model node and select the 'Matching – Merge Golden Record' component model.

2. Verify an object type is selected on the 'Keep Source Records for Golden Record Object Types' aspect. If needed, modify the component model as defined in the **Configuring the Matching - Merge Golden Record Component Model** topic.

3. Create and configure event processing plugin 'Source Record Data Management – Historical Values Cleanup' as defined in the **Event Processors** topic of the **System Setup** documentation.

4. Perform a full import of the source data either via IIEP or web service so the full dataset from each source is available for the unmerge process.

Maintenance

Once data has been stored, if needed, purge source data via Bulk Update as defined in the **Merge Golden Records: Purge Source Data Operation** topic of the **Bulk Updates** documentation.

Configuring Match and Merge

The Match and Merge setup uses a component model, an object type for golden records, a matching algorithm with match action and survivorship rules, and an event processor. These elements work together to identify potentially duplicate records and to ultimately provide golden records that hold the best data from your source records.

Prerequisites

1. Complete the one-time setup defined in the **Initial Setup for Matching Algorithms** topic.

2. Configure one or more matching algorithms, as defined in the **Configuring Matching Algorithms** topic.

3. Complete the one-time setup defined in the **Initial Setup for Match Tuning** topic.

4. Configure a match tuning configuration, as defined in the **Configuring Match Tuning** topic.

5. Review the traceability information for unmerge and create the required attributes, as defined in the **Match and Merge Traceability** topic.

Configure a Merge Solution

Use the following steps to configure your merging solution.

1. Configure the Matching component model, as defined in the **Configuring Matching Component Model** topic.

2. Configure the Merge Golden Record object type, as defined in the **Configuring the Merge Golden Record Object Type** topic.

3. Configure the Matching - Merge Golden Record component model, as defined in the **Configuring the Matching - Merge Golden Record Component Model** topic.

4. Configure the match criteria, as defined in the **Match Criteria** topic.

5. Create the clerical review workflow, as defined in the **Match and Merge Clerical Review - Merge** topic.

6. Determine the unmerge method to be ad hoc or via workflow as defined in **Match and Merge Clerical Review - Unmerge** topic.

7. Create the merge action handlers, as defined in the **Creating Merge Golden Record Match Action Handlers** topic.

8. Configure the merge golden record match action, as defined in the **Configuring Merge Golden Record Match Action** topic.

9. Set up survivorship rules, as defined in the **Survivorship in Match and Merge** topic.

10. Determine and configure the data exchange method, as defined in the **Configuring the Match Data Exchange Method** topic.

11. Set up an event processor, as defined in the **Configuring the Merge Event Processor** topic.

12. Set up and learn to use Web UI for merging and unmerging, as defined in the **Configuring and Using Match and Merge in Web UI** topic.

Configuring Matching Component Model

The Matching component model specifies the object types shared by all defined matching types. Other individual matching component models further specify object types for the specific matching being performed, such as the matching defined in the Match and Link topic or the Match and Merge topic.

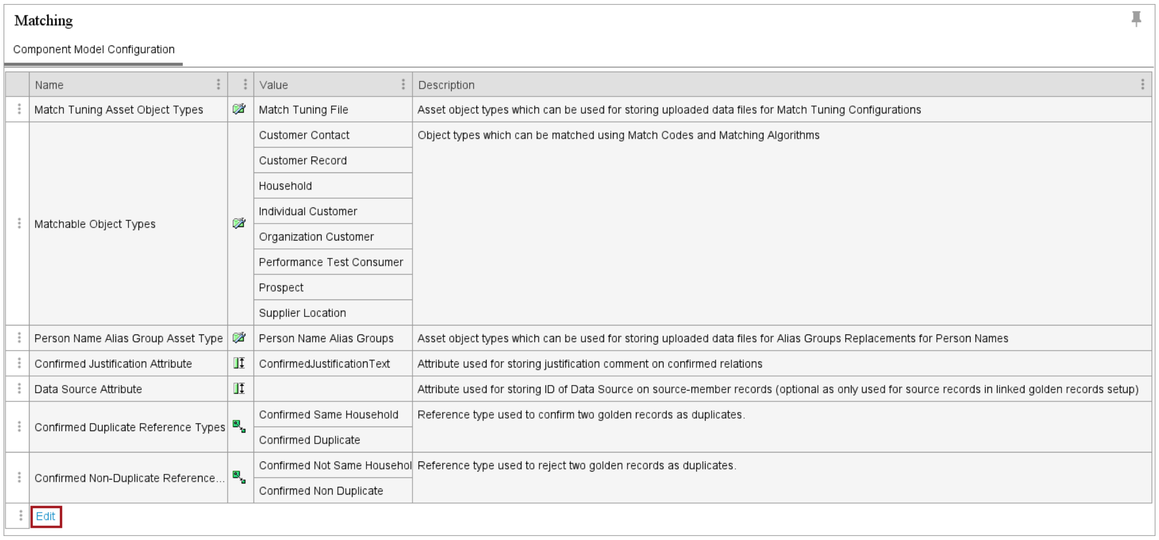
Prerequisites

Create all relevant object types, attributes, and references to make them available for selection in the component model.

Configuration

To configure the component model:

1. In System Setup, open the Component Models node and click the **Matching** component. The Component Model Configuration editor displays the aspects of the matching component.

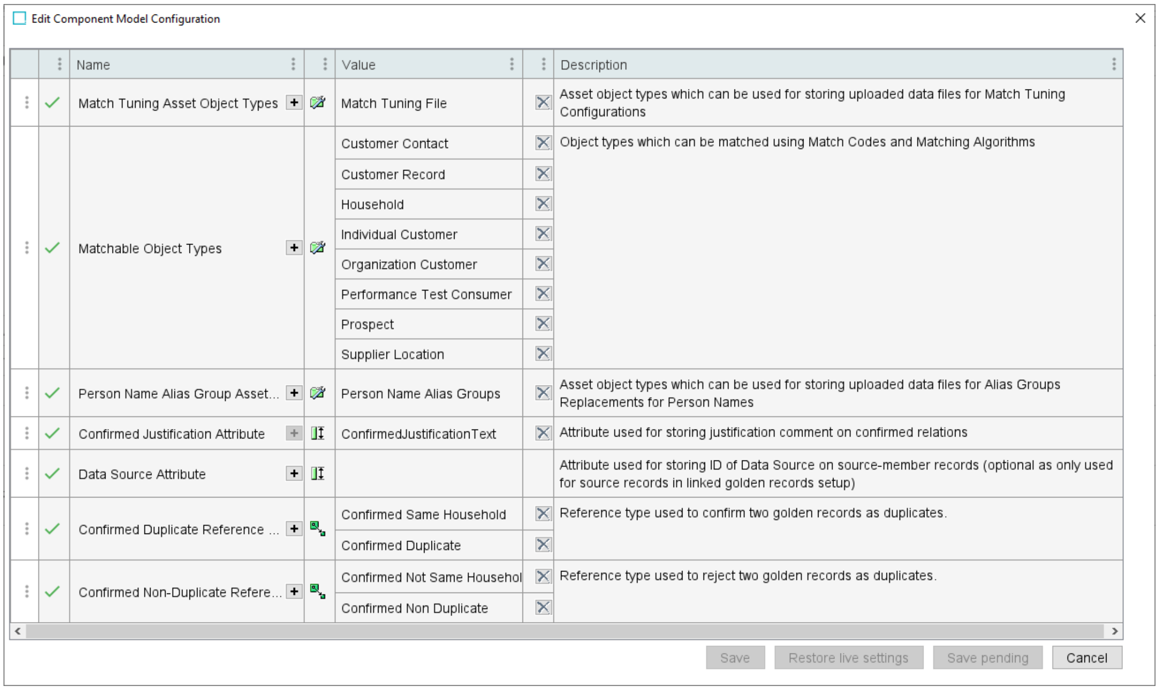


Matching

Component Model Configuration

Name	Value	Description
Match Tuning Asset Object Types	Match Tuning File	Asset object types which can be used for storing uploaded data files for Match Tuning Configurations
Matchable Object Types	Customer Contact Customer Record Household Individual Customer Organization Customer Performance Test Consumer Prospect Supplier Location	Object types which can be matched using Match Codes and Matching Algorithms
Person Name Alias Group Asset Type	Person Name Alias Groups	Asset object types which can be used for storing uploaded data files for Alias Groups Replacements for Person Names
Confirmed Justification Attribute	ConfirmedJustificationText	Attribute used for storing justification comment on confirmed relations
Data Source Attribute		Attribute used for storing ID of Data Source on source-member records (optional as only used for source records in linked golden records setup)
Confirmed Duplicate Reference Types	Confirmed Same Household Confirmed Duplicate	Reference type used to confirm two golden records as duplicates.
Confirmed Non-Duplicate Reference...	Confirmed Not Same Househol Confirmed Non Duplicate	Reference type used to reject two golden records as duplicates.

2. Click the **Edit** link (or the **Edit (pending changes)** link) to display the Edit Component Model Configuration dialog.



Edit Component Model Configuration

	Name		Value		Description
✓	Match Tuning Asset Object Types	+	Match Tuning File	☒	Asset object types which can be used for storing uploaded data files for Match Tuning Configurations
✓	Matchable Object Types	+	Customer Contact Customer Record Household Individual Customer Organization Customer Performance Test Consumer Prospect Supplier Location	☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒	Object types which can be matched using Match Codes and Matching Algorithms
✓	Person Name Alias Group Asset...	+	Person Name Alias Groups	☒	Asset object types which can be used for storing uploaded data files for Alias Groups Replacements for Person Names
✓	Confirmed Justification Attribute	+	ConfirmedJustificationText	☒	Attribute used for storing justification comment on confirmed relations
✓	Data Source Attribute	+			Attribute used for storing ID of Data Source on source-member records (optional as only used for source records in linked golden records setup)
✓	Confirmed Duplicate Reference ...	+	Confirmed Same Household Confirmed Duplicate	☒ ☒	Reference type used to confirm two golden records as duplicates.
✓	Confirmed Non-Duplicate Refere...	+	Confirmed Not Same Househol Confirmed Non Duplicate	☒ ☒	Reference type used to reject two golden records as duplicates.

To edit an aspect:

- Double click the plus button (⊞) on an aspect to display the 'Select ... for aspect' dialog and select an object type, attribute, or reference type. The button remains active for aspects that allow multiple selections.
- Double click the delete button (☒) to remove a selection.

A green check (✓) means the aspect has no errors; a red X (✗) means additional setup is required. Hover over the X for additional information.

3. For each of the following aspects choose to add object(s), attribute(s), or reference(s), and click the **Select** button.
   - **Match Tuning Asset Object Types** – Select the object types to store the input data for match tuning.
   - **Matchable Object Types** – Select the object types that need to be matched. Only the object types configured can be used as object types for match codes. On objects of these types, the 'Matching' tab is automatically enabled. The 'Matching' tab shows match code values, potential duplicates, and confirmed relations for the selected object.
   - **Person Name Alias Group Asset Type** – Select the asset object types to store uploaded data files for person name alias groups. These asset object types can be used with the Machine Learning Matcher and the Person Name and Address Match Code Generator. For more information, refer to theMatcher: Machine Learning Matcher topic and the Match Code Generator: Person Name and Address topic, both in the Matching, Linking, and Merging documentation.
   - **Confirmed Justification Attribute** – Select a description attribute valid for all reference types specified in the 'Duplicate Reference Types' and 'Non-Duplicate Reference Types' fields. This attribute stores a description explaining why two objects are marked as duplicates or non-duplicates in a match and link solution.

- **Data Source Attribute** – Select one or more description attributes valid for all source object types specified in the 'Source Object Types' field. This attribute contains the source ID of the source objects. If you select more than one attribute in this field, then exactly one of these attributes must be valid per source object type chosen in the 'Source Object Types' field. This field is only required for Link Golden Records solutions with **Trusted Source** survivorship rules configured.
- **Duplicate Reference Types** – Select one or more reference types to store the manually maintained confirmed duplicate references. These references store the reason for confirming two objects as duplicates specified in the attribute selected in the 'Confirmed Justification Attribute' field. All the selected reference types must have exactly one valid attribute from the 'Confirmed Justification Attribute' field. Only the duplicate reference types you select can be used as 'Duplicate Type' on a matching algorithm. In a typical scenario, you will have different duplicate reference types for different matching algorithms. If you reuse duplicate reference type between algorithms, the confirmed duplicates will be reused between those algorithms. Confirmed duplicate references are used in match and link solutions.
- **Non-Duplicate Reference Types** – Select one or more reference types used by the system for storing the manually maintained confirmed non-duplicate references. These references store the reason for confirming two objects as non-duplicates specified in the attribute selected in the 'Confirmed Justification Attribute' field. All the selected reference types must have exactly one valid attribute from the 'Confirmed Justification Attribute' field. Only reference types selected can be used as 'Non-Duplicate Type' on a matching algorithm. In a typical scenario, you will have different duplicate reference types for different matching algorithms. If you reuse the non-duplicate reference type between algorithms, the confirmed non-duplicates will be reused between those algorithms as well.

4. Save or cancel your work:

   - Click the **Save** button to save a configuration once it has no errors.
   - When enabled, click the **Save pending** button to save your work while errors exist.
   - When enabled, click the **Restore live settings** button to undo the changes made to a previously error-free, saved configuration.
   - Click the **Cancel** button to undo all changes made in this dialog.

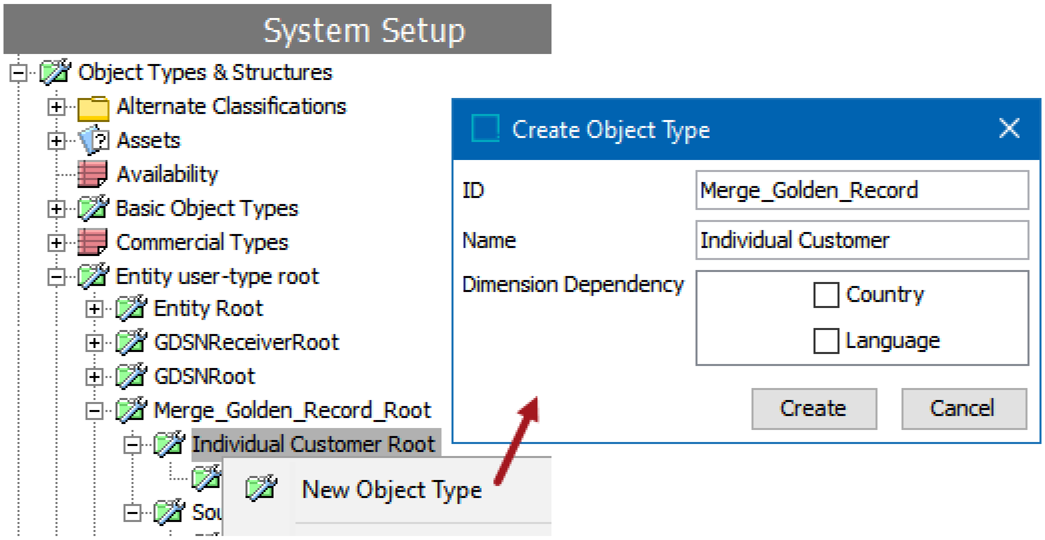
Configuring the Merge Golden Record Object Type

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

Golden records must be configured before being mapped to the component model and cited in a match action configuration.

To create a 'merge golden record' object type:

1. In System Setup, open the Object Types & Structure node, right-click on the node that identifies the type of golden record object (product or entity), and select the **New Object Type** option. In this example, the golden record is an 'Merge_Golden_Record' entity.
   - Add an **ID** and a **Name**.
   - Set **Dimension Dependency** as necessary.
   - Click the **Create** button.

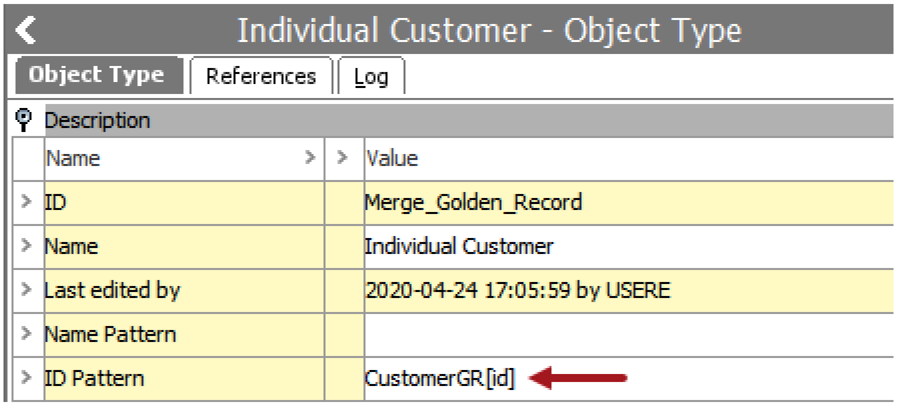


ID: Merge_Golden_Record

Name: Individual Customer

Dimension Dependency: □ Country □ Language

2. On the Description flipper, set the **ID Pattern** parameter to use the **[id]** variable. Refer to the **Autogenerate Using Name Pattern and ID Pattern** topic in the **System Setup** documentation.



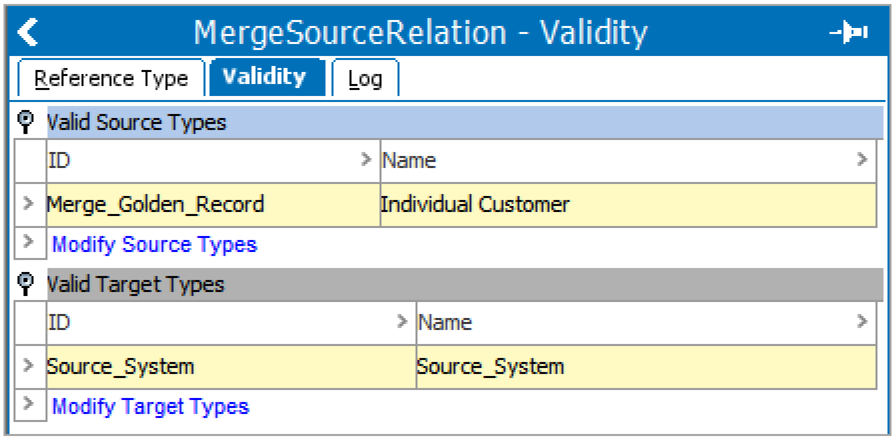
Individual Customer - Object Type

Description

Name	Value
ID	Merge_Golden_Record
Name	Individual Customer
Last edited by	2020-04-24 17:05:59 by USERE
Name Pattern	
ID Pattern	CustomerGR[id]

4. Verify that the reference type for linking 'source records' with 'merge golden records' has the following settings:

   - On the Reference Type tab, set the **Allow multiple references** parameter to 'Yes.'
   - On the Validity tab, under the **Valid Source Types** flipper add the golden record object type (such as ID=Merge_Golden_Record).
   - On the Validity tab, under the **Valid Target Types** flipper add to the source object type (such as ID=Source_System).



MergeSourceRelation - Validity

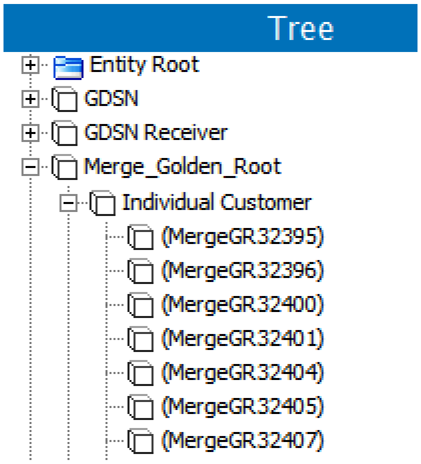
Valid Source Types

ID	Name
Merge_Golden_Record	Individual Customer
Modify Source Types	

Valid Target Types

ID	Name
Source_System	Source_System
Modify Target Types	

5. In Tree, create a root node for the merge golden records. Initially, all merge golden records will be created as children of this node.



Tree

- Entity Root
- GDSN
- GDSN Receiver
- Merge_Golden_Root
  - Individual Customer
    - (MergeGR32395)
    - (MergeGR32396)
    - (MergeGR32400)
    - (MergeGR32401)
    - (MergeGR32404)
    - (MergeGR32405)
    - (MergeGR32407)

Configuring the Matching - Merge Golden Record Component Model

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

The 'Matching - Merge Golden Record' component model identifies the golden record object types, references, and attributes applicable to the merge and unmerge golden record solution.

Prerequisites

The following tables identify the required settings on the objects needed for a successful match and merge solution.

Object Type	Revisability	Reference Target Lock Policy	Dimension Dependencies
Source System Object Type	Global Revisable	Relaxed	None

Attributes	Externally Maintained	Validation Base Type	Dimension Dependencies	Mandatory	Multivalued
Deactivated Attribute	No	List Of Values with: • ID=true, value=Yes • ID=false, value=No It is not recommended to reuse this LOV.	None	No	No
Potential Duplicate Match Score	Yes	Number	None	No	No

Attributes	Externally Maintained	Validation Base Type	Dimension Dependencies	Mandatory	Multivalued
Note: Must be valid on Potential Duplicate Reference Type.					
Source Record ID Attribute	No	Text (40-character limit)	None	No	Yes
Source System ID Attribute	No	Text	None	No	No

Important: These attributes are owned by the component model. Making them valid for object types not used by the component model results in a configuration warning on all matching algorithms that use them. If the you use a business rule to change any of the protected attributes or references, it generates an error, and the system is unable to complete its task.

Reference Types	Externally Maintained	Dimension Dependencies	Allow Multiple References	Mandatory	Inheritance	Valid Source Types	Valid Target Types
Merged-Into Relation Reference Types	No	None	No	No	None	All Merge Golden Records	All Merge Golden Records
Potential Duplicate Reference Type	Yes	None	No	No	None	All Merge Golden Records	All Merge Golden Records
Source Relation Reference Type	No	None	Yes	No	None	All Merge Golden Records	Source System Object Type
Unmerged-	No	None	No	No	None	Merge	Same

Reference Types	Externally Maintained	Dimension Dependencies	Allow Multiple References	Mandatory	Inheritance	Valid Source Types	Valid Target Types
From Relation Reference Types						Golden Records object types that should support unmerge	as the source types
Unmerge Reference Type	No	None	No	No	None	Merge Golden Records object types that should support unmerge	Same as the source types

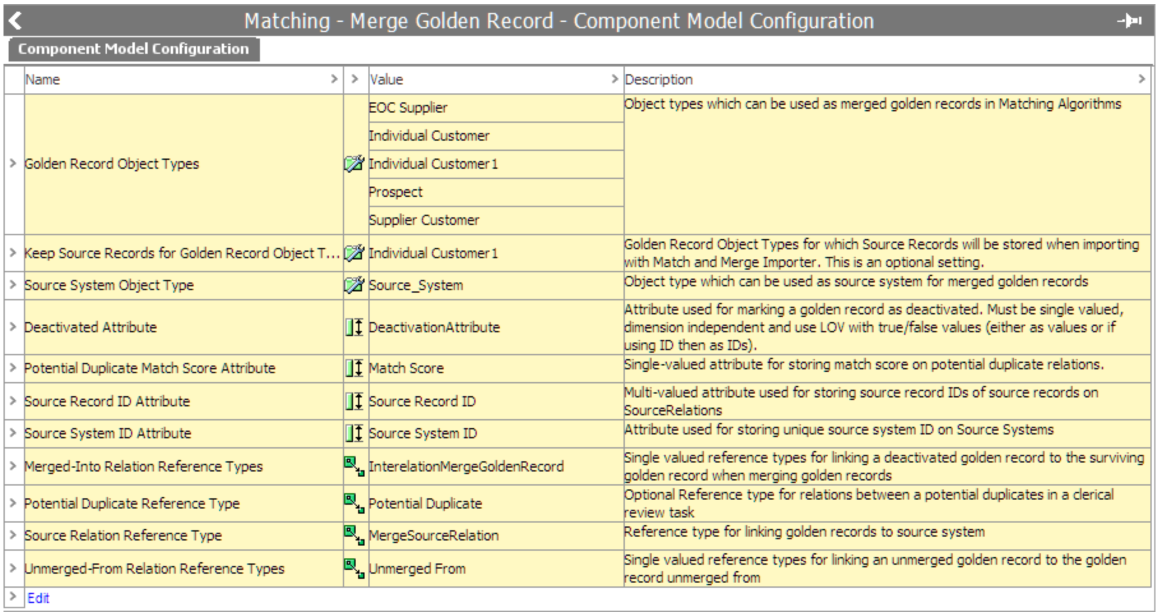
Note: These references are owned by the component model. Making them valid for object types not used by the component model results in a configuration warning on all matching algorithms that use them. Furthermore, changing or deleting a reference or a node with a reference will produce an error. To change or delete the reference type, remove the node from the workflow or remove the reference type from the component model.

Important: If the references or attributes used in the component model have data errors (e.g., the potential duplicate already had references across the system when it was designated as a potential duplicate reference type) the only way to fix it is to temporarily remove it from the component model, fix the problems manually, and then reassign it to the component model.

Configuration

To configure the component model:

1. In System Setup, expand 'Component Models' node and select the **Matching - Merge Golden Record** node.
2. On the 'Component Model Configuration' tab, click the **Edit** link (or the **Edit (pending changes)** link) to display the 'Edit Component Model Configuration' dialog.



Matching - Merge Golden Record - Component Model Configuration

Component Model Configuration

Name	Value	Description
Golden Record Object Types	EOC Supplier Individual Customer Individual Customer1 Prospect Supplier Customer	Object types which can be used as merged golden records in Matching Algorithms
Keep Source Records for Golden Record Object T...	Individual Customer1	Golden Record Object Types for which Source Records will be stored when importing with Match and Merge Importer. This is an optional setting.
Source System Object Type	Source_System	Object type which can be used as source system for merged golden records
Deactivated Attribute	DeactivationAttribute	Attribute used for marking a golden record as deactivated. Must be single valued, dimension independent and use LOV with true/false values (either as values or if using ID then as IDs).
Potential Duplicate Match Score Attribute	Match Score	Single-valued attribute for storing match score on potential duplicate relations.
Source Record ID Attribute	Source Record ID	Multi-valued attribute used for storing source record IDs of source records on SourceRelations
Source System ID Attribute	Source System ID	Attribute used for storing unique source system ID on Source Systems
Merged-Into Relation Reference Types	InterelationMergeGoldenRecord	Single valued reference types for linking a deactivated golden record to the surviving golden record when merging golden records
Potential Duplicate Reference Type	Potential Duplicate	Optional Reference type for relations between a potential duplicates in a clerical review task
Source Relation Reference Type	MergeSourceRelation	Reference type for linking golden records to source system
Unmerged-From Relation Reference Types	Unmerged From	Single valued reference types for linking an unmerged golden record to the golden record unmerged from
Edit		

- Double click the plus button (+) on an aspect to display the 'Select ... for aspect' dialog and select an object type, attribute, or reference type. The button remains active for aspects that allow multiple selections.
- Double click the delete button (☒) to remove a selection.

A green check (✓) means the aspect has no errors; a red X (✗) means additional setup is required. Hover over the X for additional information.

3. For each of the component model aspects, choose to add the object type(s), attribute(s), or reference(s) configured per the **Prerequisites** section and click the **Select** button.
   - **Golden Record Object Types** – Select the object types that can be used as golden records for Merge Golden Record configurations.
   - **Keep Source Records for Golden Record Object Types** – Select golden record object types for which the source record data should be stored.

     > **Note:** The Keep Source Records for Golden Record Object Types parameter is used in conjunction with the Source Record ID Attribute and the Source System ID Attribute to store source record information. For more information, refer to the **Match and Merge Traceability** topic.

   - **Source System Object Type** – Select the golden record object type used as a source system. This source system is referenced by golden records to signify where the record originated.
   - **Deactivated Attribute** – Select the attribute to mark a golden record as deactivated. Deactivated Attribute values are maintained via the match and merge match action. It is not advisable to maintain these by other means.

- **Potential Duplicate Match Score Attribute** – Select the attribute to store the golden record's match score on the potential duplicate reference.

- **Source Record ID Attribute** – Select the attribute used to store the IDs of source records on golden record objects. Source Record ID Attribute values are copied from source records via the match and merge match action. It is not advisable to edit Source Record ID Attribute values by other means. Source Record ID Attribute values must be unique, and an error is returned in the execution report when a duplicate ID is attempted.

- **Source System ID Attribute** – Select the attribute used for storing unique source system IDs on their respective source system objects.

- **Merged-Into Relation Reference Types** – Select the reference types that link a deactivated golden record to a surviving golden record during a merge.

- **Potential Duplicate Reference Type** – Optional, In-Memory is required. Select the reference type used by the matching algorithm from all Golden Records in a clerical review to the workflow node. For more information, refer to the **Configuring Matching Algorithms** topic. Enabling the potential duplicate reference makes filters available in the clerical review task list.

  Important: When adding a Potential Duplicate reference to a system with existing clerical review tasks, you must republish events for all Merge Golden Record nodes. Until this republishing process is completed by the event processor, the Clerical Review Task List shows incomplete data.

  Note: Potential duplicate references are only optimized and supported for systems running In-Memory. For more information, refer to the **In-Memory Database Component for STEP** topic in the online help **Resource Material** documentation.

- **Source Relation Reference Type** – Select the reference type that links golden records to source system objects. Source relation references are maintained via the match and merge match action. It is not advisable to maintain these by other means.

- **Unmerged-From Relation Reference Types** – Optional. If configuring an Unmerge workflow, select the entity-to-entity reference type that is used for the workflow.

4. Save or cancel your work:
   - Click the **Save** button to save a configuration once it has no errors.
   - When enabled, click the **Save pending** button to save your work while errors exist.
   - When enabled, click the **Restore live settings** button to undo the changes made to a previously error-free, saved configuration.
   - Click the **Cancel** button to undo all changes made in this dialog.

Match Criteria

Within a Matching Algorithm, the match criteria are responsible for matching records against each other to find those that match. When users are only interested in exact matches, the match criteria are reasonably straightforward.

For example, if the SSN (Social Security Number) for two customer objects or the EAN (European Article Number) for two product objects are identical, the records are likely duplicates and the matching criteria should return 100 percent. If the SSN or EAN does not match, the match criteria should probably return 0 percent.

In many cases you cannot work with exact matches; instead, you will deal with approximate matches or a combination of exact and approximate matches. For example, for a customer you do not have a SSN available so you will identify duplicates based on names, mailing addresses, phone numbers, and street addresses. For a product, you will identify duplicates based on the manufacturer and manufacturer part number.

This data can have variations, even in objects that represent the same real-world item. Names and addresses can be spelled differently, middle names could be omitted, abbreviations can be used in names and addresses, the customers could be registered with different phone numbers or mailing addresses, and other options that introduce ambiguity to the records.

This complexity can be handled via a decision table in the match criteria logic, which further divides the functionality into normalizers, matchers, and rules.

The Match Criteria uses a decision table to define how to compare two objects and evaluate to what degree they are similar by producing a match score. For more information, refer to the **Match Scores** topic.

Creating Match Criteria

Match Criteria is comprised of Data Elements, Matchers, Rules, Match Code Generators, and Match Code Filters for a matching algorithm. All are added and configured on the Decision Table dialog.

To create match criteria:

1. Edit the match criteria based on the type of matching algorithm:

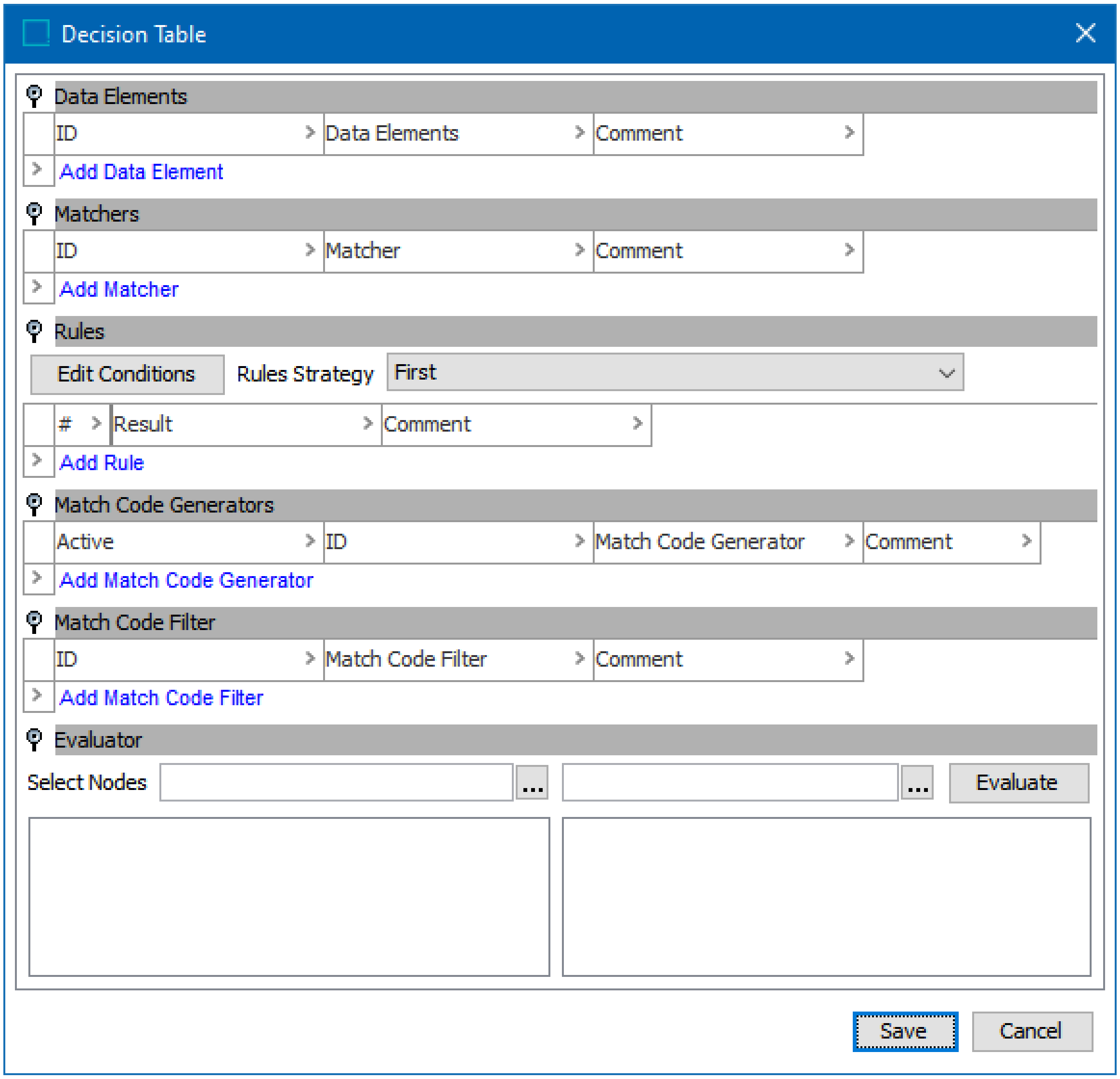
   With embedded match codes: select the Match Criteria tab and click the **Edit Match Criteria** link to display the Decision Table dialog.

   Without embedded match codes (This is a legacy matching algorithm type and has no Match Criteria tab.): on the Matching Algorithm tab open the Match Action flipper, click the **Add Criterion** link, add an ID, select **Decision Table** from the dropdown and click the **Add** button.

   - To edit an existing Decision Table criterion row, click the ellipsis button (...) for the criterion to display the Decision Table dialog.
   - To create a new match criteria click the **Add Criterion** link.

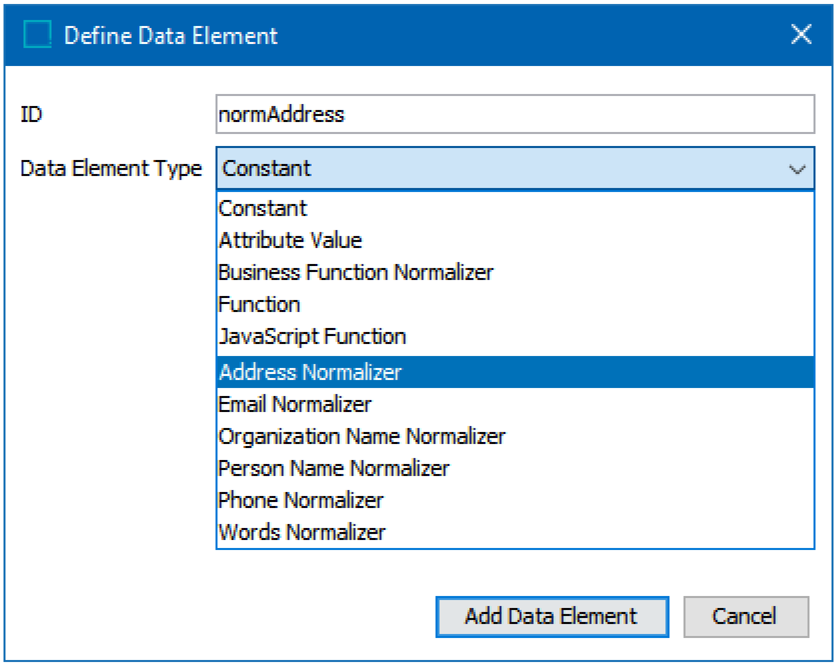
2. In the Decision Table dialog, for each of the following flippers, add one or more rows, and then configure the new row(s) as required:

> **Important:** IDs must be unique across the data elements flipper, the matchers flipper, and the match code generators flipper on the Decision Table dialog.



Rules Strategy: First

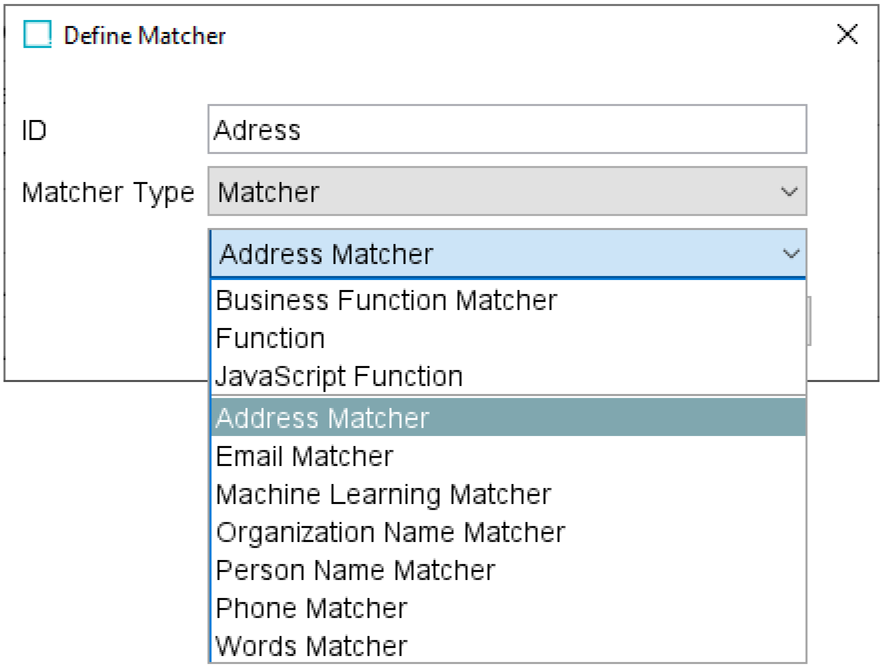
- **Data Elements** declare the input for the matchers and match code generators and allow data to be normalized to a format that is easy to compare. Refer to the **Match Criteria Data Elements** topic.


Define Data Element

ID: normAddress

Data Element Type: Constant

- **Matchers** do the actual comparisons of values from the specified data element. A matcher compares one logical aspect of the objects, assigning an equality percentage to that aspect based on the related values. Refer to the **Match Criteria Matchers** topic.

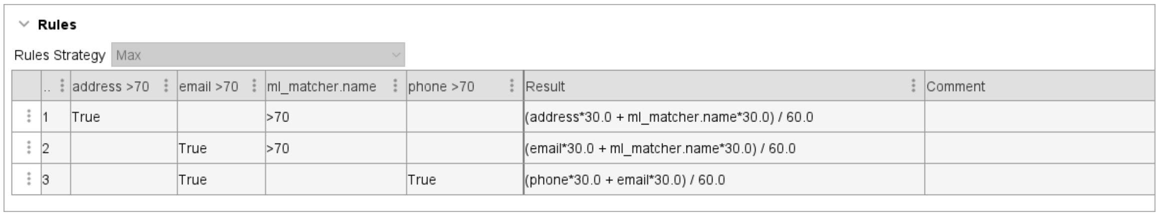

Define Matcher

ID: Adress

Matcher Type: Matcher

Address Matcher

- **Rules** combine the results of matchers into a final match score, which is a percentage that signifies if two objects are a match or are not a match. A new row is added to the flipper and can be configured as needed. Refer to the **Match Criteria Rules** topic.

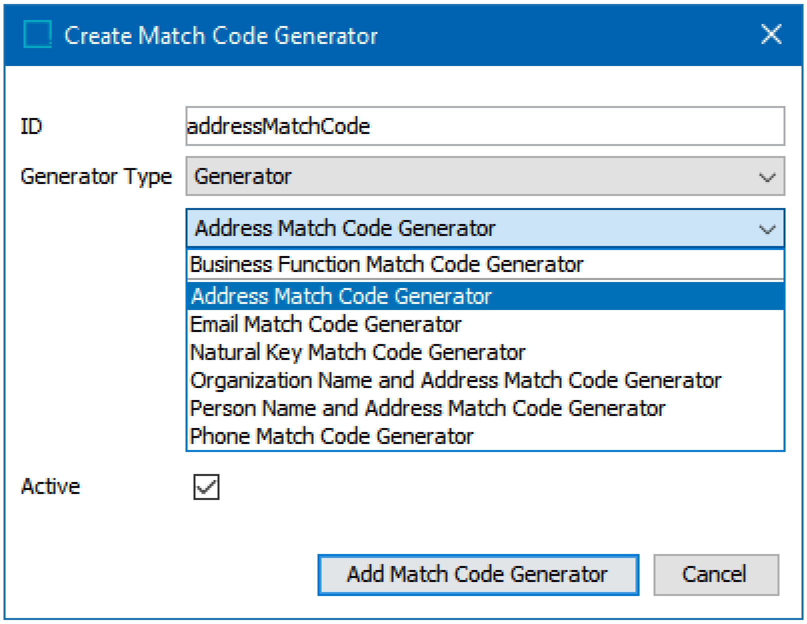


Rules

Rules Strategy Max

	..	address >70	email >70	ml_matcher.name	phone >70	Result	Comment
⋮	1	True		>70		(address*30.0 + ml_matcher.name*30.0) / 60.0	
⋮	2		True	>70		(email*30.0 + ml_matcher.name*30.0) / 60.0	
⋮	3		True		True	(phone*30.0 + email*30.0) / 60.0	

- **Match Code Generators** identify the records that should be compared. Only records with at least one equal match code are passed through the match criteria for evaluation of a match score. This allows efficient matching on a dataset of millions of objects because it prevents comparing every object with every other object. Refer to the **Match Criteria Match Code Generators** topic.



Create Match Code Generator

ID: addressMatchCode

Generator Type: Generator

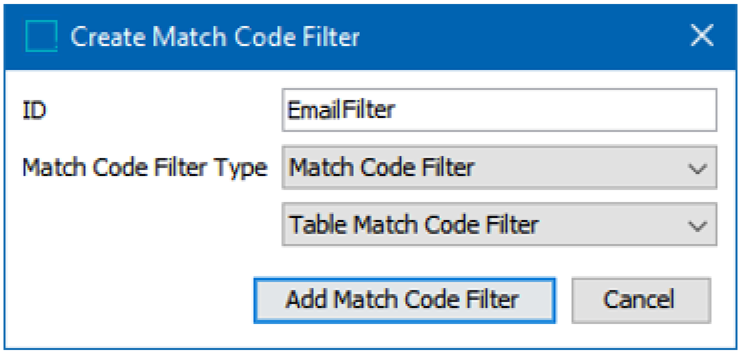
Address Match Code Generator

Business Function Match Code Generator
Address Match Code Generator
Email Match Code Generator
Natural Key Match Code Generator
Organization Name and Address Match Code Generator
Person Name and Address Match Code Generator
Phone Match Code Generator

Active ☑

Add Match Code Generator
Cancel

- **Match Code Filter** allows users to remove specific match code values based on data exceptions defined in a Transformation Lookup Table. Refer to the **Match Criteria Match Code Filter** topic.



Create Match Code Filter

ID: EmailFilter

Match Code Filter Type: Match Code Filter

Table Match Code Filter

3. In the Decision Table dialog, open the Evaluator flipper and test the configuration.

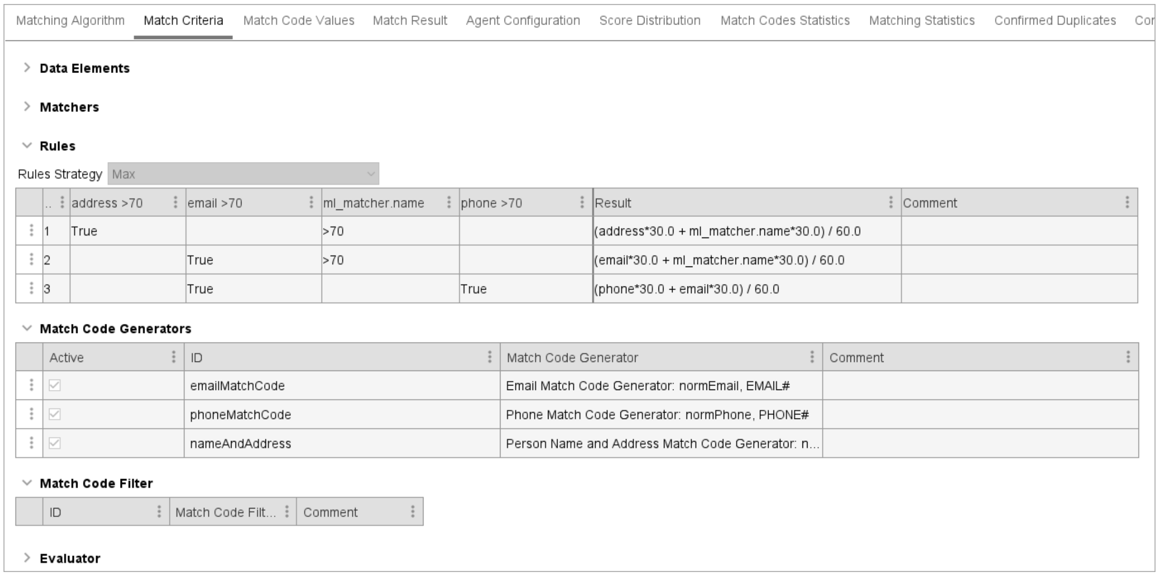
   - For the Select Nodes parameters, click the ellipsis button (...) for each field and select two objects for comparison.

- Click the **Evaluate** button.

An empty result field indicates the value is not available in the selected node. Adjust as indicated by the Evaluator results and repeat the evaluation.

4. Click the **Save** button to keep the Match Criteria changes and return to the Matching Algorithm object.

An example set of match criteria elements is shown below.



Rules Strategy Max

..	address >70	email >70	ml_matcher.name	phone >70	Result	Comment
1	True		>70		(address*30.0 + ml_matcher.name*30.0) / 60.0	
2		True	>70		(email*30.0 + ml_matcher.name*30.0) / 60.0	
3		True		True	(phone*30.0 + email*30.0) / 60.0	

Match Code Generators

Active	ID	Match Code Generator	Comment
☑	emailMatchCode	Email Match Code Generator: normEmail, EMAIL#	
☑	phoneMatchCode	Phone Match Code Generator: normPhone, PHONE#	
☑	nameAndAddress	Person Name and Address Match Code Generator: n...	

Match Code Filter

ID	Match Code Filt...	Comment

Potential Duplicate Match Scores Examples

In a Match and Merge match action solution, the match score can be stored on a potential duplicate reference type.

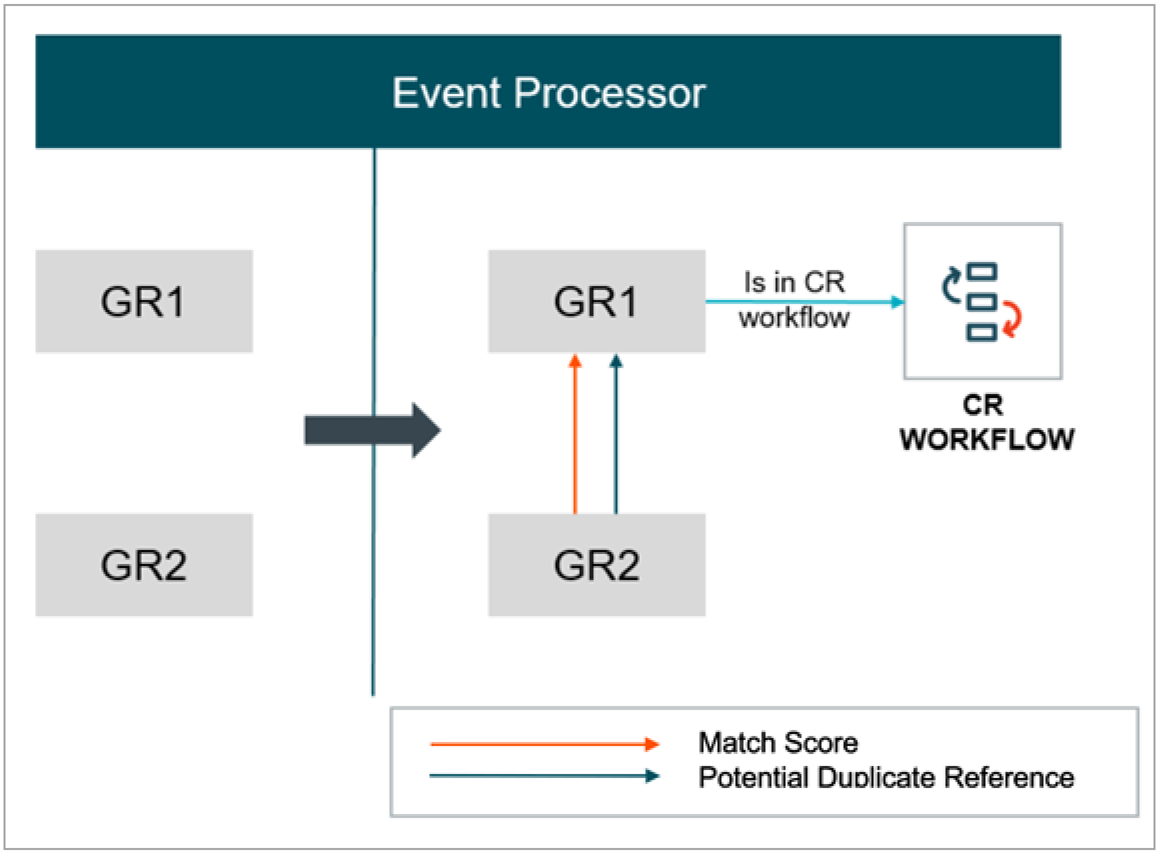
STEP matching algorithms use match criteria which produce match scores to quantitatively measure the likeness of two golden records. This likeness may result in Clerical Review tasks, which can materialize as potential duplicate references. The match scores between a potential duplicate and a workflow node can be stored on that reference.

Important: Clerical Review Task Lists can function even without a potential duplicate reference type clarified in the Matching - Merge Golden Record component model; however, it will be much less stable and could result in errors. For more information, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.

While it is not a comprehensive list, the following sections include examples of how golden records might link as potential duplicates produce match scores when a new golden record is matched with an existing one.

New Record Match with Neither in a Workflow

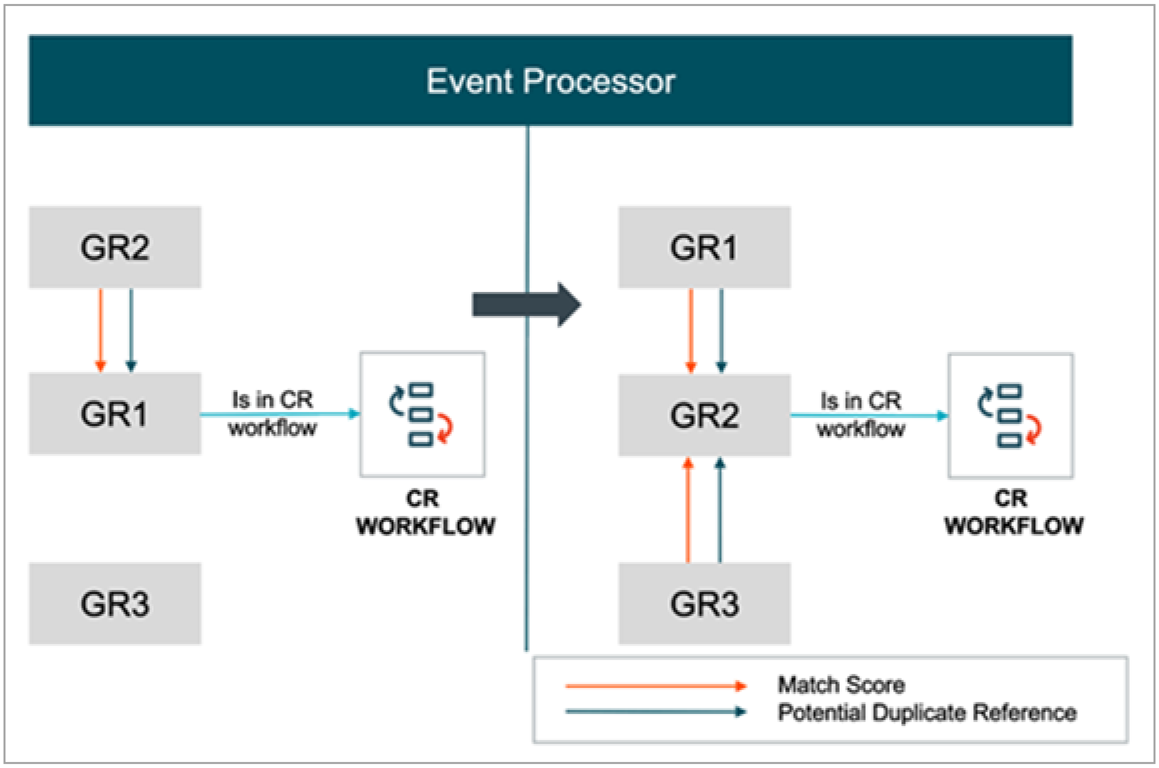
The following scenario represents when two golden records are compared as potential duplicates when neither is in a Clerical Review workflow.



1. One of the records is edited (e.g. a user adds a phone number) or created.
2. The event processor acknowledges the change and initiates the matching algorithm; it finds a shared match code between GR1 and GR2.
3. GR1 and GR2 are compared and receive a match score within the clerical review threshold.
4. The event processor creates a new task and links the two records with a potential duplicate reference. For easy searching, a potential duplicate reference is also added from GR2 to GR1 when the task is created.
5. GR1 is now in the Clerical Review workflow, GR2 now references GR1 with a potential duplicate reference type, and the match score is saved as metadata on the potential duplicate reference.

New Record Match to an Existing Potential Duplicate

The following scenario represents when a third golden record is introduced and compared to two already matched golden records.

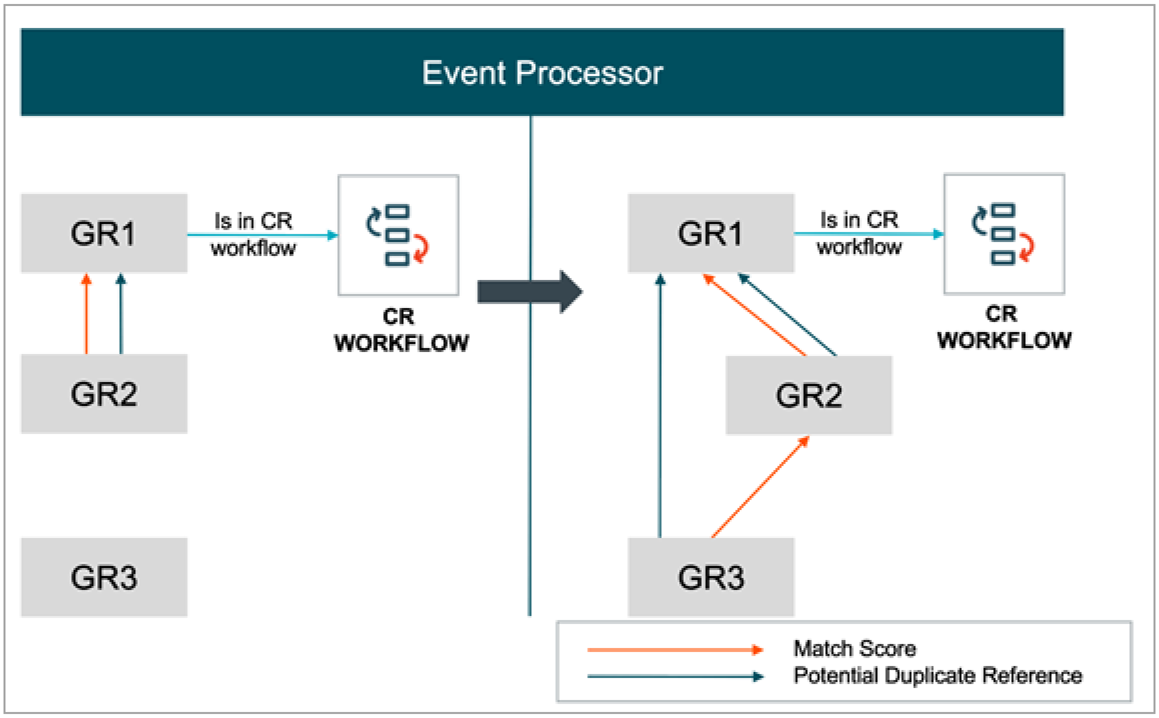


1. A user updates GR3, which now shares a match code with GR1.
2. The event processor acknowledges the update to GR3 and initiates the matching algorithm; it finds a shared match code between GR1 and GR3.
3. GR1 and GR3 are compared and receive a match score within the Clerical Review threshold.
4. The event processor identifies the existing task and links the GR3 to GR1 with a potential duplicate reference.

GR1 now shares a potential duplicate reference with both GR2 and GR3. The match scores for both are saved as metadata on their respective potential duplicate references.

New Indirect Match to an Existing Potential Duplicate

The following scenario represents when a third golden record is introduced and matches with another potential duplicate (GR2), but not with the workflow node (GR1).



1. A user updates GR3, which now shares a match code with GR1.
2. The event processor acknowledges the update to GR3 and initiates the matching algorithm; it finds a shared match code between GR2 and GR3.
3. GR2 and GR3 are compared and receive a match score within the Clerical Review threshold range.
4. The event processor identifies the existing task and links GR3 to GR1 with a potential duplicate reference. There is no potential duplicate reference between GR2 and GR3, and no match score between GR1 and GR3.
5. GR1 now shares a potential duplicate reference with both GR2 and GR3. As there is no match code shared between GR3 and GR1, they receive no match score. GR3 is only indirectly included in the review. This is indicated in the Clerical Review Task List by an 'N/A' displaying, instead of a match score.

Match and Merge Clerical Review - Merge

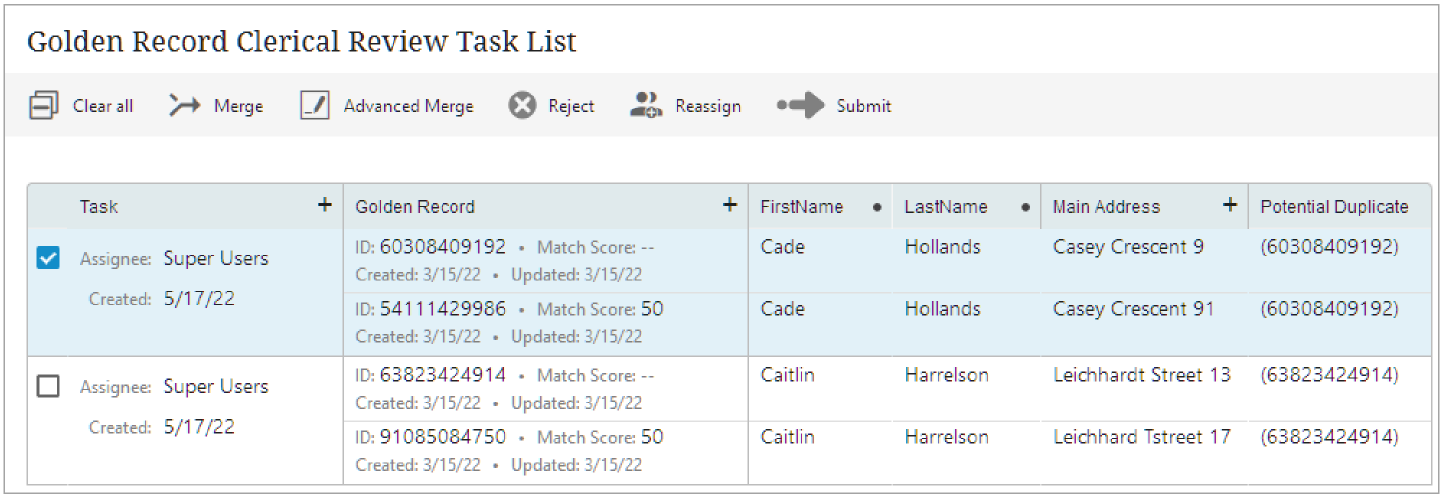
The match and merge solution is supplemented by a Web UI clerical review task list and an advanced merge screen that assist in clerical reviews for potential duplicates.

> **Note:** This screen is only to be used with the match and merge solution. The primary users of this screen are data stewards who can decide if entities are duplicates or non-duplicates. Stibo Systems does **not** recommend you use the same clerical review screen for more than one match algorithm. Instead, assign each matching algorithm clerical review to a specialized user group based on the group's function. For more information, refer to the **Golden Record Clerical Review Task List** topic.

The Golden Record Clerical Review Task List screen displays all potential duplicates found in a specific golden record clerical review workflow or workflow state. From this screen, golden records are grouped into tasks and can be:

- Rejected as duplicates via the 'Reject' action button.
- Acknowledged as duplicates and merged via the 'Merge' or 'Advanced Merge' action buttons
- Reassigned to other users via the 'Reassign' action button.
- Submitted to another state in the workflow via the 'Submit' action button.

To be included on this screen, a golden record must have been flagged as a potential duplicate by the relevant matching algorithm during import. This means that it fell within the clerical review threshold of the matching algorithm and was initiated into a clerical review workflow where it can be evaluated against other records. Potential duplicates that are matched together are grouped into distinct tasks in the workflow, as pictured below.



	Task	Golden Record	FirstName	LastName	Main Address	Potential Duplicate
☑	Assignee: Super Users Created: 5/17/22	ID: 60308409192 • Match Score: -- Created: 3/15/22 • Updated: 3/15/22	Cade	Hollands	Casey Crescent 9	(60308409192)
		ID: 54111429986 • Match Score: 50 Created: 3/15/22 • Updated: 3/15/22	Cade	Hollands	Casey Crescent 91	(60308409192)
☐	Assignee: Super Users Created: 5/17/22	ID: 63823424914 • Match Score: -- Created: 3/15/22 • Updated: 3/15/22	Caitlin	Harrelson	Leichhardt Street 13	(63823424914)
		ID: 91085084750 • Match Score: 50 Created: 3/15/22 • Updated: 3/15/22	Caitlin	Harrelson	Leichhard Tstreet 17	(63823424914)

For information on setting up and using the golden record clerical review screen as well as the advanced merge feature, refer to the **Golden Record Clerical Review Task List** topic of the **Web User Interfaces** documentation.

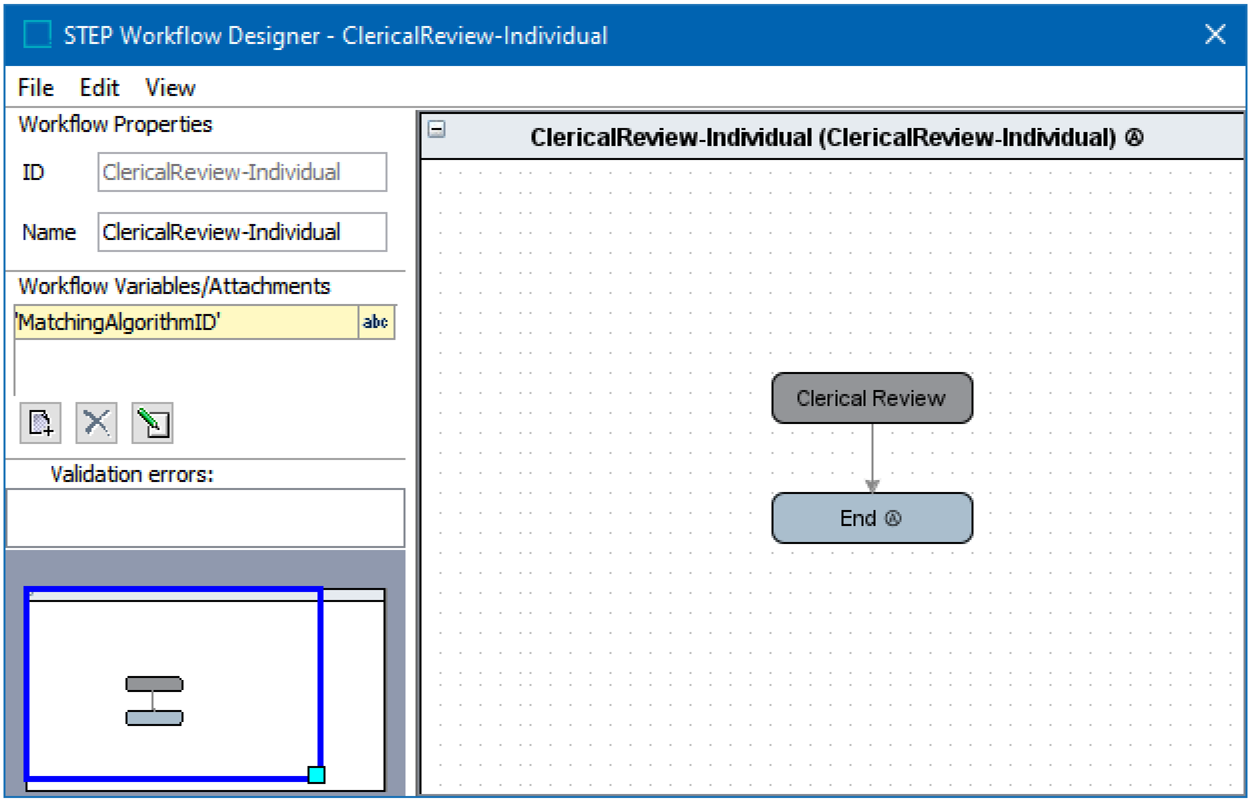
For information about Submit Action, refer to the **Submit Action** topic.

Creating a Merge Golden Record Clerical Review Workflow

A clerical review workflow in Web UI allows data steward to manually determine the match status of objects when the match score falls between the Auto Threshold and the Clerical Review Threshold. Optionally, a workflow status flag and a business condition allows a high priority setting when required. For details, refer to the **Match and Merge Clerical Review - Merge** topic.

Configuration

The workflow and thresholds are part of configuring a Match and Merge solution and are selected when configuring the Merge Golden Record match action as defined in the **Configuring Merge Golden Record Match Action** topic.



STEP Workflow Designer - ClericalReview-Individual

Workflow Properties

ID: ClericalReview-Individual

Name: ClericalReview-Individual

Workflow Variables/Attachments

'MatchingAlgorithmID'

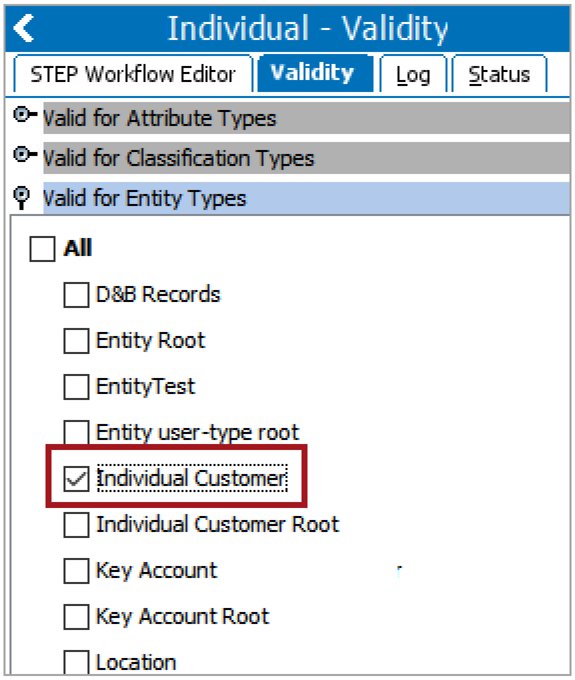
Validation errors:

The following elements are available for the clerical review workflow:

- **Workflow** - (required) create a workflow as simple or elaborate as needed. For more information, refer to the **Creating a Workflow** topic in the **Workflows** documentation.

Use a case sensitive Event ID to allow access to the Submit button on the Golden Record Clerical Review Task List in Web UI, as defined in the **Golden Record Clerical Review Task List** topic.

- **Clerical Review High Priority Status Flag -** (optional) if desired, create a workflow status flag used to designate high priority tasks in the clerical review workflow. No other status flags should be set on the clerical review workflow. For details on setup, refer to the **Status Flags** topic in the **Workflows** documentation.
- **Clerical Review High Priority Business Condition -** (required when the status flag is used), create a business condition to verify if a task is high priority. The business condition is evaluated on each potential duplicate object in the clerical review task in the context of the matcher and has access to the 'Current Object' bind. For details on setup, refer to the **Creating a Business Rule, Function, or Library** topic in the **Business Rules** documentation.
- On the Validity tab of the workflow, select the merge object type.



□ All
- □ D&B Records
- □ Entity Root
- □ EntityTest
- □ Entity user-type root
- ☑ Individual Customer
- □ Individual Customer Root
- □ Key Account
- □ Key Account Root
- □ Location

Considerations

The following rules apply when using a clerical review workflow to configure the Merge Golden Record match action:

- If a status flag is configured, but a business condition is not configured, the status flag behaves as if a business condition evaluated to true.
- If a business condition is configured and a status flag is not configured, the business condition is ignored.
- Although the business condition runs as a part of matching and it involves a clerical review, no matching or workflow binds are available.

- No additional status flags should be configured on the clerical review workflow since the matching algorithm in the Merge Golden Record match action determines which status flags are set (or not set).

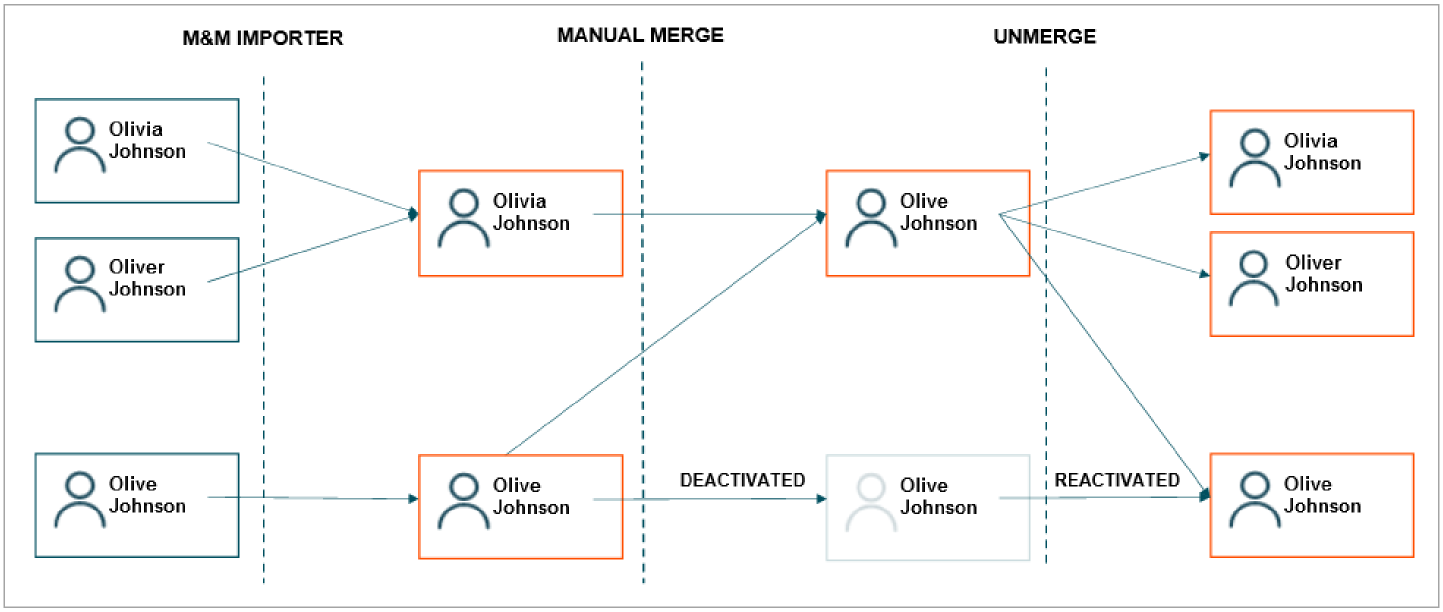
Match and Merge Clerical Review - Unmerge

Unmerge allows users to remove connections between records that have been wrongly merged either as the result of a manual action or by auto merge. Unmerge requires its own configuration and is only available in Web UI. Unmerge uses the survivorship rules for the 'Merge Golden Records' object types.

This functionality is available to incoming records with source record IDs as well as deactivated golden records. The unmerge operation restores relevant data back to the golden record and can be done as part of a workflow or as an ad hoc operation.

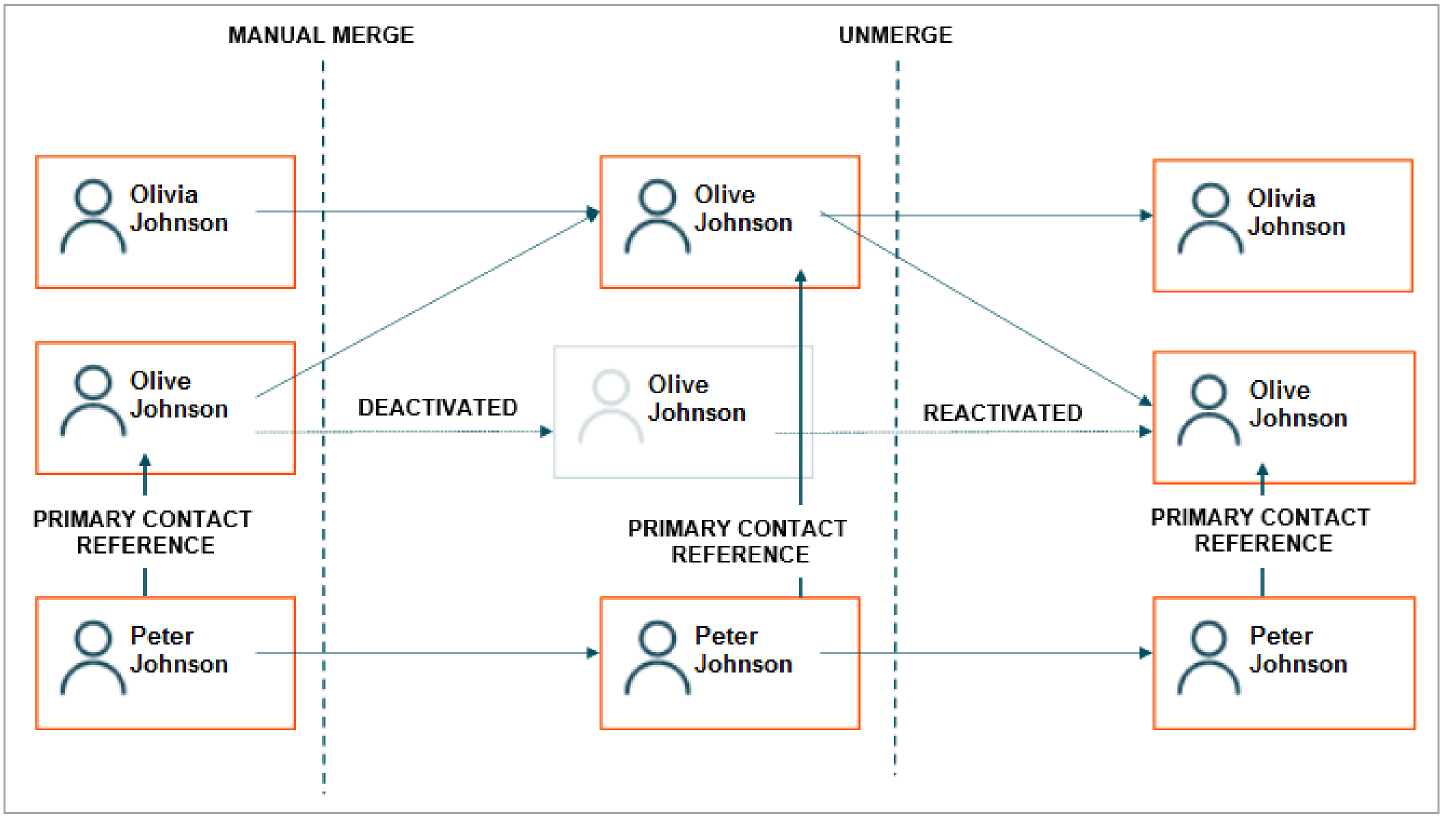
> **Important:** The unmerge logic to revert the changes is not supported on multi-valued data containers and references.

In the example below, the three customer records (Olivia, Oliver, and Olive) were merged and the unmerge operation must separate the record, effectively splitting a single golden record into individual golden records. In the Unmerge operation, the deactivated Olive Johnson record is reactivated and all updates concerning Oliver Johnson are moved to a new record.



When reactivating golden records or moving source records, the user identifies the source records and the manual updates that belong on each reactivated golden record or moving source record. Then the unmerge uses survivorship rules to calculate the possible values for the golden records. If it is not possible to automatically determine the correct version of the golden record the second step of unmerge allows the user to verify and correct data. The user can choose each of the final values on each golden record.

Below, Peter Johnson has an inbound reference that was created from a Match and Merge import using source record ID and stored as source records.



The unmerge operation first attempts to revert to the values that existed prior to being incorrectly merged. This action is applicable for both merged golden records that are now being reactivated as well as source records that were wrongly merged into the golden record. The reversion logic has two paths for removing values and reverting to the original source records.

- For merged golden records, the record is reactivated. The ‘Merged into’ traceability determines whether or not to revert back to a certain value when the values originally came from either manual entry or imports without source record IDs. For more information on traceability, refer to the **Match and Merge Traceability** topic.
- For source records that were incorrectly automatically merged into the golden record during an import, since all existing revisions have the source information, moving a source record to another golden record reverts the values coming from that particular source.

The unmerge operation next applies the configured survivorship rules to the remaining associated source records, if any. This ensures that attributes with no valid value for reverting get the correct original value from the sources. For more information, refer to the **Survivorship in Match and Merge** topic.

Considerations

- On the Matching - Merge Golden Record component model, the 'Keep Source Records for Golden Record Object Types' aspect must be configured to revert to the original records without the potential for data loss. For imports with source record IDs, enabling and configuring the storage of source record data improves the unmerge result. The data of imports done before this configuration is not stored. For more information, refer to **Storing Source Records for Golden Records** topic.
- When unmerging, the system restores historical values and uses the current time as the STEP update timestamp on the golden record. This means that value data appears to be more recent than it actually is, which can impact ‘most recent’ survivorship functionality since the rules can choose an unexpected surviving value. To avoid this, it is recommended to always use ‘Last Edit’ attributes when configuring the survivorship rules for import. If 'Last Edit' attributes are used, unmerge also reverts these last edit dates, and the latter survivorship rules correctly determine the surviving values.
- If the matching algorithm has ‘Auto Approve’ enabled on the match action settings and the object type is workspace revisable, the golden records are auto-approved and any business conditions and/or business actions with ‘on approve’ enabled are evaluated.
- Unmerge attempts to assign inbound references back to the correct golden record. When completing the unmerge operation, inbound reference types that cannot be automatically reassigned are left unchanged and a count (grouped by inbound reference type) is displayed in a confirmation dialog.

For information on ad hoc unmerging, refer to the **Unmerging Golden Records** topic.

For information on an unmerge workflow, refer to the **Creating an Unmerge Golden Record Clerical Review Workflow** topic.

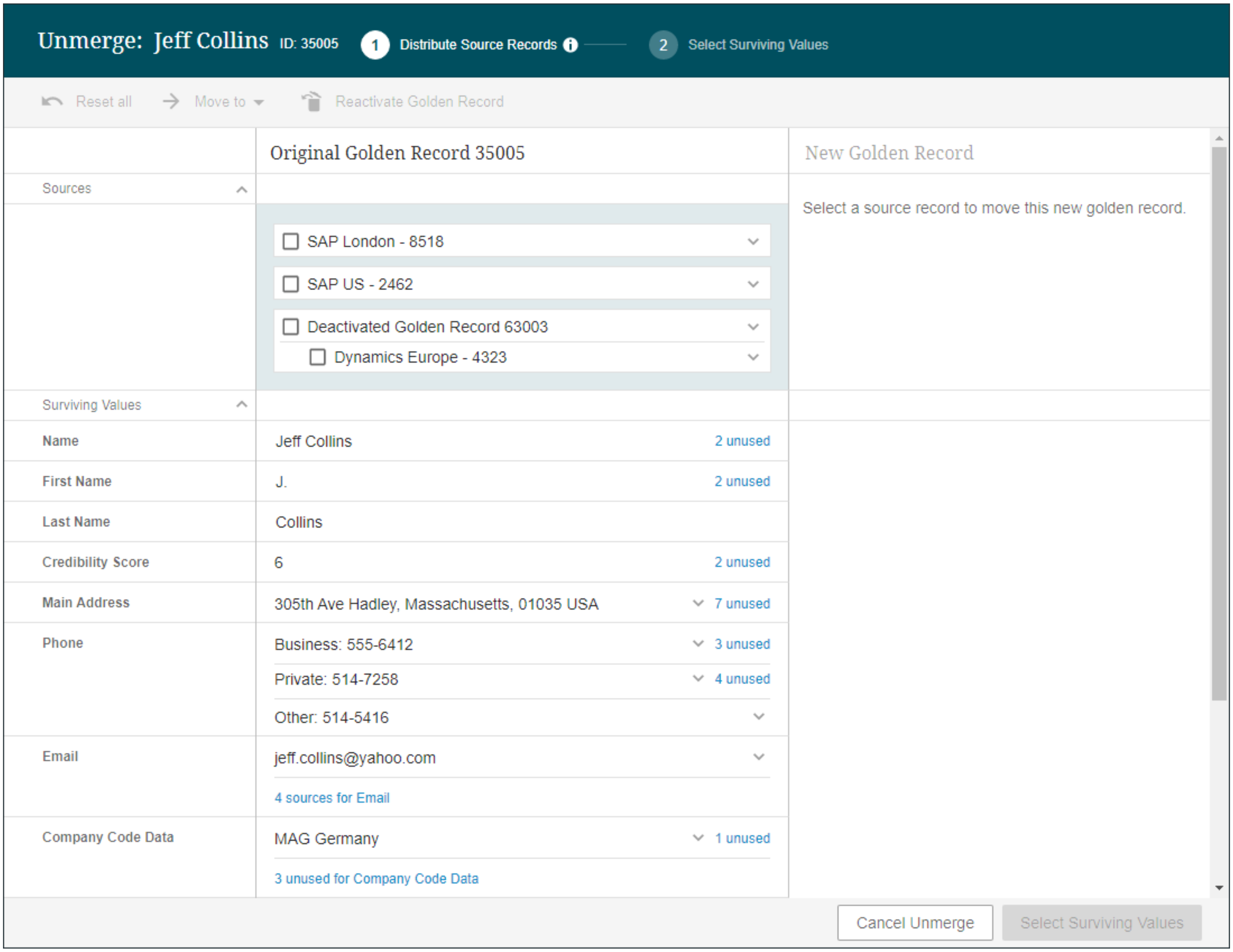
Unmerging Golden Records

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

Ad hoc unmerging is intended for users who are knowledgeable about the data and want to start the unmerge wizard. This topic covers ad hoc unmerging, which is performed outside of a workflow. Unmerging via a workflow is defined in the **Creating an Unmerge Golden Record Clerical Review Workflow** topic.

For the complete unmerge process, refer to the **Match and Merge Clerical Review - Unmerge** topic.

The Unmerge wizard in Web UI (shown below) provides a collaborative process for all unmerge operations.



Original Golden Record 35005

New Golden Record

Select a source record to move this new golden record.

☐ SAP London - 8518

☐ SAP US - 2462

☐ Deactivated Golden Record 63003

☐ Dynamics Europe - 4323

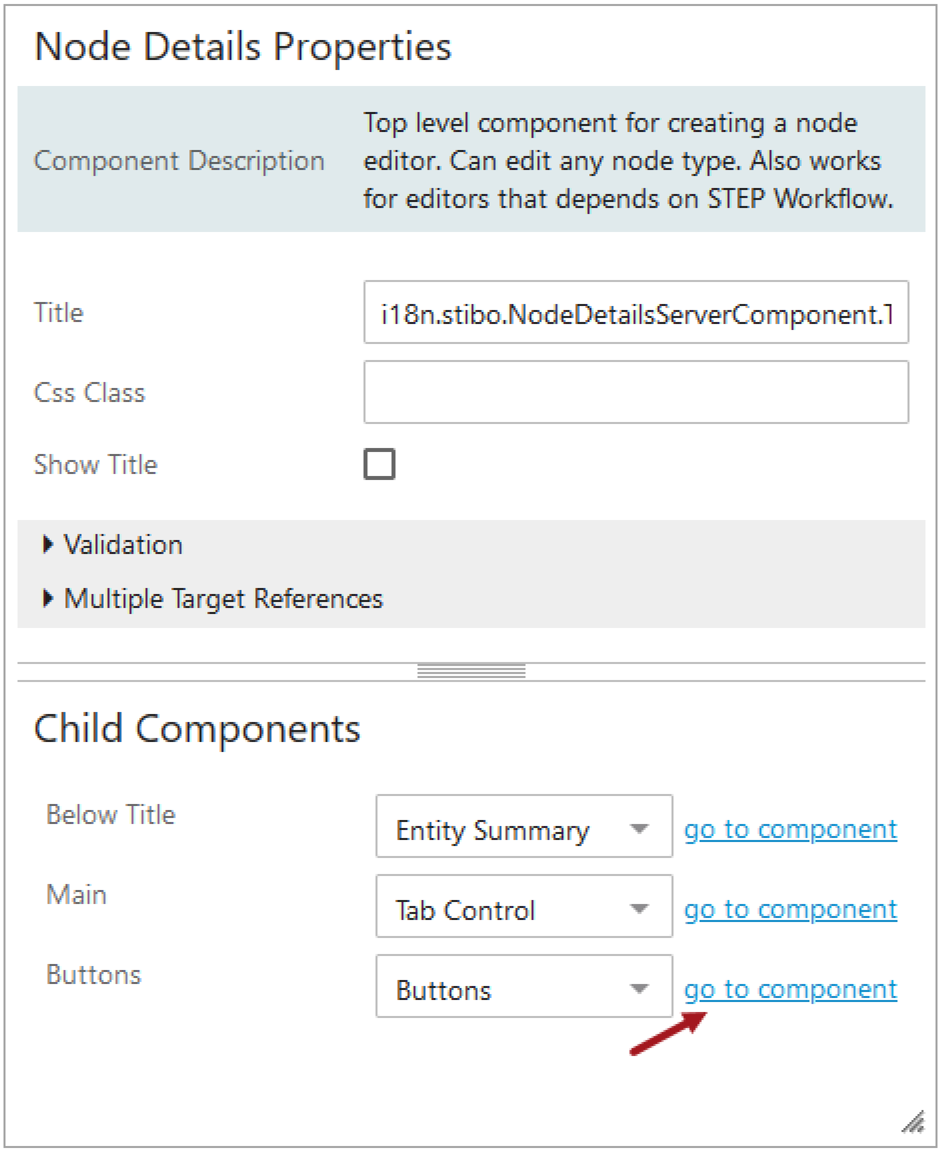
Name	Jeff Collins	2 unused
First Name	J.	2 unused
Last Name	Collins	
Credibility Score	6	2 unused
Main Address	305th Ave Hadley, Massachusetts, 01035 USA	7 unused
Phone	Business: 555-6412 Private: 514-7258 Other: 514-5416	3 unused 4 unused
Email	jeff.collins@yahoo.com 4 sources for Email	
Company Code Data	MAG Germany 3 unused for Company Code Data	1 unused

Configuration

Use these steps to configure an unmerge button which opens the unmerge wizard for ad hoc use.

1. In Web UI, open a node details screen used for the entities to be unmerged. Refer to the **Node Details Screen** topic of the **Web User Interfaces** documentation.

2. Open the Web UI Design Mode to display the Node Details Properties dialog. In the Child Components section, on the Buttons parameter, click the **go to component** link.



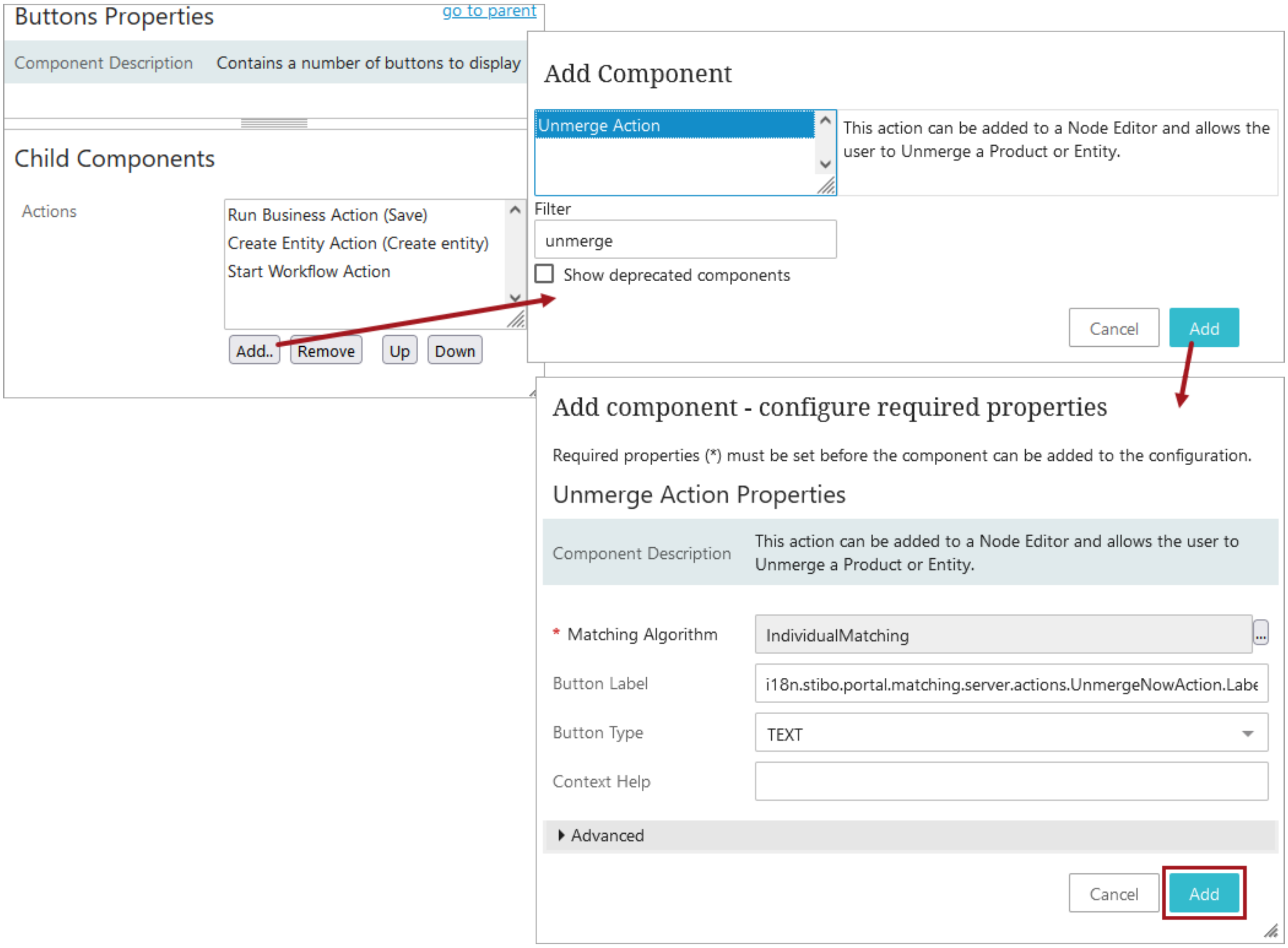
Node Details Properties

Component Description	Top level component for creating a node editor. Can edit any node type. Also works for editors that depends on STEP Workflow.
Title	i18n.stibo.NodeDetailsServerComponent.1
Css Class	
Show Title	□

Child Components

Below Title	Entity Summary	go to component
Main	Tab Control	go to component
Buttons	Buttons	go to component

3. For the Buttons Properties dialog, in the Child Components section, on the Actions parameter, click the **Add** button and select the **Unmerge Action** component. Click **Add** to close the dialog.



Buttons Properties

Component Description: Contains a number of buttons to display

Child Components

Actions:
Run Business Action (Save)
Create Entity Action (Create entity)
Start Workflow Action

Add Component

Unmerge Action — This action can be added to a Node Editor and allows the user to Unmerge a Product or Entity.

Filter: unmerge

□ Show deprecated components

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Unmerge Action Properties

Component Description	This action can be added to a Node Editor and allows the user to Unmerge a Product or Entity.
* Matching Algorithm	IndividualMatching
Button Label	i18n.stibo.portal.matching.server.actions.UnmergeNowAction.Labe
Button Type	TEXT
Context Help	

▸ Advanced

4. For the Unmerge Actions Properties dialog, provide the following information:

   - Matching Algorithm - select the algorithm for the Golden Record object type. Unmerge uses the Survivorship Rules defined in the algorithm.
   - Button Label - add the text to display on the button.
   - Button Type - select to use icon and text, icon only, or text only on the button.
   - Context Help - add text to display when hovering over the button.
   - Style Class - legacy parameter; leave unchanged.

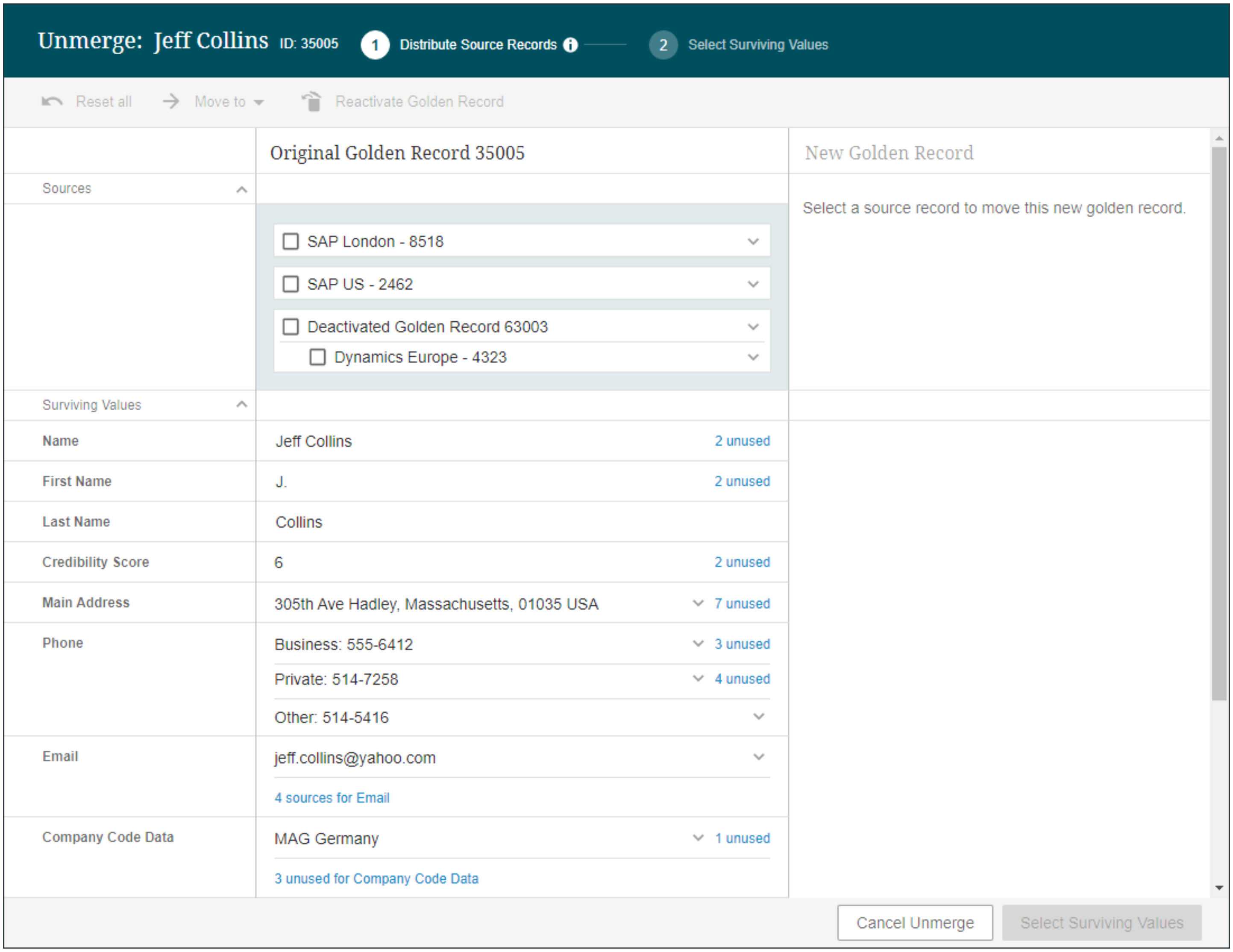
5. Click the **Add** button. Click **Save** and **Close** to exit the designer.

Creating an Unmerge Golden Record Clerical Review Workflow

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

Workflow-based unmerging is intended to add a level of control to the unmerge process by initiating a merged record into the initial state of the unmerge workflow where a knowledgeable user can decide to continue or exit the unmerge process. This workflow setup is optional since users can perform ad hoc unmerge operations, refer to the **Unmerging Golden Records** topic.

The Unmerge wizard in Web UI (shown below) provides a collaborative process for all unmerge operations. The workflow states allow users to support the overall process such as preparing data in the source systems before unmerging and validating data in the downstream systems after unmerging.



Select a source record to move this new golden record.

- ☐ SAP London - 8518
- ☐ SAP US - 2462
- ☐ Deactivated Golden Record 63003
  - ☐ Dynamics Europe - 4323

Name	Jeff Collins	2 unused
First Name	J.	2 unused
Last Name	Collins	
Credibility Score	6	2 unused
Main Address	305th Ave Hadley, Massachusetts, 01035 USA	7 unused
Phone	Business: 555-6412	3 unused
	Private: 514-7258	4 unused
	Other: 514-5416	
Email	jeff.collins@yahoo.com	
	4 sources for Email	
Company Code Data	MAG Germany	1 unused
	3 unused for Company Code Data	

For the complete unmerge process, refer to the **Match and Merge Clerical Review - Unmerge** topic.

For details on unmerge in Web UI, refer to the **Configuring and Using Match and Merge Unmerge in Web UI** topic.

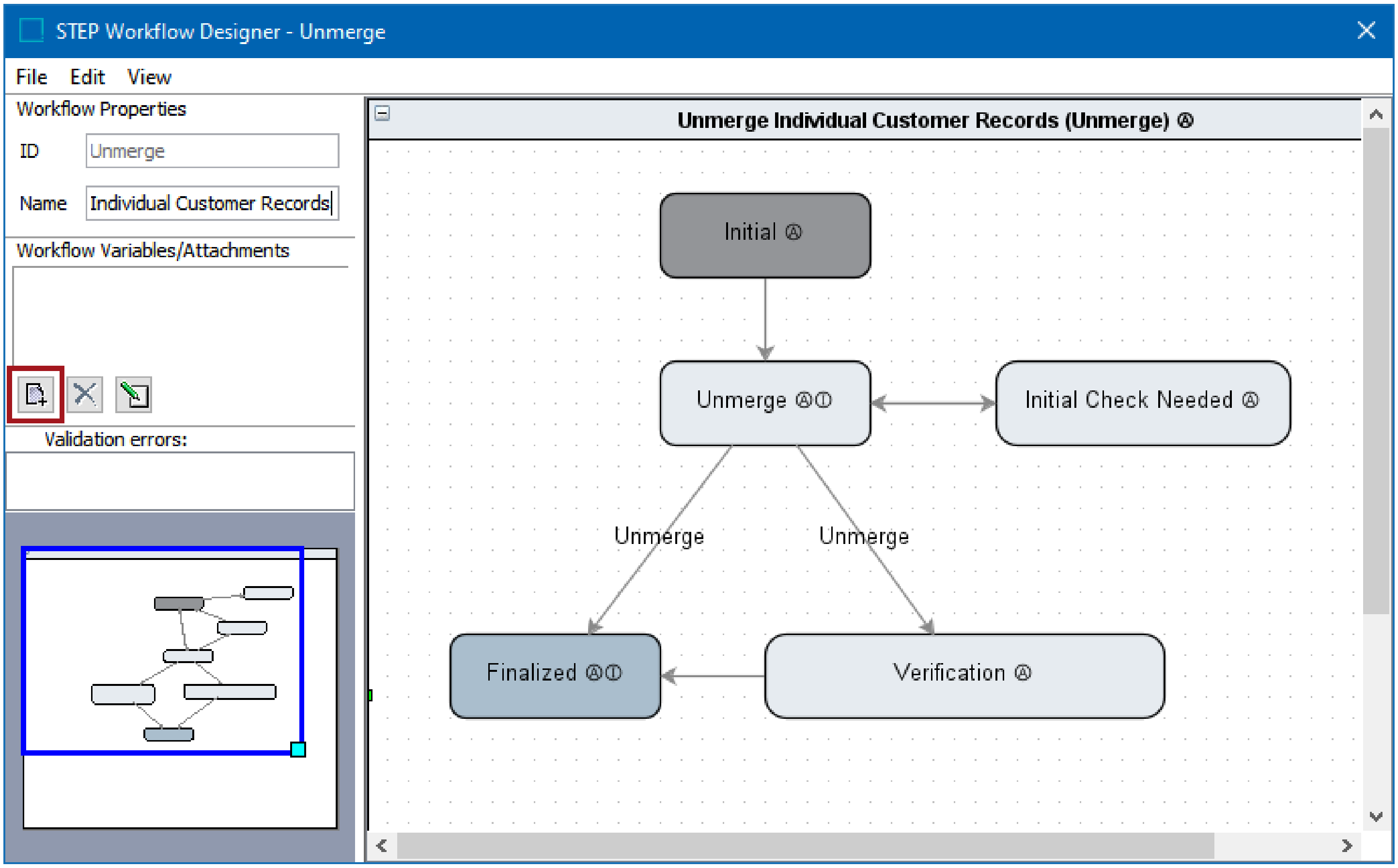
Configuration

Use these steps to configure an unmerge process via workflow in a Match and Merge solution.

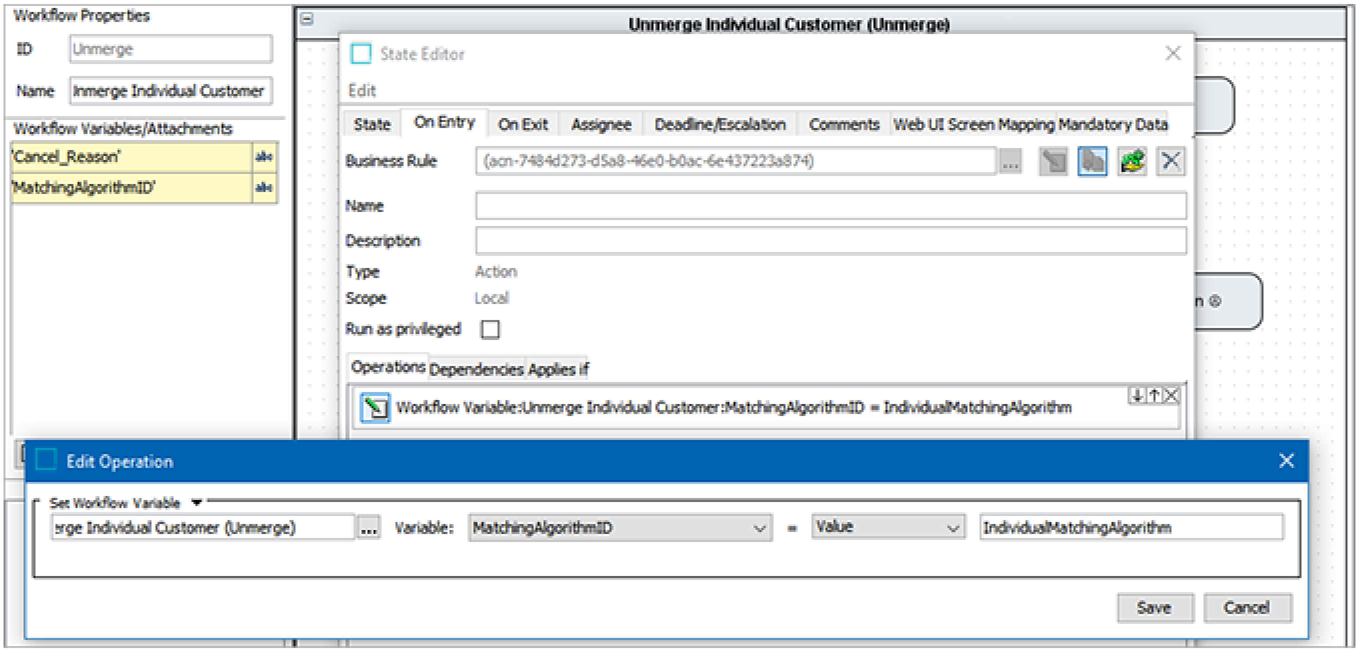
> **Note:** The unmerge workflow below is an example of a complex unmerge workflow. The only requirements for an unmerge workflow are the **matching algorithm ID** and the **object type validity**.

1. On System Setup, create a new workflow for unmerge. For information on setting up a new workflow, refer to the **Creating a Workflow** topic in the **Workflows** documentation.

2. Create the required states: 'Initial', 'Final', and 'Unmerge'. Additional state can be added as needed.
3. In the ‘Workflow Variables/Attachments’ area, click the **Add Workflow Variable** button, set the ID to ‘MatchingAlgorithmID’. Click **OK** to close the dialog, click the File menu and click **Save**.



4. Right-click the Unmerge state in the workflow, click **Edit State**, and make the following updates:
   - On the OnEntry tab click the **Add new Business Action** link.
   - Click the **Edit Operation** button and select **Set Workflow Variable** from the dropdown.
   - In the parameters, select the current workflow, the **MatchingAlgorithmID** variable, 'Value' from the dropdown, and the ID of the matching algorithm. For more information, refer to the **Workflow Variables** topic in the **Workflows** documentation.



Workflow Properties

ID: Unmerge

Name: Inmerge Individual Customer

Workflow Variables/Attachments: 'Cancel_Reason'; 'MatchingAlgorithmID'

Unmerge Individual Customer (Unmerge)

State Editor

Business Rule: (acn-7484d273-d5a8-46e0-b0ac-6e437223a874)

Name:

Description:

Type: Action

Scope: Local

Run as privileged ☐

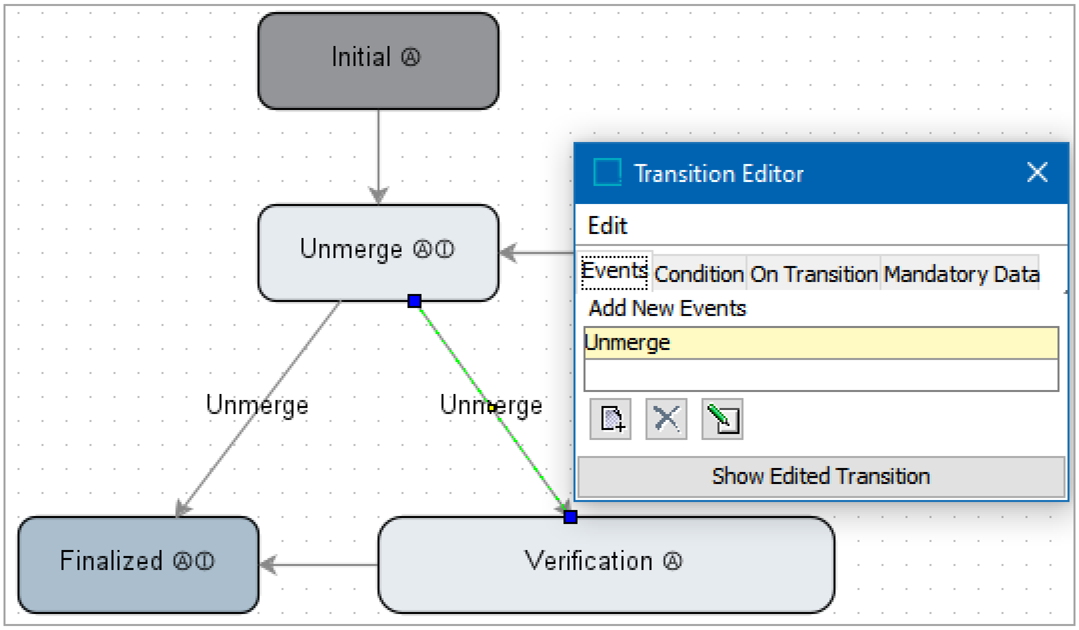
Workflow Variable:Unmerge Individual Customer:MatchingAlgorithmID = IndividualMatchingAlgorithm

Edit Operation

Set Workflow Variable: ırge Individual Customer (Unmerge) ... Variable: MatchingAlgorithmID = Value IndividualMatchingAlgorithm

5. If a transition out of the Unmerge state exists but is not valid for a completed unmerge operation, edit the valid transitions to add events named 'Unmerge' to ensure the expected data flow.

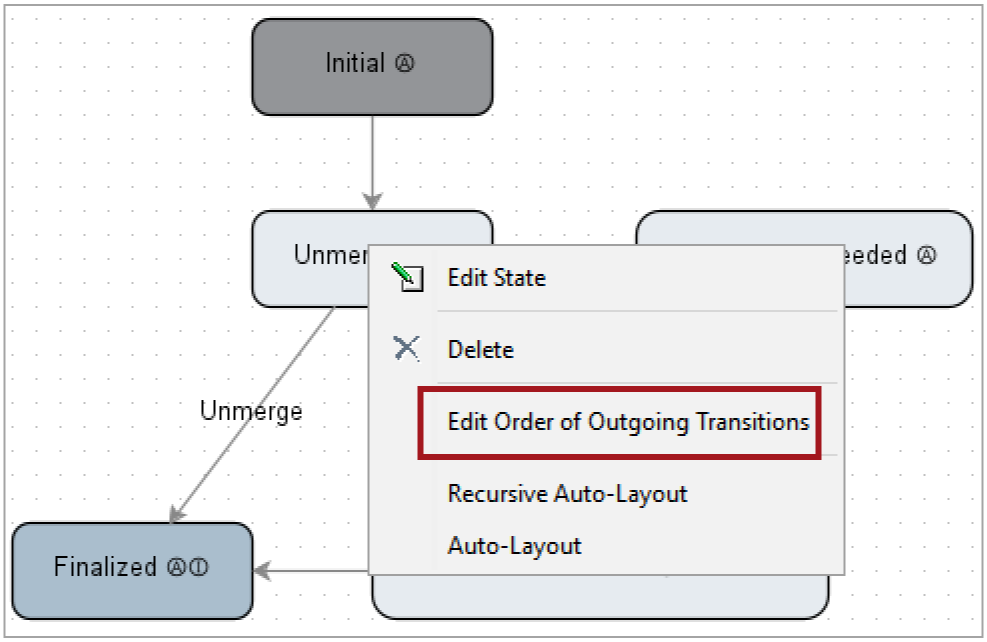
> **Note:** Only transitions (one or more) with events named 'Unmerge' are used when completing the unmerge operation.



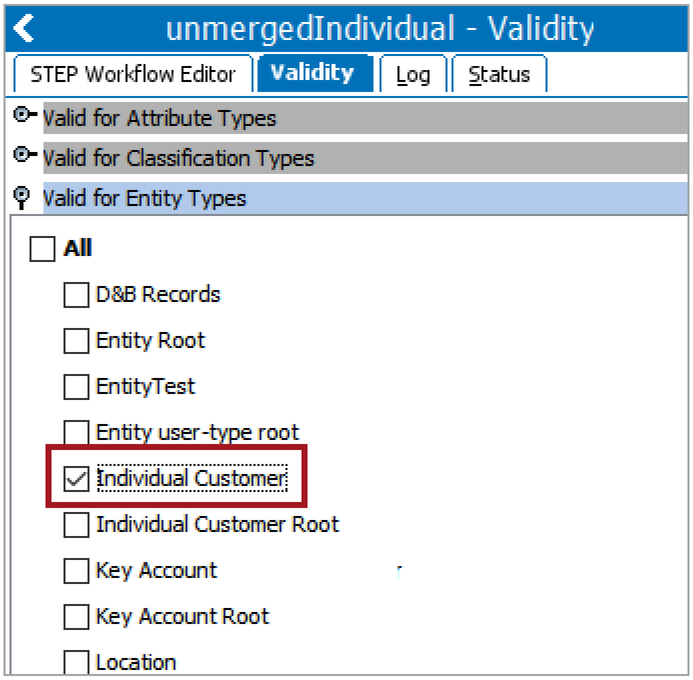
Transition Editor

Events: Unmerge

6. If required, set the order that the transitions should be evaluated on the 'Unmerge' state. Right-click a state, select **Edit Order of Outgoing Transitions** and arrange the outgoing options.



7. On the Workflow, click the Validity tab, and select the unmerge object type.

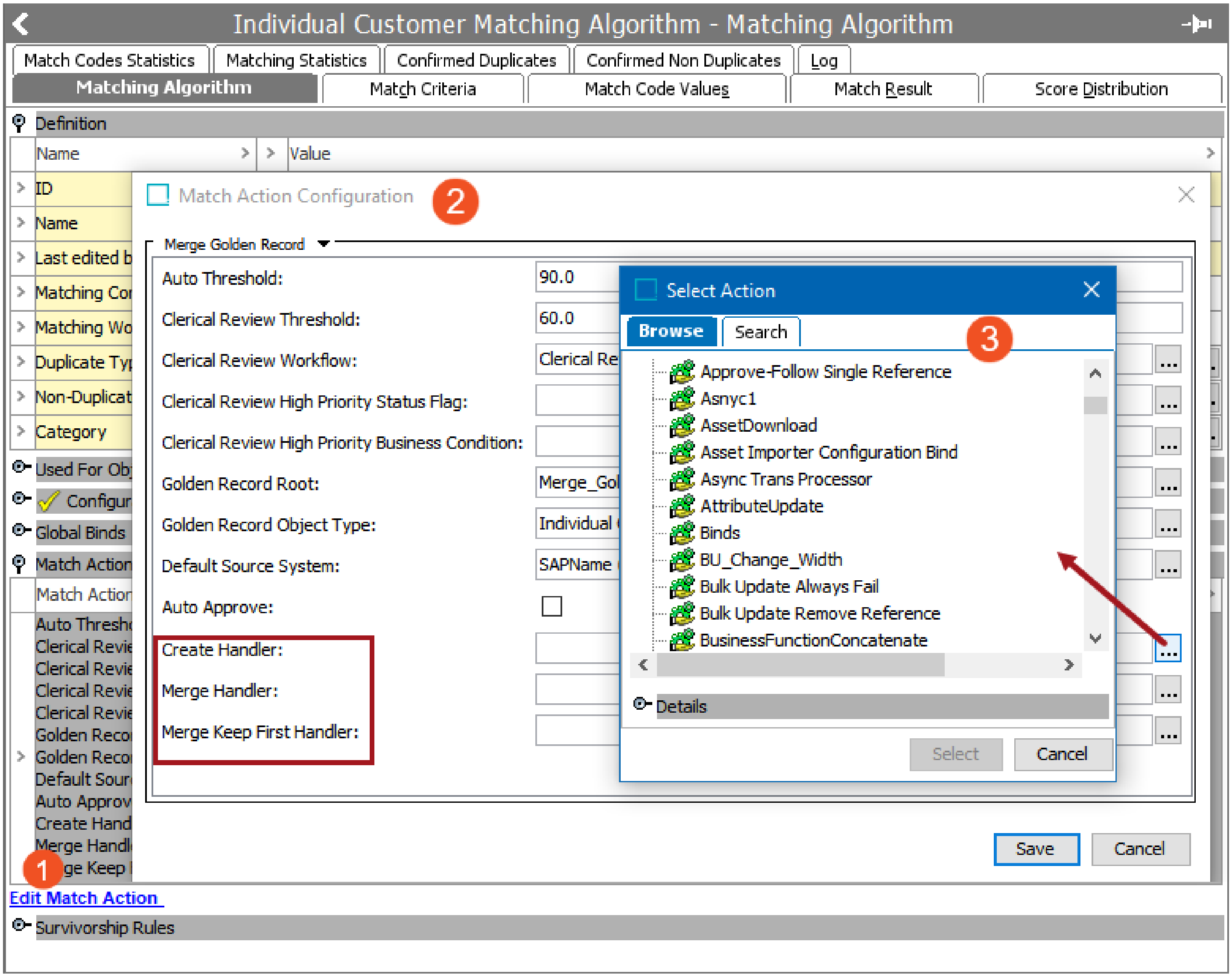

unmergedIndividual - Validity
STEP Workflow Editor
Validity
Log
Status
Valid for Attribute Types
Valid for Classification Types
Valid for Entity Types

☐ All
- ☐ D&B Records
- ☐ Entity Root
- ☐ EntityTest
- ☐ Entity user-type root
- ☑ Individual Customer
- ☐ Individual Customer Root
- ☐ Key Account
- ☐ Key Account Root
- ☐ Location

Configuring Merge Golden Record Match Action Handlers

The match action configuration includes handlers which are invoked at specific times during processing.

> **Note:** These handlers are optional and may not be needed in all solutions.



Auto Threshold: 90.0

Clerical Review Threshold: 60.0

Clerical Review Workflow: Clerical Re

Clerical Review High Priority Status Flag:

Clerical Review High Priority Business Condition:

Golden Record Root: Merge_Gol

Golden Record Object Type: Individual

Default Source System: SAPName

Auto Approve: □

Create Handler:

Merge Handler:

Merge Keep First Handler:

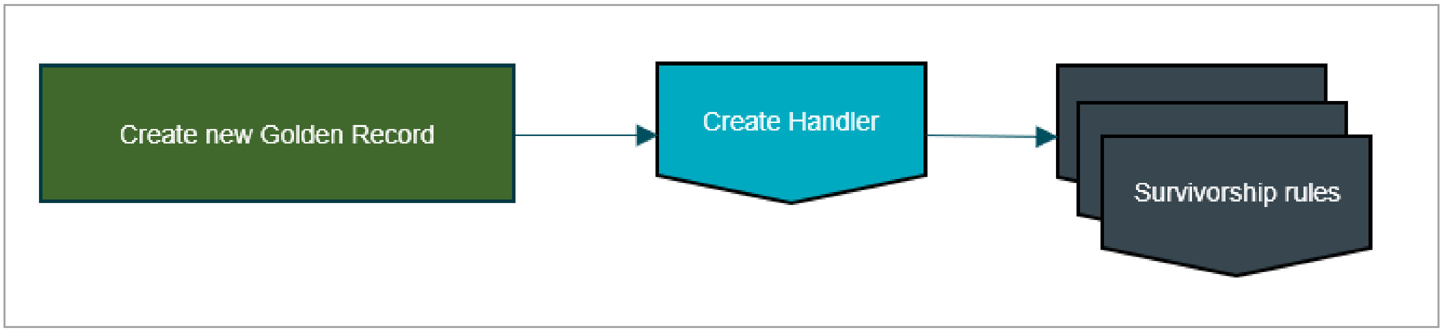
- Approve-Follow Single Reference
- Asnyc1
- AssetDownload
- Asset Importer Configuration Bind
- Async Trans Processor
- AttributeUpdate
- Binds
- BU_Change_Width
- Bulk Update Always Fail
- Bulk Update Remove Reference
- BusinessFunctionConcatenate

For all handlers, the supplied golden records are retrieved by the STEP manager with the context and workspace defined by the matching algorithm. Even if the Approved workspace is selected, the Main workspace is used since changes are not allowed in the Approved workspace.

Create Handler

Any business action added as a Create Handler is run on the golden record after it has been created but before survivorship rules run. This action is only called when a new golden record is created, not when the source record is merged into an identified existing golden record. For more information, refer to the **Match and Merge Flow Details** topic.

- **Input**: The newly created golden record is bound to the 'Current Object' parameter. Refer to the **Current Object Bind** topic in the online help **Resource Materials** documentation.



Merge Handlers

When merging golden records, the Merge Handler and the Merge Keep First Handler are available.

In Match and Merge, existing golden records can be merged as follows:

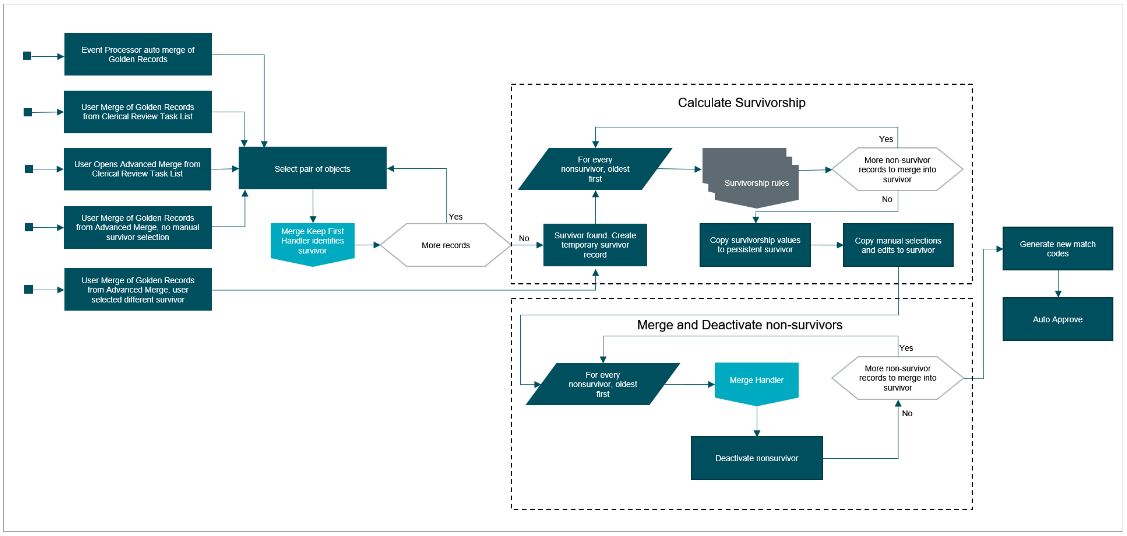
- Automatically - After an update, if the Event Processor detects that two records match each other with a score above the auto-merge threshold.
- Manually - When the records are in clerical review. This requires a user take the merge action from the Clerical Review Task List or Advanced Merge screen.

For more detail on the flow of these merges, refer to the **Match and Merge Flow Details** topic.

While the flows are very similar, the following scenarios differ slightly:

- When the user of Advanced Merge manually selects the surviving record, the Merge Keep First Handler is entirely skipped.
- When loading the Advanced Merge screen, the full merge procedure runs, including a call into the handlers. The result is not saved but is used to populate the screen. As a consequence, the Merge Handler should not make calls to external systems.

View this topic in online help to explore this flowchart.


Event Processor auto merge of Golden Records
User Merge of Golden Records from Clerical Review Task List
User Opens Advanced Merge from Clerical Review Task List
User Merge of Golden Records from Advanced Merge, no manual survivor selection
User Merge of Golden Records from Advanced Merge, user selected different survivor
Select pair of objects
Merge Keep First Handler identifies survivor
More records
Yes
No
Calculate Survivorship
For every nonsurvivor, oldest first
Survivorship rules
More non-survivor records to merge into survivor
Yes
No
Survivor found. Create temporary survivor record
Copy survivorship values to persistent survivor
Copy manual selections and edits to survivor
Generate new match codes
Auto Approve
Merge and Deactivate non-survivors
For every nonsurvivor, oldest first
Merge Handler
More non-survivor records to merge into survivor
Yes
No
Deactivate nonsurvivor

Merge Keep First Handler

This business condition is called to determine which record will be the survivor. If no Merge Keep First Handler is provided, the oldest record in STEP survives.

- **Input 1**: One golden record is bound to the 'Current Object' parameter. Refer to the **Current Object Bind** topic in the online help **Resource Materials** documentation.
- **Input 2**: Another golden record to be deactivated / deleted is bound to the 'Secondary Object' parameter. Refer to the **Secondary Object Bind** topic in the online help **Resource Materials** documentation.
- **Output**: True if the Golden Record bound to the Current Object should survive; false if the object bound to Secondary Object Bind should survive.

Merge Handler

This business action is called when the survivorship rules have merged the information of the two entities, but before the non-surviving entity is deactivated.

- Input 1: The newly updated survivor golden record is bound to the 'Current Object' parameter. Refer to the **Current Object Bind** topic in the online help **Resource Materials** documentation.
- Input 2: The golden record to be deactivated / deleted is bound to the 'Secondary Object' parameter. Refer to the **Secondary Object Bind** topic in the online help **Resource Materials** documentation.

Note: The Merge Handler is designed to make updates on the survivor. It should not be used to call external systems. When loading the Advanced Merge screen, the Merge Handler is called, but the user may cancel the merge. This is handled by STEP by not committing the change made to the survivor by the Merge Handler.

Configuring Merge Golden Record Match Action

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

The Merge Golden Record match action is the part of the matching algorithm that defines the thresholds for records to be merged, the object and reference types used to identify golden records, and the action that should be taken when a golden record is created, deleted, or merged.

Prerequisites

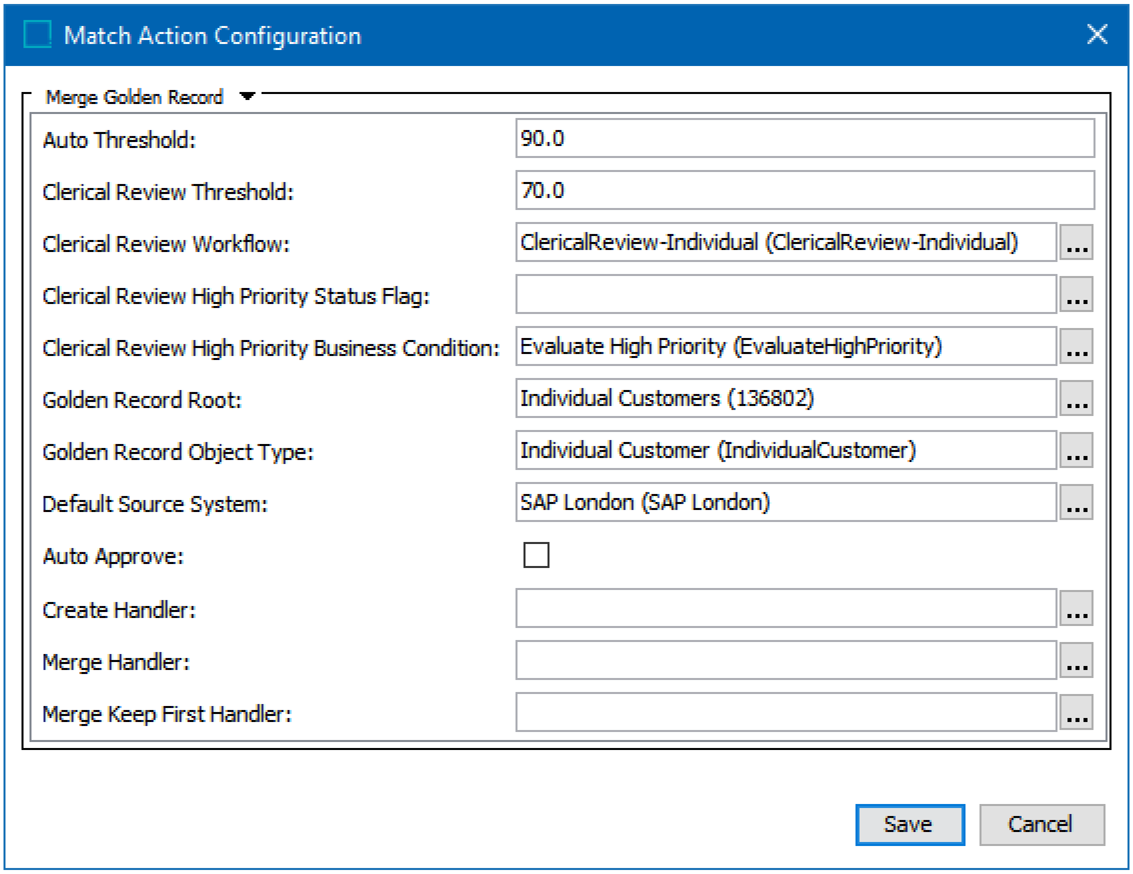
Create or identify the following objects:

- Matching Algorithm as defined in the **Configuring Matching Algorithms** topic.
- Clerical review workflow as defined in the **Creating a Merge Golden Record Clerical Review Workflow** topic.
- Match action handlers as defined in the **Creating Merge Golden Record Match Action Handlers** topic.

Configuration

To configure the merge golden record match action, follow these steps:

1. Open the matching algorithm and click the 'Matching Algorithm' tab.
2. Open the 'Match Action' flipper and click the **Edit Match Action** link to display the 'Match Action Configuration' dialog.
3. Select **Merge Golden Record** from the dropdown and provide the data for the following parameters:



Match Action Configuration

Merge Golden Record

Auto Threshold:	90.0
Clerical Review Threshold:	70.0
Clerical Review Workflow:	ClericalReview-Individual (ClericalReview-Individual) ...
Clerical Review High Priority Status Flag:	...
Clerical Review High Priority Business Condition:	Evaluate High Priority (EvaluateHighPriority) ...
Golden Record Root:	Individual Customers (136802) ...
Golden Record Object Type:	Individual Customer (IndividualCustomer) ...
Default Source System:	SAP London (SAP London) ...
Auto Approve:	□
Create Handler:	...
Merge Handler:	...
Merge Keep First Handler:	...

For information on a parameter, hover over the parameter label to display help text.

- **Auto Threshold** - add a match score (percentage) to indicate how equal two objects must be to automatically merge them. For more information, refer to the **Match and Merge** topic.
- **Clerical Review Threshold** - add a match score (percentage) lower than the Auto Threshold to indicate how equal two objects must be to enter the Clerical Review Workflow (where potential duplicates are manually addressed). For more information, refer to the **Match and Merge** topic.
- **Clerical Review Workflow** - click the ellipsis button (**...**) and select the relevant clerical review workflow. For more information, refer to the **Creating a Merge Golden Record Clerical Review Workflow** topic.
- **Clerical Review High Priority Status Flag** - click the ellipsis button (**...**) and select the workflow status flag that is used to designate high priority tasks in the clerical review workflow. For more information, refer to the **Creating a Merge Golden Record Clerical Review Workflow** topic.
- **Clerical Review High Priority Business Condition** - click the ellipsis button (**...**) and select the business condition that is used to verify if a task is of high priority. For more information, refer to the **Creating a Merge Golden Record Clerical Review Workflow** topic.

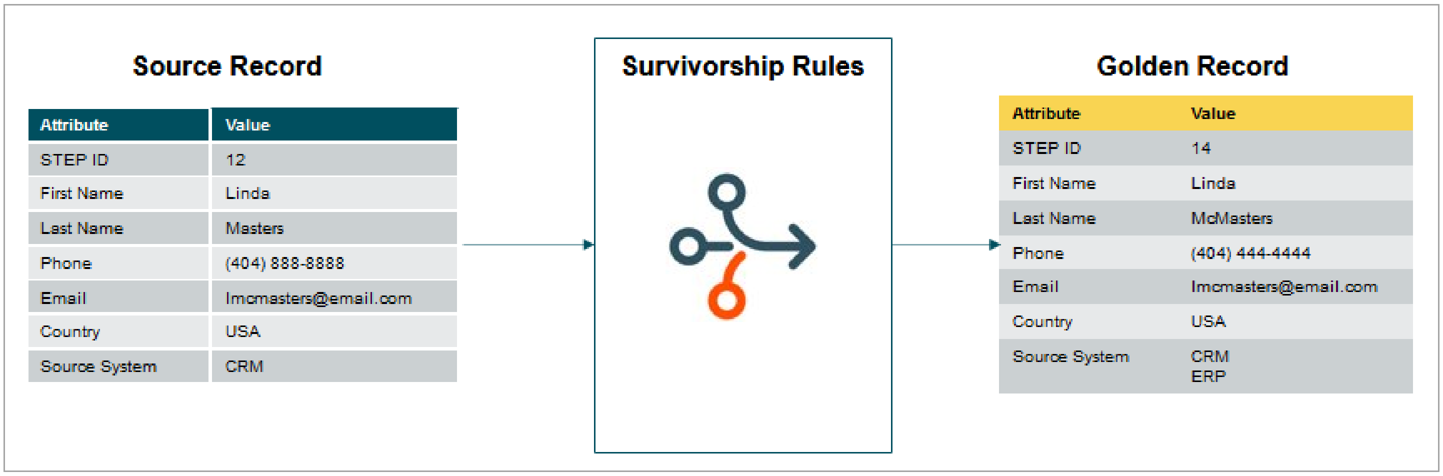
- **Golden Record Root** - specify the Tree location created to hold the golden records. For more information, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.
- **Golden Record Object Type** - specify the object type selected for golden records. For more information, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.
- **Default Source System** - select the source system that should be used if no source system information is available upon import / merging of records. Match and Merge supports the import of records without source system references. For more information, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.
- **Auto Approve** - check to automatically approve the golden records being created.
- **Create Handler** - select a business action to run on the golden record after it has been created but before survivorship rules run.
- **Merge Handler** - select a business action to run when two golden records are merged. This business action runs after the surviving record has been determined and the record to be deactivated has been merged.
- **Merge Keep First Handler** - select a business condition to determine which golden record survives when two golden records are being merged. If the business condition evaluates 'True', it keeps the first golden record; 'False' keeps the second golden record. If this handler is not used, the default behavior keeps the golden record that was created first.

4. Click the **Save** button.

Survivorship in Match and Merge

In match and merge, survivorship rules promote information from exactly one source to exactly one target by comparing information from the source with information from the target and writing the relevant updates to the target.

- In the match and merge IIEP and match and merge web service endpoint, information is promoted from incoming entities to existing or newly created golden records.
- In the matching event processing and in the clerical review Web UI, information is promoted from non-surviving golden records to surviving golden records as those records are merged.
- In the unmerge Web UI actions, as the association between source records and golden records are changed, the content of the resulting golden records is resolved.



Source Record

Attribute	Value
STEP ID	12
First Name	Linda
Last Name	Masters
Phone	(404) 888-8888
Email	lmcmasters@email.com
Country	USA
Source System	CRM

Golden Record

Attribute	Value
STEP ID	14
First Name	Linda
Last Name	McMasters
Phone	(404) 444-4444
Email	lmcmasters@email.com
Country	USA
Source System	CRM ERP

Keep in mind the difference between initial modifications of a golden record and an update to a golden record. An initial modification is when the source system supplies source records without knowing the golden record within STEP. This is an unconnected source, and it needs the normal trusted source priority to work. An update modification is when the source system supplies a source record while knowing which golden record to merge it with. This is a connected source, as it has picked up on a golden record feedback loop from STEP. In this case, these connected sources are treated equally.

Important: Survivorship on values for **Externally Maintained Attributes** is not recommended since survivorship logic depends on revision traceability. Externally maintained values may not figure correctly in the traceability view in Web UI, may be survived wrongly, and in some cases, this can lead to errors in survivorship rules when writing the values.

For more information, refer to the **Configuring Survivorship Rules** topic.

Configuring the Match Data Exchange Method

As defined in the following sections, a match and merge solution communicates with external systems using either an asynchronous IIEP or a synchronous web service setup.

For a detailed explanation of how inbound records are identified as either updates to existing records or creation of new records, refer to the **Inbound Record Flow** section of the **Match and Merge Flow Details** topic.

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

Asynchronous Merge Inbound Integration Endpoint

Data can flow into STEP via an asynchronous inbound integration endpoint (IIEP). The IIEP is designed to receive large batches of source records from any of a number of Receiver plugins.

The incoming source data is translated into STEPXML import files. These input files are typically handled one at a time in sequence, according to the parallel settings of the IIEP queue, as defined in the **IIEP - Configure Endpoint** topic of the **Data Exchange** documentation. The result of the import operation is logged in workbench on the IIEP configuration's Background Processes tab and on the background process execution log.

Any failed records are stored on the BGP in a separate error file which allows the failed updates to be reattempted when errors have been corrected.

For configuration details, refer to the **Match and Merge IIEP Configuration** section of the **IIEP - Configure Match and Merge Importer Processing Engine** topic in the **Data Exchange** documentation.

Synchronous Match and Merge Web Service Endpoint

The match and merge synchronous web service endpoint is an alternative to the asynchronous IIEP. It delivers an answer to each request, alerting the external system to the result of the match and merge operation.

The request sent to this service includes the following information:

- User name and password for access validation.
- A reference to a STEP context.
- A reference to a STEP Match and Merge web service endpoint.
- Entity representations of each record to be imported. Non-duplicates can be declared via the non-duplicate reference types, as defined by the matching component model.

The web service receives a request and completes the following process on incoming data:

1. **Validation -** Ensures minimum data requirements are satisfied (e.g., record has an address or a last name). Records that are not successfully validated are rejected and not stored in STEP.
2. **Standardization -** Standardizes data based on the configuration (e.g., address standardization).

3. **Matching** - Identifies existing record matches and potential record matches. The outcome is one of the following:
    - new or updated golden records in STEP
    - rejection from the web service

    In all cases the web service response includes if:

    - the incoming record was validated.
    - any potential duplicates were found.
    - there is new / updated information on the record itself.
    - the record will be handled manually in a clerical review workflow.

The following topics include more information on:

- Web service endpoints - refer to the **Web Service Endpoints** topic in the **Data Exchange** documentation.
- Web service merging configuration details - refer to the **Web Service Endpoint - Match and Merge** topic in the **Data Exchange** documentation.

Parallel Constraints

STEP imports use several users to import records in parallel. STEP tries to avoid two users updating the same golden record simultaneously, which will slow down imports. Often, input files contain a series of records where the sequences of records are updates of the same contact. STEP use parallel constraints to avoid running these in parallel. Avoiding a series of updates for the same records in the input data can sometimes improve performance.

There are two kinds of parallel constraints: strict parallel constraints and relaxed parallel constraints.

- **Strict:** If a golden record or source system record ID exists in the input, STEP adds it as a strict parallel constraint. The STEP import cannot benefit from multiple if too many consecutive records in the input and/or all have the same source record ID for the same source system.
- **Relaxed:** When you initiate two or more golden records through a match and merge matching algorithm, the algorithm calculates match codes for those golden records. STEP then adds parallel constraints based on those match codes, preventing entities with overlapping match codes from importing simultaneously. However, STEP adds the calculated match codes as a relaxed parallel constraint. When checking constraints, STEP will start ignoring relaxed constraint values that have occurred too frequently (e.g., thousands of contacts all use the same reception phone number, so STEP ignores this repeated value).

Configuring the Merge Event Processor

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

An event processor monitors the system for actionable events on specified objects, ensures match codes are regenerated, and runs the matching algorithms in response to any relevant change. For example, consider an object that is subject to a matching algorithm. When the match code assignment or data on that object is approved, the approval can trigger the event processor to regenerate the match code for that object and run the algorithm. Alternatively, events can be passed to the event processor via a republish business rule as part of a workflow or integration.

Event processors write to a background process log so you can identify when events were processed and what actions were taken in response. Additionally, event processor performance measurements are available on the Statistics tab for both matching algorithms and match code configurations.

The merge event processor compares golden records that already exist in the system and initiates possible duplicates into the merge clerical review workflow.

> **Important:** It is recommended to use a single matching event processor to handle events across all matching algorithms.

Configuration

To configure an event processor for a matching solution:

1. Create a matching event processor as defined in the **Creating an Event Processor** topic and the **Matching Processing Plugin Parameters and Triggers** topic of the **System Setup** documentation.
2. In System Setup, open your event processor and review the following parameter settings:

   Open the Configuration flipper and click the **Edit Configuration** link to display the wizard.

   - On the Configure Event Processor step, verify the Select Processor parameter is set to 'Matching'
   - On the Configure Processing Plugin step, verify the Event Processing parameter is set to 'Generate/Update Match Code Values and Run Matching Algorithm'
   - On the Configure Processing Plugin step, verify the Matching Algorithms parameter displays the desired matching algorithm(s)
   - On the Schedule Event Processor step, verify the Start parameter shows the desired schedule (Every 1 minute is recommended.)

   Close the wizard and review the event processor editor.

  - On the Event Processor tab, open the Configuration flipper, and verify the Queue Status parameter is set to Read Events
  - On the Event Triggering Definitions tab, verify the appropriate event triggering definitions are selected

    For a **match and merge** scenario, based on the selected algorithm, for existing golden records, the event processor performs a merge or initiates a clerical review. Add triggers for the following:

    - references defined by your **Matching component model**: Non-Duplicate Reference Types
    - references defined by your **Matching - Merge Golden Record References component model**: Unmerged-From Relation Reference Types
    - attributes, references, and data containers included in your **Match Criteria**

      > **Important:** For accurate match and merge functionality, the event processor must trigger on updates that can change the outcome of the record comparisons. To accomplish this, the recommendation is to trigger on any attribute, reference, or data container that is used in the match criteria.

    For a match and merge scenario, avoid triggers on the following attributes and reference types as defined by your component models:

    - Potential Duplicate Reference Type
    - Merged-Into Relation Reference Types
    - Source Relation Reference Type
    - Potential Duplicate Match Algorithm ID Attribute
    - Source Record ID Attribute
    - Deactivated Attribute

3. Enable the matching event processor as defined in the **Enable Event Processor** section of the **Running an Event Processor** topic in the **System Setup** documentation.

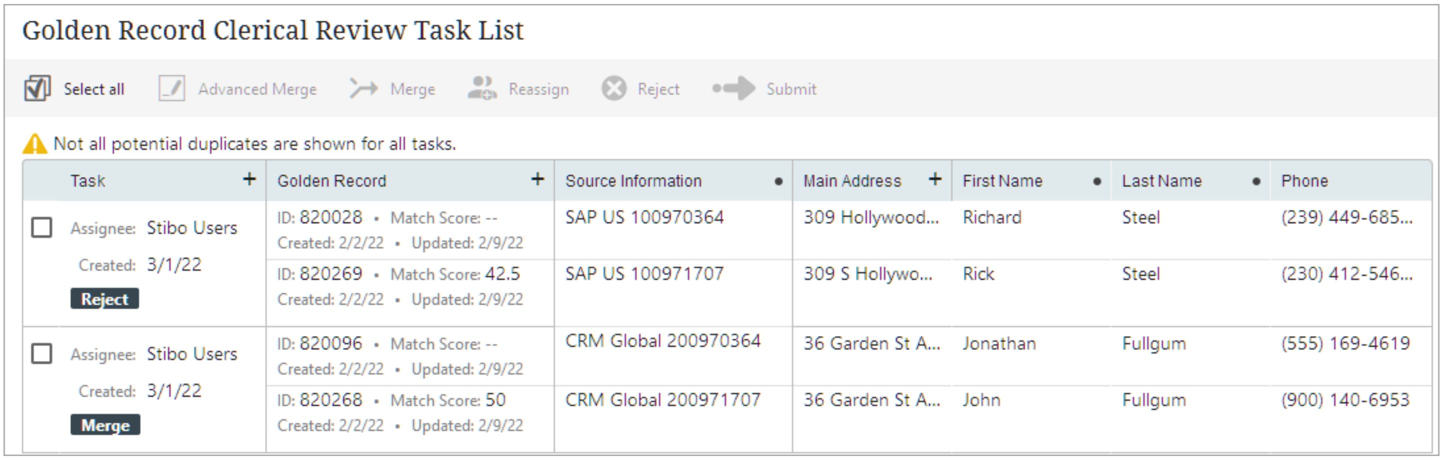
For more information, refer to the **Maintaining an Event Processor** topic of the **System Setup** documentation.

Configuring and Using Match and Merge in Web UI

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

In Web UI, the elements that are available for merging and unmerging are defined in the topics below:

- **Clerical Review Task List** - refer to the **Golden Record Clerical Review Task List** topic.

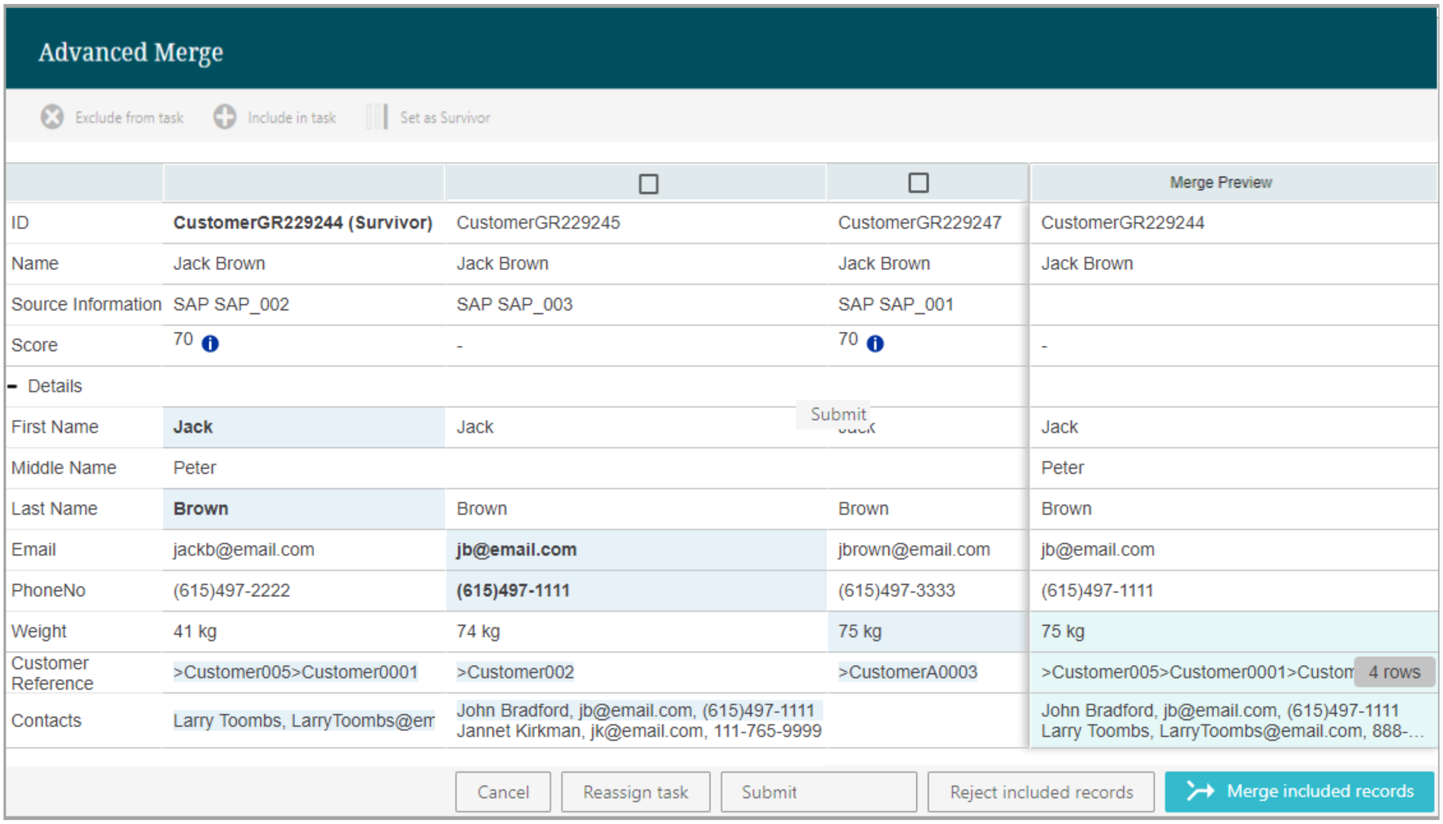


Golden Record Clerical Review Task List

Not all potential duplicates are shown for all tasks.

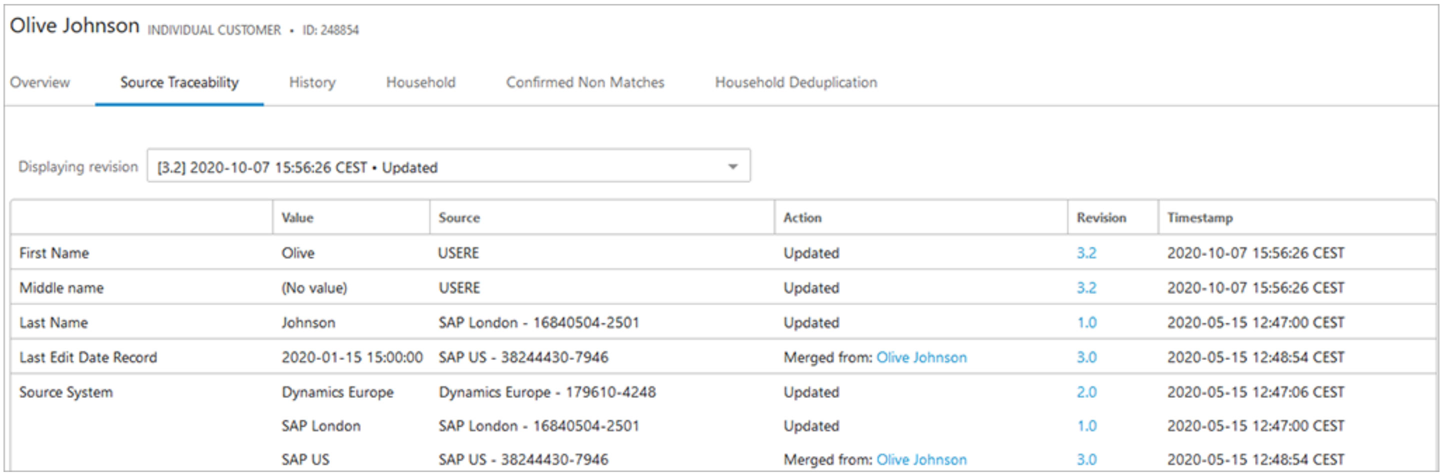
Task	Golden Record	Source Information	Main Address	First Name	Last Name	Phone
□ Assignee: Stibo Users Created: 3/1/22 Reject	ID: 820028 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	SAP US 100970364	309 Hollywood...	Richard	Steel	(239) 449-685...
	ID: 820269 • Match Score: 42.5 Created: 2/2/22 • Updated: 2/9/22	SAP US 100971707	309 S Hollywo...	Rick	Steel	(230) 412-546...
□ Assignee: Stibo Users Created: 3/1/22 Merge	ID: 820096 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	CRM Global 200970364	36 Garden St A...	Jonathan	Fullgum	(555) 169-4619
	ID: 820268 • Match Score: 50 Created: 2/2/22 • Updated: 2/9/22	CRM Global 200971707	36 Garden St A...	John	Fullgum	(900) 140-6953

- **Advanced Merge Dialog** - refer to the **Golden Record Advanced Merge Dialog** topic.



		☐	☐	Merge Preview
ID	**CustomerGR229244 (Survivor)**	CustomerGR229245	CustomerGR229247	CustomerGR229244
Name	Jack Brown	Jack Brown	Jack Brown	Jack Brown
Source Information	SAP SAP_002	SAP SAP_003	SAP SAP_001	
Score	70	-	70	-
− Details				
First Name	**Jack**	Jack	Jack	Jack
Middle Name	Peter			Peter
Last Name	**Brown**	Brown	Brown	Brown
Email	jackb@email.com	**jb@email.com**	jbrown@email.com	jb@email.com
PhoneNo	(615)497-2222	**(615)497-1111**	(615)497-3333	(615)497-1111
Weight	41 kg	74 kg	75 kg	75 kg
Customer Reference	>Customer005>Customer0001	>Customer002	>CustomerA0003	>Customer005>Customer0001>Custom 4 rows
Contacts	Larry Toombs, LarryToombs@em	John Bradford, jb@email.com, (615)497-1111 Jannet Kirkman, jk@email.com, 111-765-9999		John Bradford, jb@email.com, (615)497-1111 Larry Toombs, LarryToombs@email.com, 888-…

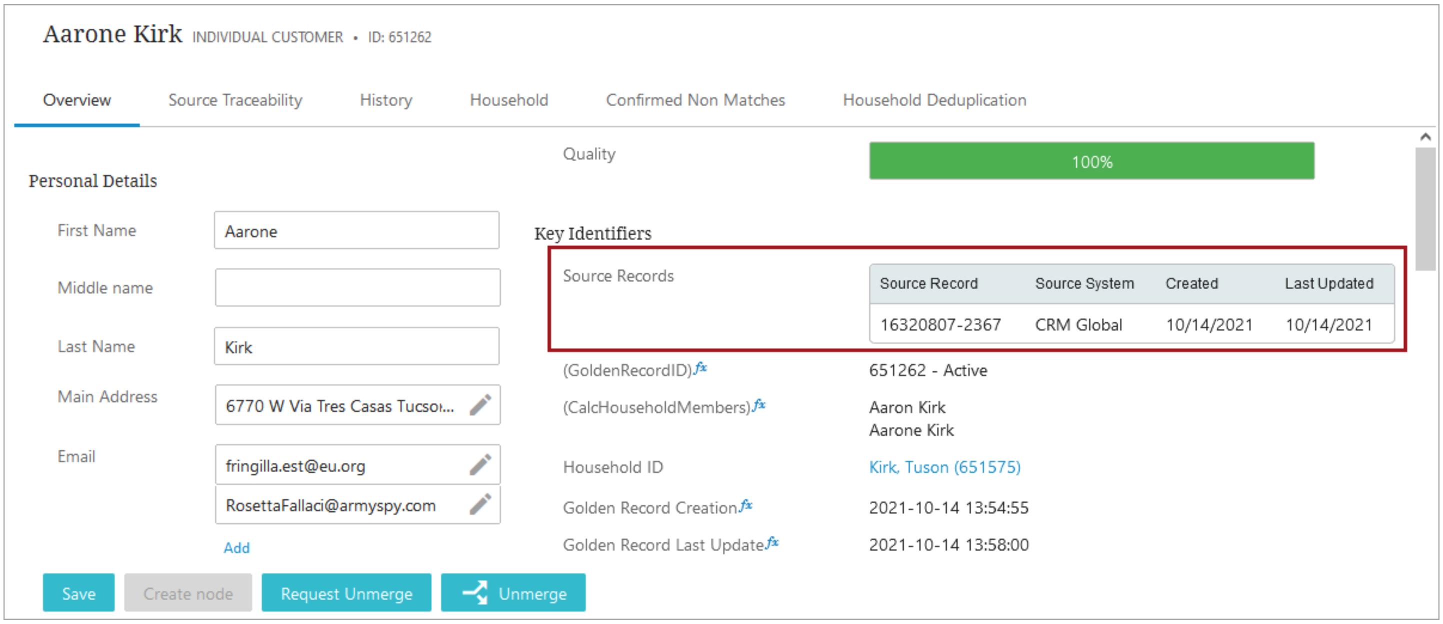
- **Source Traceability Screen** - refer to the **Golden Record Source Traceability Screen** topic.



Displaying revision [3.2] 2020-10-07 15:56:26 CEST • Updated

	Value	Source	Action	Revision	Timestamp
First Name	Olive	USERE	Updated	3.2	2020-10-07 15:56:26 CEST
Middle name	(No value)	USERE	Updated	3.2	2020-10-07 15:56:26 CEST
Last Name	Johnson	SAP London - 16840504-2501	Updated	1.0	2020-05-15 12:47:00 CEST
Last Edit Date Record	2020-01-15 15:00:00	SAP US - 38244430-7946	Merged from: Olive Johnson	3.0	2020-05-15 12:48:54 CEST
Source System	Dynamics Europe	Dynamics Europe - 179610-4248	Updated	2.0	2020-05-15 12:47:06 CEST
	SAP London	SAP London - 16840504-2501	Updated	1.0	2020-05-15 12:47:00 CEST
	SAP US	SAP US - 38244430-7946	Merged from: Olive Johnson	3.0	2020-05-15 12:48:54 CEST

- **Golden Record Source Information Component** - refer to the **Golden Record Source Information** section of the **Match and Merge Traceability** topic.



Aarone Kirk INDIVIDUAL CUSTOMER • ID: 651262

Personal Details

First Name	Aarone
Middle name	
Last Name	Kirk
Main Address	6770 W Via Tres Casas Tucso...
Email	fringilla.est@eu.org RosettaFallaci@armyspy.com

Quality 100%

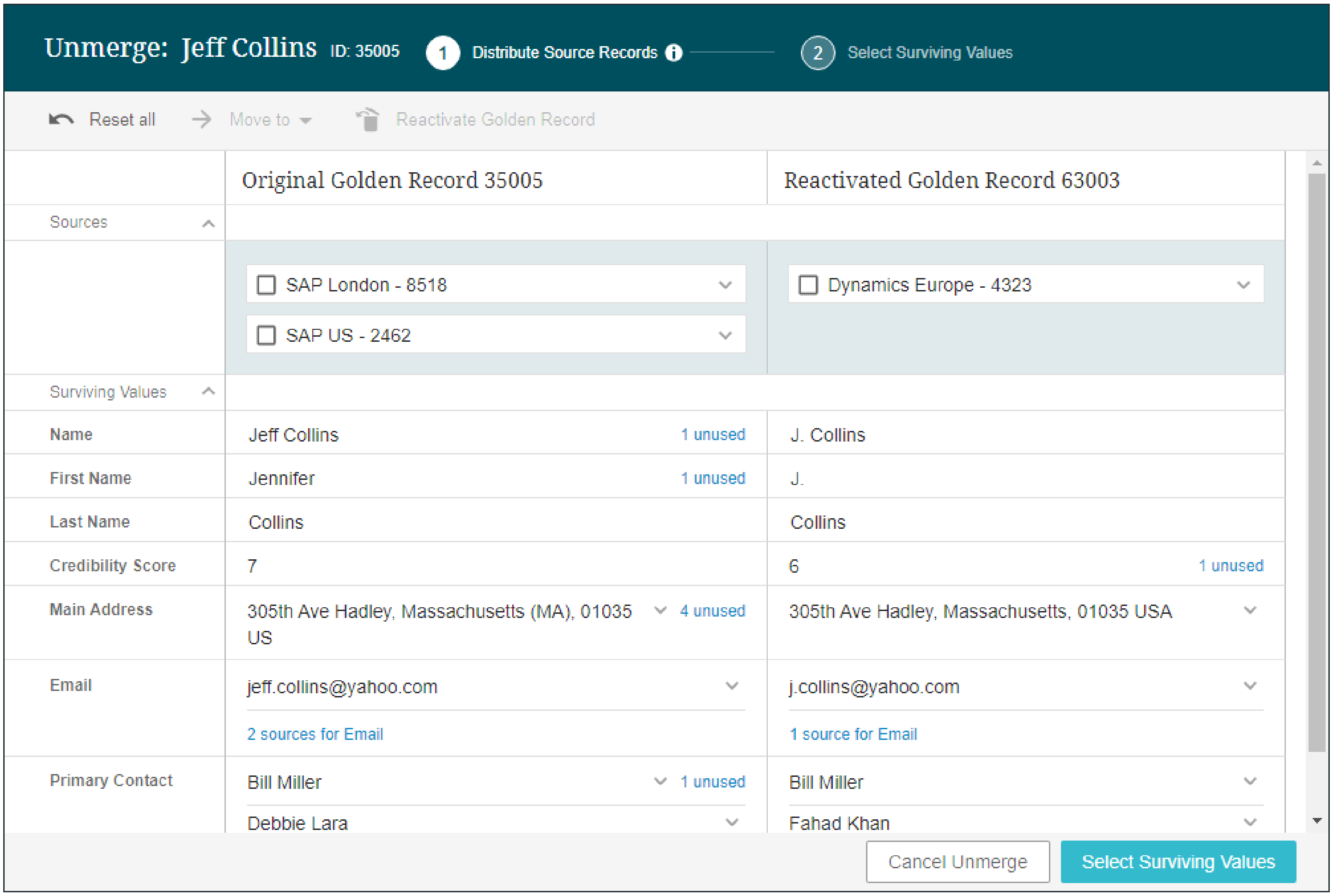
Key Identifiers

Source Records

Source Record	Source System	Created	Last Updated
16320807-2367	CRM Global	10/14/2021	10/14/2021

(GoldenRecordID)*fx*	651262 - Active
(CalcHouseholdMembers)*fx*	Aaron Kirk Aarone Kirk
Household ID	Kirk, Tuson (651575)
Golden Record Creation*fx*	2021-10-14 13:54:55
Golden Record Last Update*fx*	2021-10-14 13:58:00

- **Unmerge Wizard** - refer to the **Configuring the Unmerge Wizard** topic and the **Using the Unmerge Wizard** topic.



Unmerge: Jeff Collins ID: 35005

1 Distribute Source Records

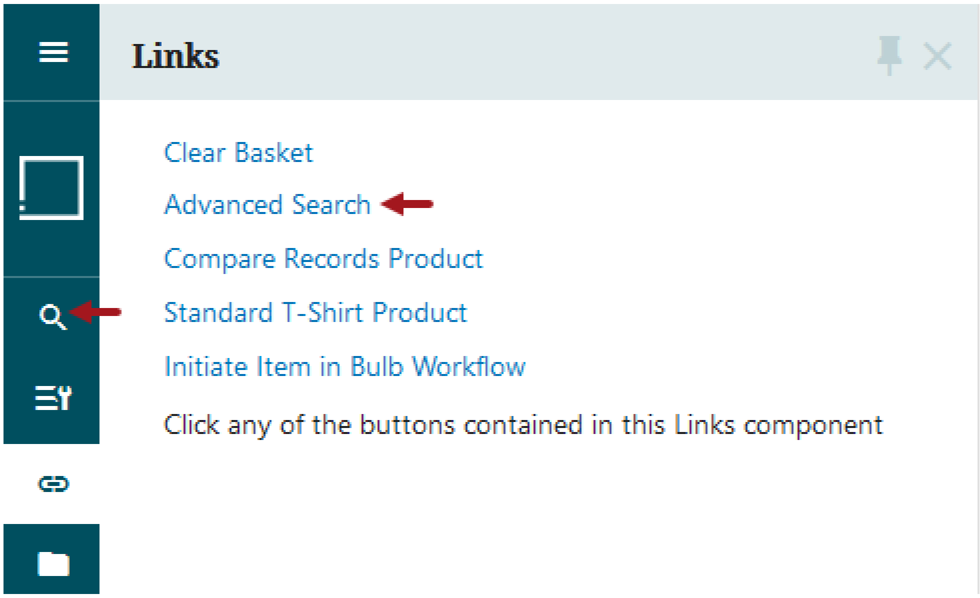
2 Select Surviving Values

Reset all · Move to · Reactivate Golden Record

	Original Golden Record 35005	Reactivated Golden Record 63003
Sources		
	□ SAP London - 8518 □ SAP US - 2462	□ Dynamics Europe - 4323
Surviving Values		
Name	Jeff Collins 1 unused	J. Collins
First Name	Jennifer 1 unused	J.
Last Name	Collins	Collins
Credibility Score	7	6 1 unused
Main Address	305th Ave Hadley, Massachusetts (MA), 01035 US 4 unused	305th Ave Hadley, Massachusetts, 01035 USA
Email	jeff.collins@yahoo.com 2 sources for Email	j.collins@yahoo.com 1 source for Email
Primary Contact	Bill Miller 1 unused Debbie Lara	Bill Miller Fahad Khan

Cancel Unmerge · Select Surviving Values

- **Advanced Search** or **Search**
  - via panel to search for golden records via golden record ID or source record ID - refer to the **Global Navigation Panel** topic in the **Web User Interfaces** documentation.
  - via home page link or widget to search for golden records via golden record ID or source record ID - refer to the **Homepage Widgets** topic in the **Web User Interfaces** documentation.



Links

Clear Basket

Advanced Search

Compare Records Product

Standard T-Shirt Product

Initiate Item in Bulb Workflow

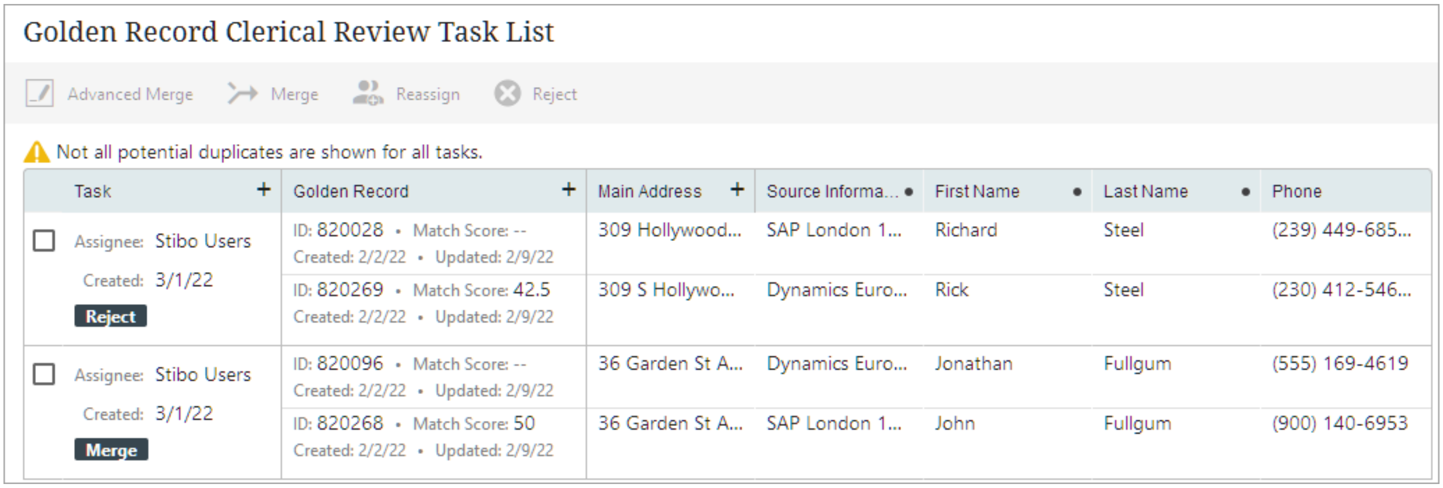
Click any of the buttons contained in this Links component

For information about the 'Confirmed Matches' component screen, refer to the **Confirmed Matches Component** topic.

Golden Record Clerical Review Task List

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

> **Note:** As defined below, the Golden Record Clerical Review Task List must be configured as a result screen for a Status Selector Homepage widget and configured with a Node List and a Display Mode. This display mode should be configured with a specific set of table headers relevant to Golden Records.



Not all potential duplicates are shown for all tasks.

	Task	Golden Record	Main Address	Source Informa...	First Name	Last Name	Phone
☐	Assignee: Stibo Users Created: 3/1/22 Reject	ID: 820028 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	309 Hollywood...	SAP London 1...	Richard	Steel	(239) 449-685...
		ID: 820269 • Match Score: 42.5 Created: 2/2/22 • Updated: 2/9/22	309 S Hollywo...	Dynamics Euro...	Rick	Steel	(230) 412-546...
☐	Assignee: Stibo Users Created: 3/1/22 Merge	ID: 820096 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	36 Garden St A...	Dynamics Euro...	Jonathan	Fullgum	(555) 169-4619
		ID: 820268 • Match Score: 50 Created: 2/2/22 • Updated: 2/9/22	36 Garden St A...	SAP London 1...	John	Fullgum	(900) 140-6953

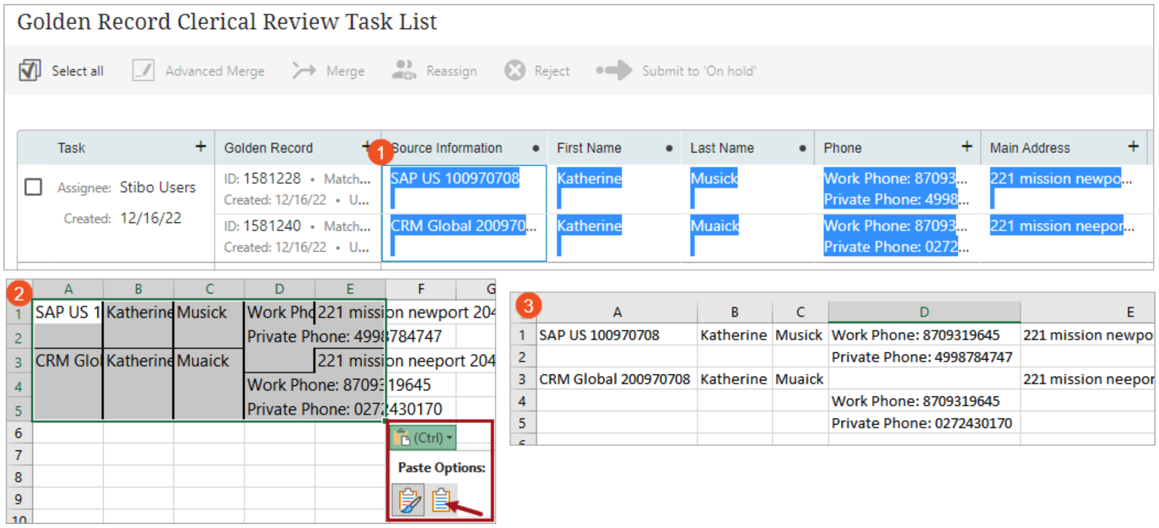
When using a Golden Record Clerical Review Task List screen, consider the following:

- Records do not appear in Clerical Review during import; records display only after a matching event processor has run on all the records.
- Although you can apply a single clerical review workflow to multiple matching algorithms, it is not recommended.
- It is recommended to use a separate user group for this task.

Copy / Paste Table Values

Use the following steps to work with the data outside of the Web UI task list table:

1. Highlight the text from single or multiple cells to copy. All text in the cell is copied, including the hidden text indicated by an ellipsis (...).
2. Paste into an Excel (or CSV) file and use the 'Match Destination Formatting' paste option to remove rule lines displayed in the Web UI.
3. Use additional Excel formatting options to display the text as desired.



Task	Golden Record	Source Information	First Name	Last Name	Phone	Main Address
□ Assignee: Stibo Users Created: 12/16/22	ID: 1581228 • Match... Created: 12/16/22 • U...	SAP US 100970708	Katherine	Musick	Work Phone: 87093... Private Phone: 4998...	221 mission newpo...
	ID: 1581240 • Match... Created: 12/16/22 • U...	CRM Global 200970...	Katherine	Muaick	Work Phone: 87093... Private Phone: 0272...	221 mission neepor...

	A	B	C	D	E	F	G
1	SAP US 1	Katherine	Musick	Work Pho	221 mission newport 204		
2				Private Phone: 4998784747			
3	CRM Glo	Katherine	Muaick		221 mission neeport 204		
4				Work Phone: 8709319645			
5				Private Phone: 0272430170			
6							
7							
8							
9							
10							

	A	B	C	D	E
1	SAP US 100970708	Katherine	Musick	Work Phone: 8709319645	221 mission newpo
2				Private Phone: 4998784747	
3	CRM Global 200970708	Katherine	Muaick		221 mission neepor
4				Work Phone: 8709319645	
5				Private Phone: 0272430170	

Note: Values cannot be copied / pasted from the Advanced Merge screen.

Prerequisites

It is expected that anyone configuring the Golden Record Clerical Review Task List component is familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** topic of the **Web User Interfaces** documentation.

Note: If you split your data model into multiple entity types (i.e., individual customers versus organization customers), you must configure a separate Clerical Review Task List for each entity type.

This topic includes details on **Configuration**, **Using Action Buttons**, **Filtering Task List**, and **MLMR Support Guidelines**.

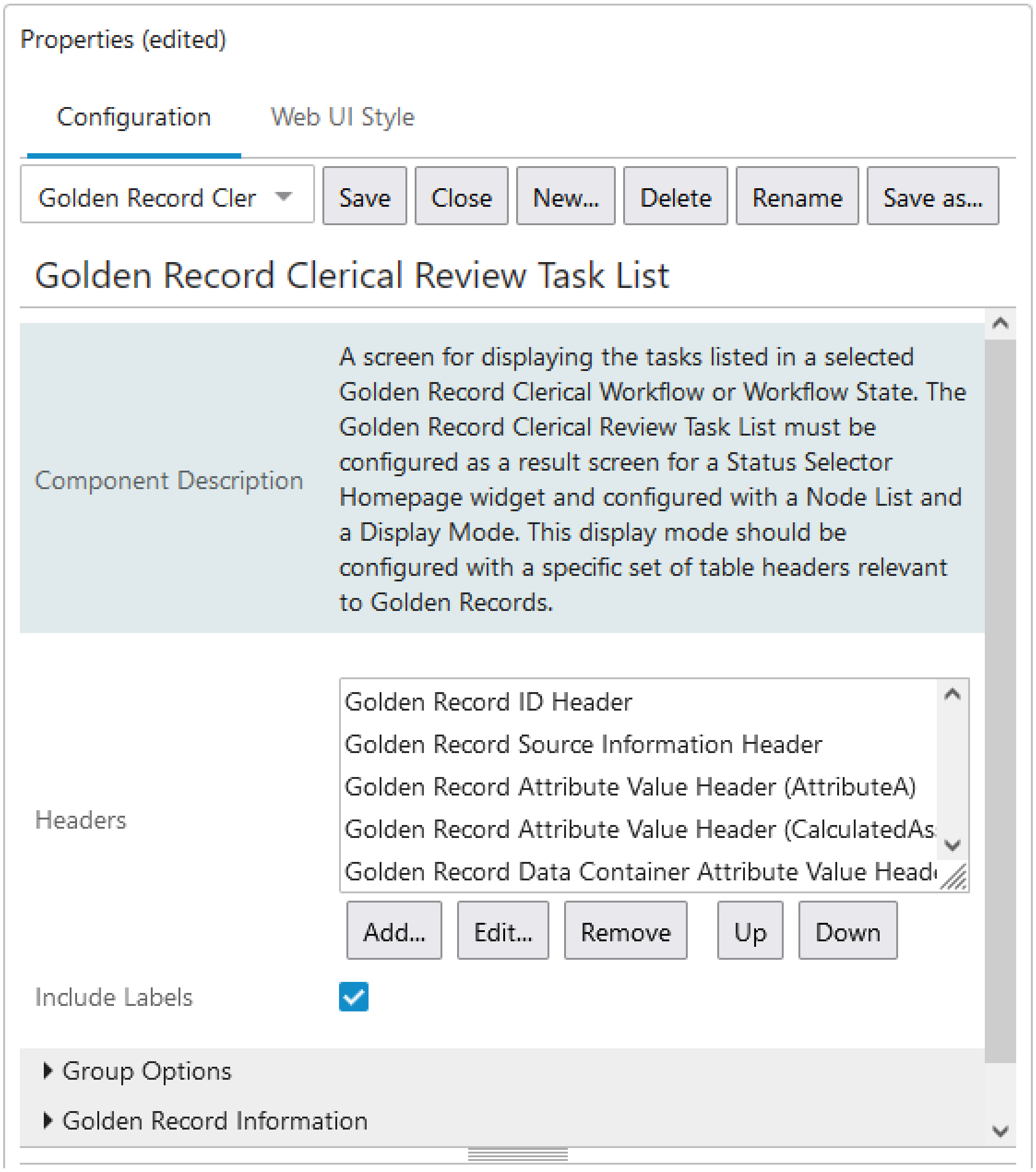
Configuration

The Golden Record Clerical Review Task List can be configured to present the most relevant headers to the reviewer. Additionally, attribute group display options and configurable action buttons are available.

Use the following steps to add and configure the screen.

1. Open the designer and click **New...**.

2. Select 'Golden Record Clerical Review Task List', enter a Screen ID, and click **Add**.

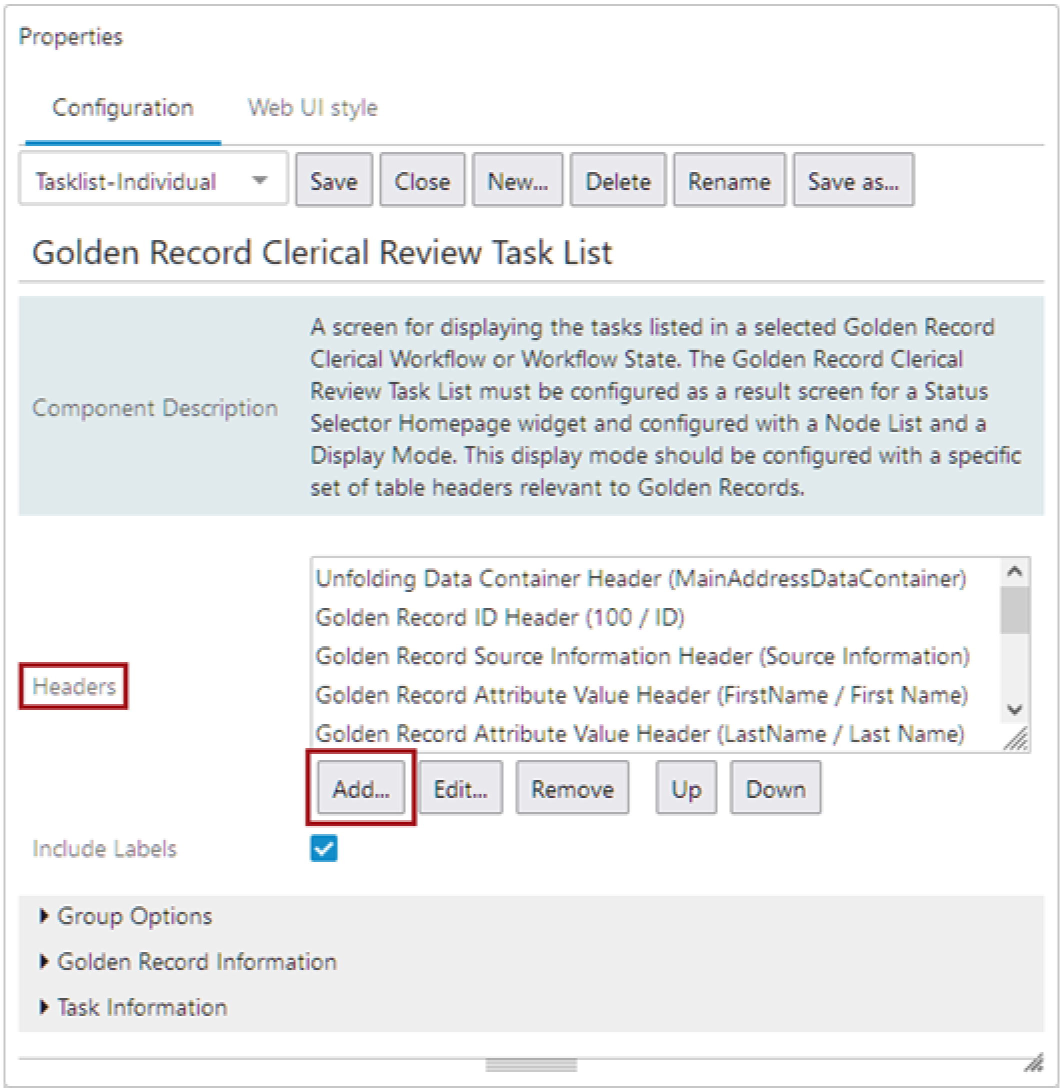


Component Description: A screen for displaying the tasks listed in a selected Golden Record Clerical Workflow or Workflow State. The Golden Record Clerical Review Task List must be configured as a result screen for a Status Selector Homepage widget and configured with a Node List and a Display Mode. This display mode should be configured with a specific set of table headers relevant to Golden Records.

Headers:
Golden Record ID Header
Golden Record Source Information Header
Golden Record Attribute Value Header (AttributeA)
Golden Record Attribute Value Header (CalculatedAs
Golden Record Data Container Attribute Value Head

Include Labels ☑

3. For the Headers parameter, click **Add...** and select one of the listed header components. Once a selection is made and required configurations are complete, click the **Add** button.



Properties

Golden Record Clerical Review Task List

Component Description: A screen for displaying the tasks listed in a selected Golden Record Clerical Workflow or Workflow State. The Golden Record Clerical Review Task List must be configured as a result screen for a Status Selector Homepage widget and configured with a Node List and a Display Mode. This display mode should be configured with a specific set of table headers relevant to Golden Records.

Headers:
Unfolding Data Container Header (MainAddressDataContainer)
Golden Record ID Header (100 / ID)
Golden Record Source Information Header (Source Information)
Golden Record Attribute Value Header (FirstName / First Name)
Golden Record Attribute Value Header (LastName / Last Name)

Include Labels ☑

- Golden Record Attribute Value Group Header - Displays the values of the specified group's attributes on the golden records. Attributes added to a selected Attribute Group will automatically be included and displayed. Once selected, additional configuration is required:
    - Attribute Group - Specify the attribute group values to display.
    - Blacklist Attribute Group - Specify the attribute groups **not** to display, even if they also appear in the Attribute Group specified above.
    - Dimensions - Specify height / width of the of the header cell.
    - Included Nested Groups - When checked, attributes from nested parameter groups should be included.
    - Label - Specify a label for the header.
    - Read Only - When checked, the values listed under this header cannot be edited.

  - Show LOV IDs - When checked, relevant LOVs display their IDs next to the corresponding values.
  - Table Sorting - Select a method of sorting the values in the header.
  - Enable Locale Formatting - When checked, 'ISO Date' and 'ISO Date and Time' values are formatted according to locale.
  - Context Help - Enter help text to display for the component.
  - Display Context help - When checked, display context help text for attributes.

- Golden Record Attribute Value Header - Displays the values of the specified attribute on the golden records. Once selected, specify an Attribute, Label, and Preferred Column Width in the configuration dialog. If potential duplicate references are set up for the matching algorithm, this header allows filtering on records, not tasks. Records filtering means that a task is displayed when at least one record in the task meets all filter criteria. For example, filtering on both a 'first name' and a 'last name' returns all tasks where both the selected first and last names are included.

> **Note:** To enable filtering, the Potential Duplicate reference must be configured into the Merge Golden Record Component model. For more information, refer to **Configuring the Matching - Merge Golden Record Component Model Prerequisites** topic.

- Golden Record Data Container Attribute Value Header - Displays the values for the specified data container attribute on the golden records. Once selected, specify a Data Container, Attribute, Label, and Preferred Column Width in Pixels in the configuration dialog.

> **Note:** The data container display attribute must be valid on the data container. Using a calculated attribute that is not valid as an attribute within the data container can cause the system to display the data container ID by default.

- Golden Record ID Header - Displays the IDs of the golden records in the table. Once selected, double click the component and specify a Label and Preferred Column Width in the configuration dialog.

> **Important:** The Golden Record ID Header has been superseded by the Golden Record Information header, described in the steps below.

- Golden Record Name Header - Displays the names of the golden records in the table. Once selected, double click the component and specify a Label and Preferred Column Width in the configuration dialog.
- Golden Record Reference Type Header - Displays the object the golden record references via the specified reference type. Once selected, specify a Reference Type, Label, and Preferred Column Width in the configuration dialog.
- Golden Record Source Information Header - Displays source system and record information for the golden records. Once selected, double click the component and specify a Label and Preferred Column Width in the configuration dialog.

- Unfolding Data Container Header - Displays data container information that unfolds into its own columns to show all data container elements when you click the plus (+) button on the header. Once selected, locate and select the desired data container and specify a Label and Preferred Column Width in the configuration dialog. Additionally, the width of individual element columns is configurable. For more information, refer to the **Configuring Column Width** section below.

  > **Important:** For maximum efficiency in reviewing content, you must configure the Dynamic Table Layout component. To optimize the view for object comparisons, uncheck the 'Allow Wrap Of Header Title' and 'Allow Wrap of Cell Content' parameters. This applies to both the Golden Record Clerical Review Task List and Advanced Merge Dialog. For more information, refer to the **Dynamic Table Layout** topic of the **Web User Interfaces** documentation.

4. For the **Include Labels** parameter, when checked, all toolbar actions on the screen display a label below the icon.

5. For the **Group Options** flipper, set the following options:

   - Groups Initially Open - When checked, the attribute groups are opened by default.

   - Show Empty Group Headers - When checked, the headers are visible if the group is empty.

   - Show Group Headers - When checked, a header for attribute groups are visible.

6. For the **Golden Record Information** flipper, set the following options:

   - Created Date - Select to display the date and time of creation.

   - Last Update Date - Select to display the date and time of the last update.

   - Record Summary Column Width - Specify the width of the unfoldable Golden Record column.

   - ID - Select to display the STEP IDs of the potential duplicates.

   - ID Column Width - Specify the width of the ID column.

   - Match Score - Select to display the match score of the linked golden record towards a workflow node. The workflow node is the top golden record and does not display a match score (as it cannot be matched with itself) and instead displays a '--'. If a potential duplicate is indirectly included (e.g., it matches with one of the linked golden records but not the workflow node), it displays a 'N/A'.

7. For the **Task Information** flipper, set the following options:

   - Assignee - Select to display the assignee for a task.

   - Assignee Column Width - Specify the width of the assignee column.

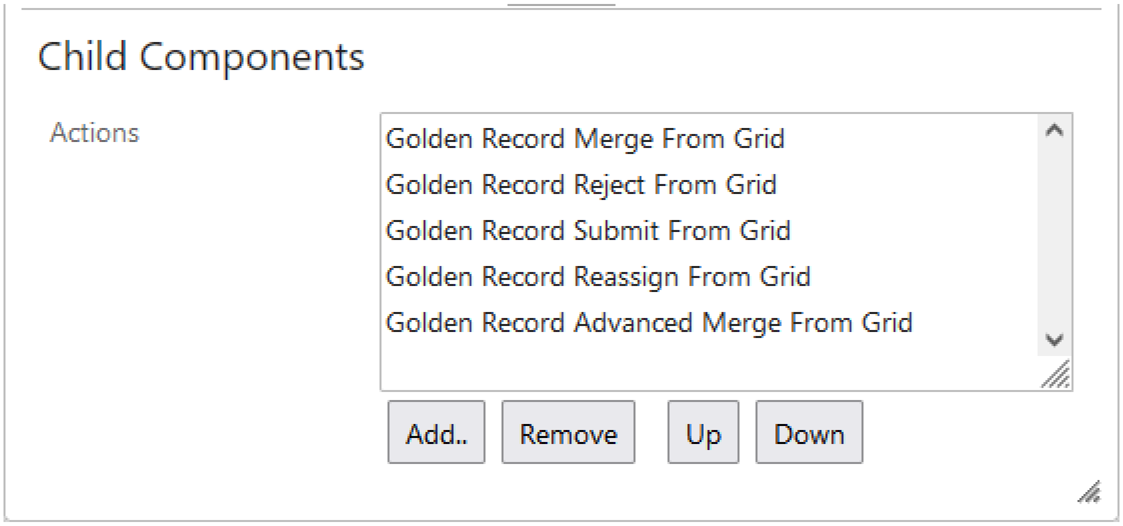
   - High Priority - Select to display priority flags.

   - Match Recommendation - Select to display the merge / reject recommendations given via **Machine Learning Match Recommendations**.

- Task Summary Column Width - Specify the width of the Task column.
- Created Date - Select to display the date and time of creation.

Note: Users can filter based on ID, match score, assignee, match recommendation, and created date. For more information, refer to the **Filtering Task List** section below.

8. For the **Child Components** section, click **Add...** and select the actions to add to the toolbar. The Submit, Reject, and Reassign buttons assigned in this step are inherited to the Advanced Merge dialog.



Child Components

Actions

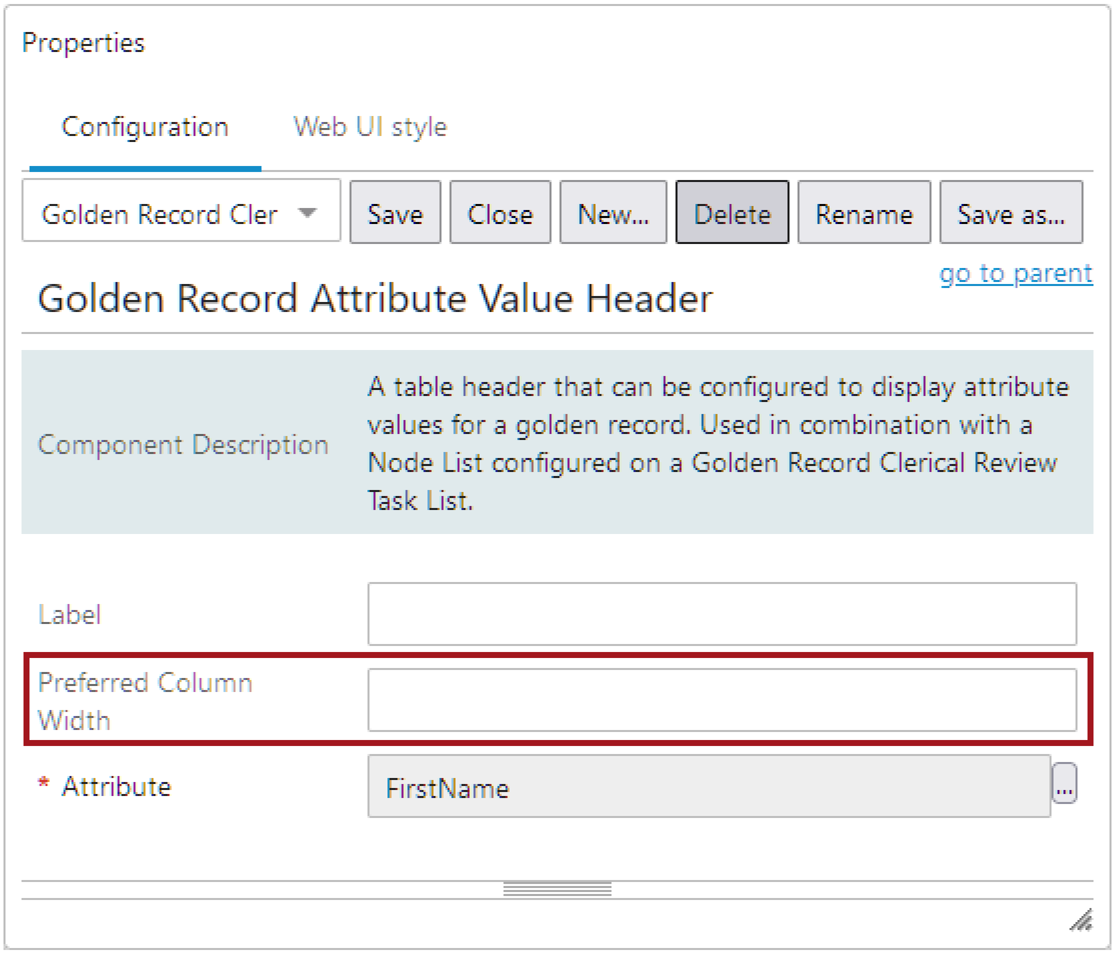
Golden Record Merge From Grid
Golden Record Reject From Grid
Golden Record Submit From Grid
Golden Record Reassign From Grid
Golden Record Advanced Merge From Grid

Note: Once an action is added, double click it to complete the required configuration.

9. Click **Save**.
10. Assign the Golden Record Clerical Review Task List to a Status Selector Homepage widget or a workflow Status Selector child component on the Global Navigation Panel. For more information on configuring and using this widget, refer to the **Status Selector Homepage Widget** topic of the **Web User Interfaces** documentation.

Column width configuration

Users can configure the columns on a Golden Record Clerical Review Task List to have their own default widths which will apply each time the page loads. Most columns widths are configurable by selecting the component in Design Mode and filling out the 'Preferred Column Width' component.



Golden Record Attribute Value Header

Component Description: A table header that can be configured to display attribute values for a golden record. Used in combination with a Node List configured on a Golden Record Clerical Review Task List.

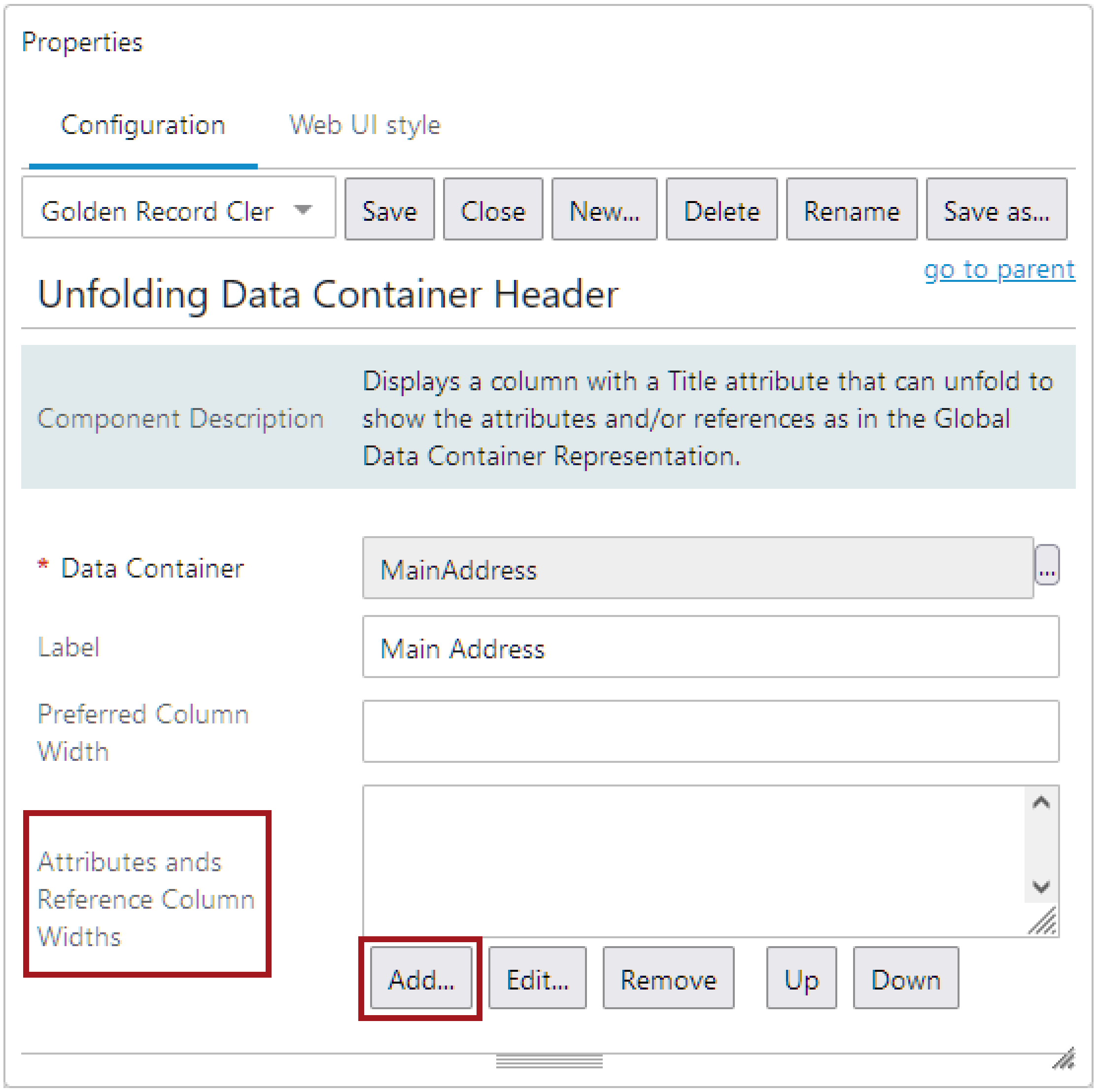
Label:

Preferred Column Width:

* Attribute: FirstName

However, Unfoldable Data Container columns require additional configuration, as they can contain several attribute and reference columns. To configure the individual attribute and reference columns, use the following steps:

1. In the Design Mode, select the Unfoldable Data Container component that you wish to edit.
2. Navigate to the 'Attributes and Reference Column Widths' parameter and click **Add**.



Properties

Unfolding Data Container Header

Component Description: Displays a column with a Title attribute that can unfold to show the attributes and/or references as in the Global Data Container Representation.

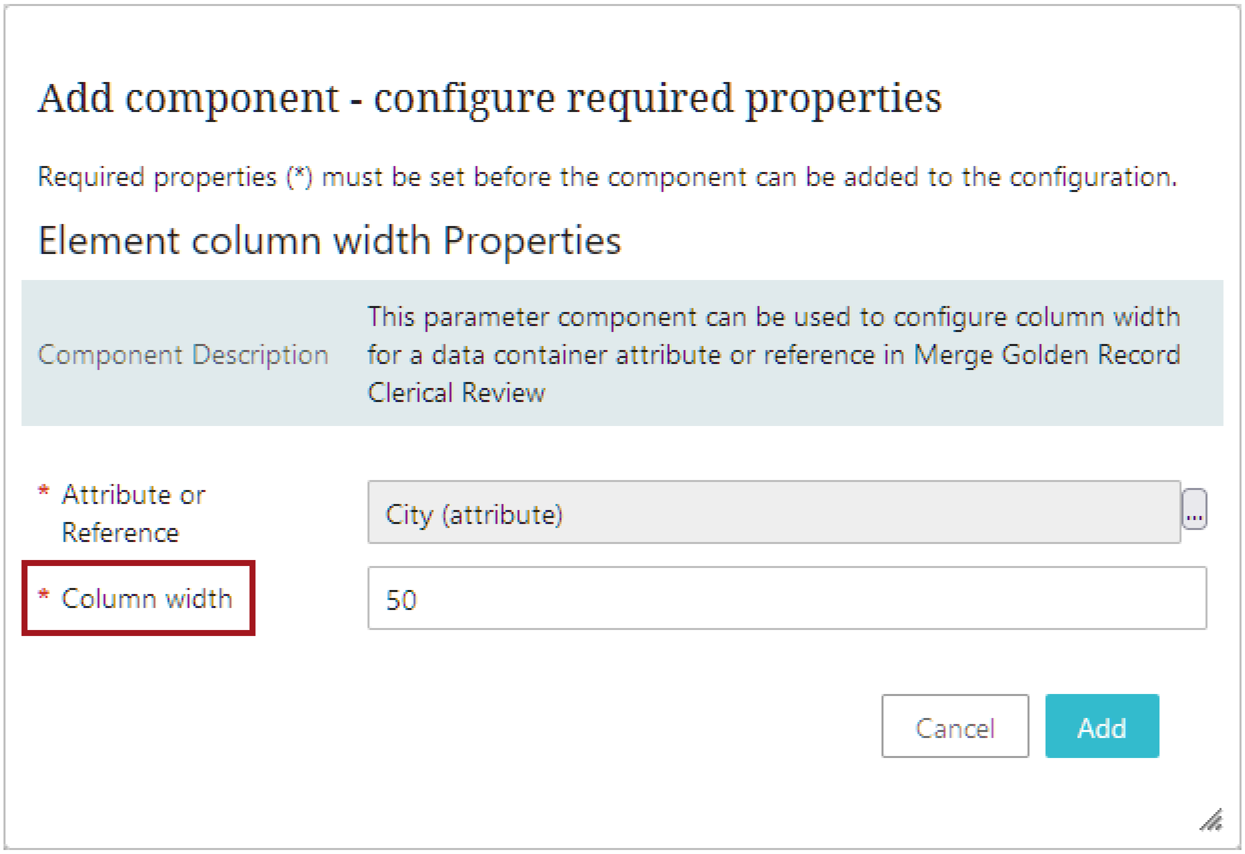
* Data Container: MainAddress

Label: Main Address

Preferred Column Width:

Attributes ands Reference Column Widths:

3. Select an attribute or reference used in that data container, specify the column width, and click **Add**.



Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Element column width Properties

Component Description	This parameter component can be used to configure column width for a data container attribute or reference in Merge Golden Record Clerical Review
* Attribute or Reference	City (attribute)
* Column width	50

4. Repeat steps 2 and 3 for each attribute or reference column within the data container.

5. Save and close Design Mode.

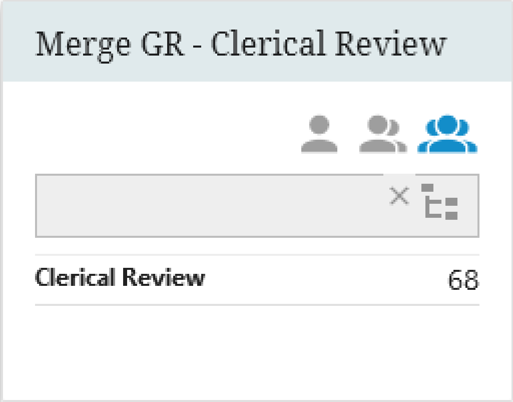
Using Action Buttons

The Golden Record Clerical Review Task List allows a user to evaluate the potential duplicate tasks assigned to them and perform actions on them, including rejecting or merging the records.

> **Note:** If 10 or more tasks are selected, or when using the **Select all** button, all actions run as bulk updates in a background process (BGP). Keep in mind that because it is running in the background, the Task List does not automatically update, and will not update until the BGP is done. For Merge, Reject, and Submit actions, this means that the tasks will not be removed from the view until the BGP is finished and the task list is manually refreshed.

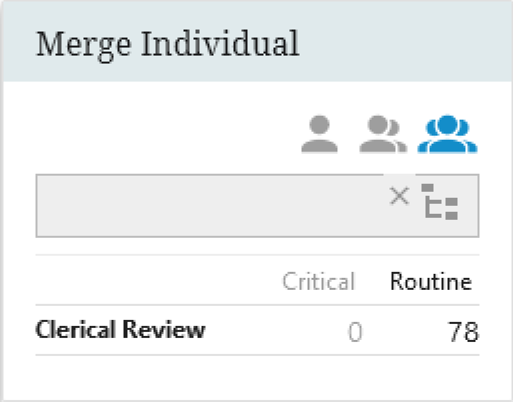
Follow these steps to manage the task list.

1. In Web UI, navigate to Golden Record Clerical Review Task List via a relevant Status Selector Homepage widget or workflow Status Selector. In these images, the Merge Individual status selector is configured for a 'Critical' high priority status flag.



Merge GR - Clerical Review

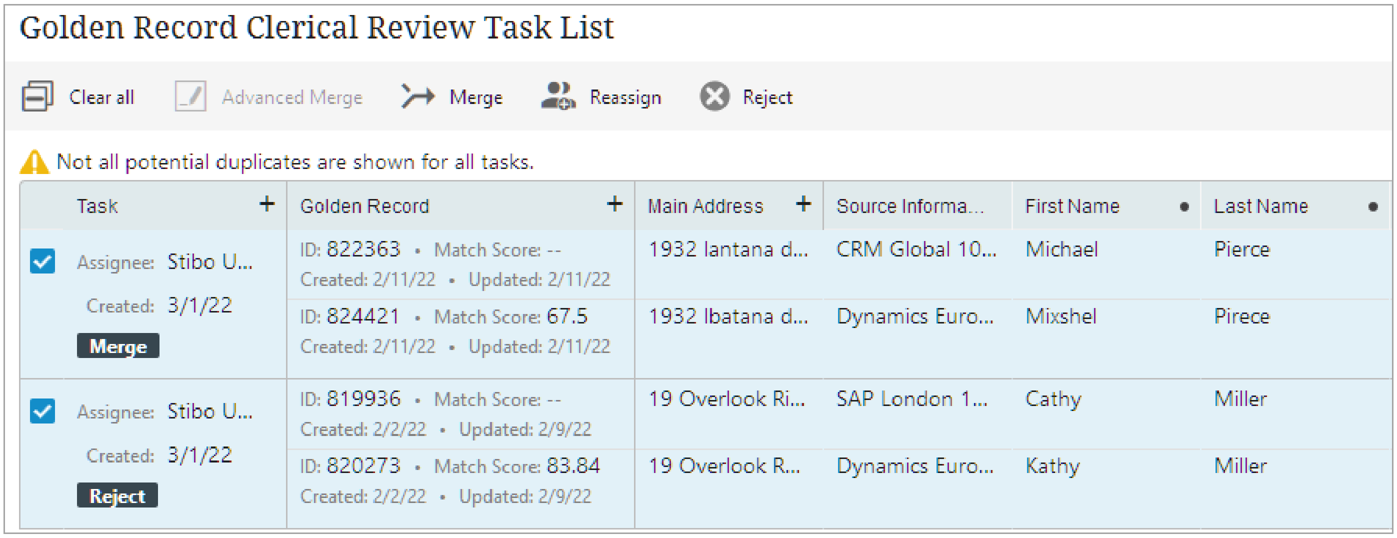
Clerical Review	68



Merge Individual

	Critical	Routine
Clerical Review	0	78

2. Select the desired tasks using one of the following:



Golden Record Clerical Review Task List

⚠ Not all potential duplicates are shown for all tasks.

Task	Golden Record	Main Address	Source Informa...	First Name	Last Name
☑ Assignee: Stibo U... Created: 3/1/22 Merge	ID: 822363 • Match Score: -- Created: 2/11/22 • Updated: 2/11/22	1932 lantana d...	CRM Global 10...	Michael	Pierce
	ID: 824421 • Match Score: 67.5 Created: 2/11/22 • Updated: 2/11/22	1932 lbatana d...	Dynamics Euro...	Mixshel	Pirece
☑ Assignee: Stibo U... Created: 3/1/22 Reject	ID: 819936 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	19 Overlook Ri...	SAP London 1...	Cathy	Miller
	ID: 820273 • Match Score: 83.84 Created: 2/2/22 • Updated: 2/9/22	19 Overlook R...	Dynamics Euro...	Kathy	Miller

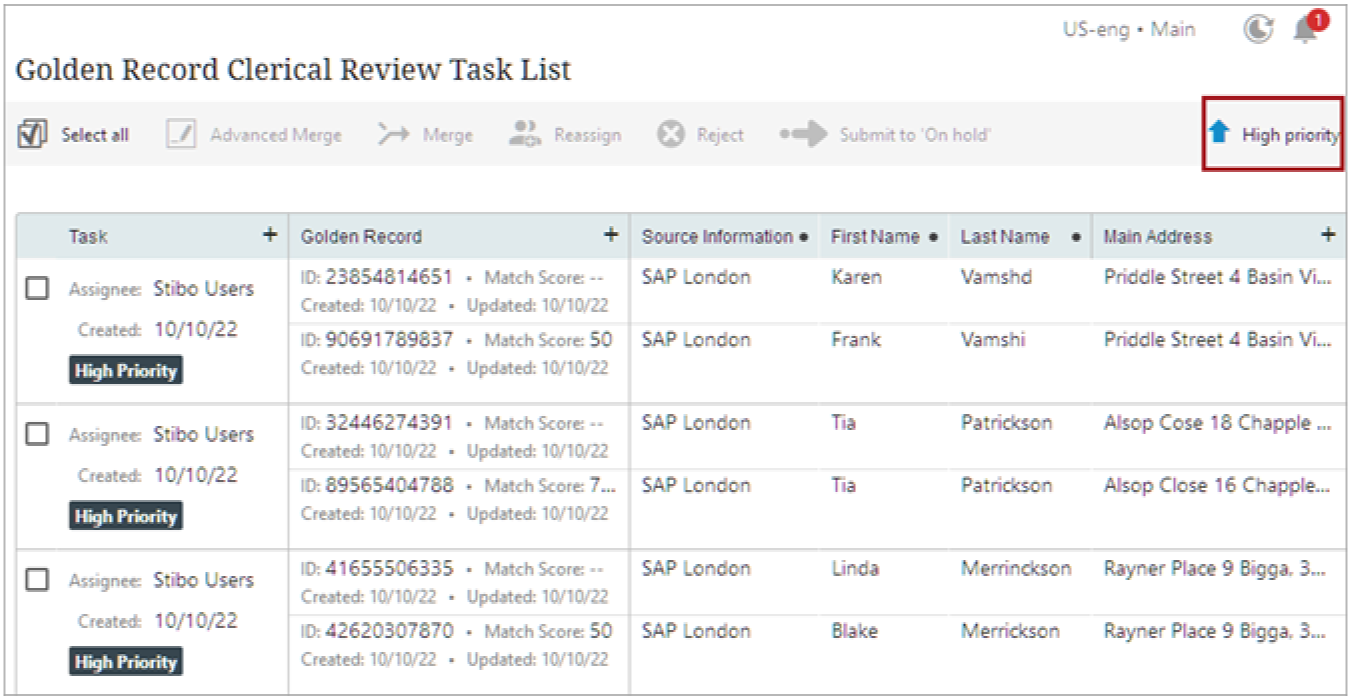
- Select an individual task via the checkbox. If multiple tasks are selected, all actions except the **Advanced merge** action are available.
- Click **Select all** to select every task; if you have applied a filter, this will only select the tasks that apply to that filter.

  > **Note:** The potential duplicate reference type must be configured in the component model and all tasks must be republished for the 'Select all' button to be enabled.

- Click **Clear all** to remove all selections on the screen.
- If priority flags are configured, select the filter action to show only high priority tasks. The high priority filter does not recognize the default workflow flag as defined on the workflow. Tasks without high priority flag are not handled as having the workflow default. For information on displaying this filter, refer to the **High Priority Status Flag Global Representation Type** section of the **Main Properties** topic of the **Web**

User Interfaces documentation.

Important: If only one flag is configured, STEP considers it the default and assigns it to all items in a workflow. It does not display in a Web UI, and you are unable to filter by that flag. To circumvent this, configure a default flag in workbench in addition to the specified flag. For more information, refer to the **Configuring Status Flags in Workflows** topic in the **Workflows** documentation.



Golden Record Clerical Review Task List

Task	Golden Record	Source Information	First Name	Last Name	Main Address
☐ Assignee: Stibo Users Created: 10/10/22 High Priority	ID: 23854814651 • Match Score: -- Created: 10/10/22 • Updated: 10/10/22	SAP London	Karen	Vamshd	Priddle Street 4 Basin Vi...
	ID: 90691789837 • Match Score: 50 Created: 10/10/22 • Updated: 10/10/22	SAP London	Frank	Vamshi	Priddle Street 4 Basin Vi...
☐ Assignee: Stibo Users Created: 10/10/22 High Priority	ID: 32446274391 • Match Score: -- Created: 10/10/22 • Updated: 10/10/22	SAP London	Tia	Patrickson	Alsop Cose 18 Chapple ...
	ID: 89565404788 • Match Score: 7... Created: 10/10/22 • Updated: 10/10/22	SAP London	Tia	Patrickson	Alsop Close 16 Chapple...
☐ Assignee: Stibo Users Created: 10/10/22 High Priority	ID: 41655506335 • Match Score: -- Created: 10/10/22 • Updated: 10/10/22	SAP London	Linda	Merrinckson	Rayner Place 9 Bigga, 3...
	ID: 42620307870 • Match Score: 50 Created: 10/10/22 • Updated: 10/10/22	SAP London	Blake	Merrickson	Rayner Place 9 Bigga, 3...

Note: The filter icon only displays when High Priority tasks exist for the current user based on assignees.

For information on configuring a status flag, refer to the **Creating a Merge Golden Record Clerical Review Workflow** topic.

3. Click an action button to perform the action on the golden records of the selected task(s):
   - **Reject** - marks as 'Confirmed Non Duplicates' and removes from the workflow. For information on the 'Confirmed Non Duplicates' reference type, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.
   - **Reassign** - assigns to another user or group from the 'Select assignee' dialog.
   - **Submit** - moves to another state in the workflow. Requires that the case sensitive 'Event ID' parameter is configured.

Note: This case sensitive Event ID must correspond with the ID of the relevant event on a transition in the workflow.

- ○ **Merge** - merges each task selected into a single surviving golden record. The values that get promoted to the surviving golden record are determined by the survivorship rules specified on the corresponding matching algorithm. The non-surviving golden records in the task are set to deactivated and will not be matched again.

  For information on survivorship rules, refer to the **Golden Records Survivorship Rules** topic.

  For information on deactivation, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.

- ○ **Advanced Merge** - allows manual inspection and after viewing the relevant details of each golden record and performing actions such as: rejecting specific golden records as duplicates, reassigning the task to another user, submitting the task to another state in the workflow, or merging the selected records. Values on Advanced Merge cannot be copied and pasted.

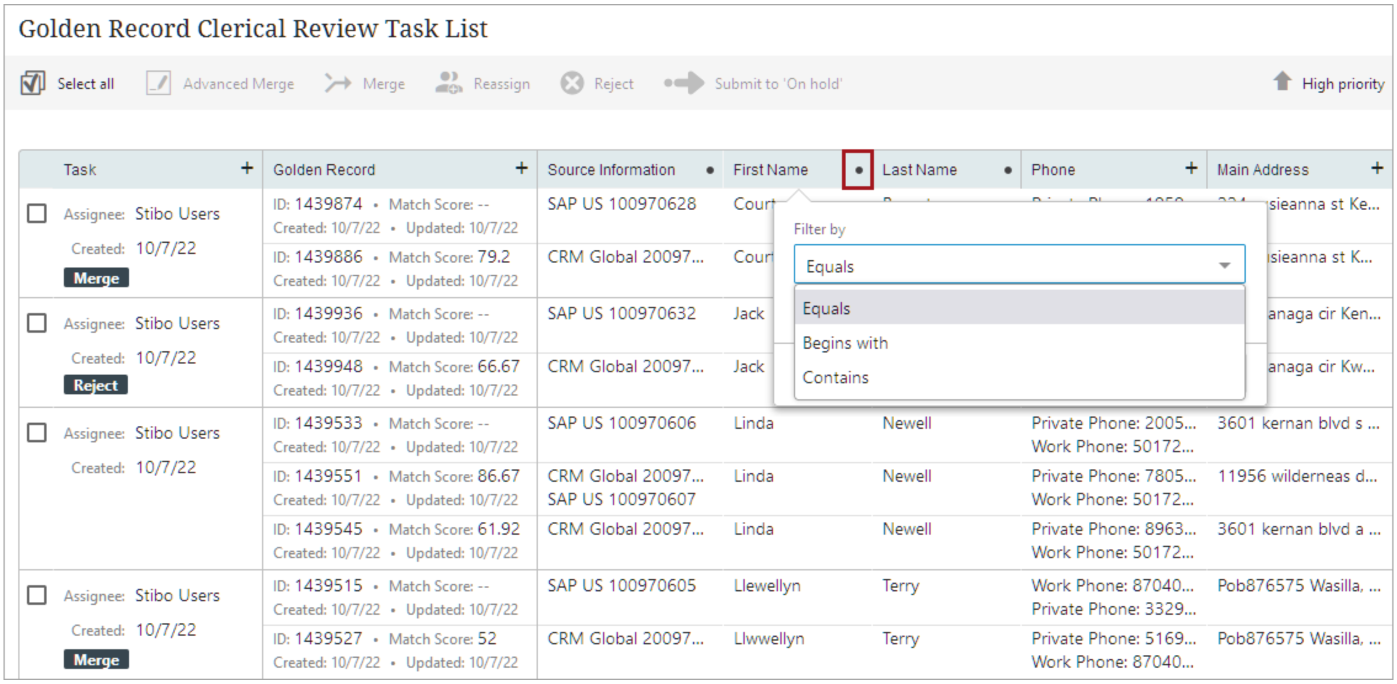
  For more information on configuring and using the advanced merge dialog, refer to the **Golden Record Advanced Merge Dialog** topic.

Filtering Task List

The Golden Record Clerical Review Task List allows users to filter on any column with a circle (●) button.

Follow these steps to filter a task list:

1. Click the circle (●) button next to a filterable column. If the desired selection is within an unfoldable column or data container, unfold them by clicking the plus (+) button and choose the column. After clicking, a dialog appears.



Golden Record Clerical Review Task List

Task +	Golden Record +	Source Information ●	First Name ●	Last Name ●	Phone +	Main Address +
□ Assignee: Stibo Users Created: 10/7/22 Merge	ID: 1439874 • Match Score: -- Created: 10/7/22 • Updated: 10/7/22	SAP US 100970628	Court			sieanna st Ke...
	ID: 1439886 • Match Score: 79.2 Created: 10/7/22 • Updated: 10/7/22	CRM Global 20097...	Cour			sieanna st K...
□ Assignee: Stibo Users Created: 10/7/22 Reject	ID: 1439936 • Match Score: -- Created: 10/7/22 • Updated: 10/7/22	SAP US 100970632	Jack			anaga cir Ken...
	ID: 1439948 • Match Score: 66.67 Created: 10/7/22 • Updated: 10/7/22	CRM Global 20097...	Jack			anaga cir Kw...
□ Assignee: Stibo Users Created: 10/7/22	ID: 1439533 • Match Score: -- Created: 10/7/22 • Updated: 10/7/22	SAP US 100970606	Linda	Newell	Private Phone: 2005... Work Phone: 50172...	3601 kernan blvd s ...
	ID: 1439551 • Match Score: 86.67 Created: 10/7/22 • Updated: 10/7/22	CRM Global 20097... SAP US 100970607	Linda	Newell	Private Phone: 7805... Work Phone: 50172...	11956 wilderneas d...
	ID: 1439545 • Match Score: 61.92 Created: 10/7/22 • Updated: 10/7/22	CRM Global 20097...	Linda	Newell	Private Phone: 8963... Work Phone: 50172...	3601 kernan blvd a ...
□ Assignee: Stibo Users Created: 10/7/22 Merge	ID: 1439515 • Match Score: -- Created: 10/7/22 • Updated: 10/7/22	SAP US 100970605	Llewellyn	Terry	Work Phone: 87040... Private Phone: 3329...	Pob876575 Wasilla, ...
	ID: 1439527 • Match Score: 52 Created: 10/7/22 • Updated: 10/7/22	CRM Global 20097...	Llwwellyn	Terry	Private Phone: 5169... Work Phone: 87040...	Pob876575 Wasilla, ...

2. Select the parameter to filter by.
3. Enter a value or text to filter by. Click **Apply filter**.
4. To clear a filter, click the filter (▼) button and click **Clear filter**.

MLMR Support Guidelines

The quality of recommendations provided by Machine Learning Match Recommendations (MLMR) in the Clerical Review Task List are dependent on the 'merge' and 'reject' decisions made by the user charged with training the matching agent. To improve the quality of the recommendations, Stibo Systems provides customers with a dedicated team prepared to engage in a collaborative process with customers to help improve the customer's understanding of the recommendations, and to improve the quality of those recommendations.

If the matching agent recommendations you receive results in questions for you or your team, in the Support Portal, create a ticket with the Issue Type 'Customer Request.' Find below a list of the fields required when creating a support issue for the MLMR and descriptions of how to provide the requested content.

> **Note:** Before you create a support issue, verify the relevant data to match on is mapped to the matching agent.

Summary: Add a short description of the issue you are experiencing in this field. Preface your summary content with ‘MLMR' so it is clear to the support team that the issue relates to the matching agent recommendations. The format will look like this: 'MLMR - <description of the issue>’.

Description: In this field, copy the data points listed below and paste it into the 'Description' field in the issue. Then add the requested information for each data point:

Description of problem:

Total number of recommendations:

Number of wrong recommendations identified:

Description of wrong recommendations: Describe what is wrong with the recommendations from the matching agent.

System Name / URL:

Training Process BGP ID:

Training Process BGP Started timestamp:

Training Process Execution Report:

<paste text>

Recommendation Process BGP ID:

Recommendation Process BGP Started timestamp:

Recommendation Process Execution Report:

<paste text>

Issue Category: ML Matching Agent

Business Domain: CMDM

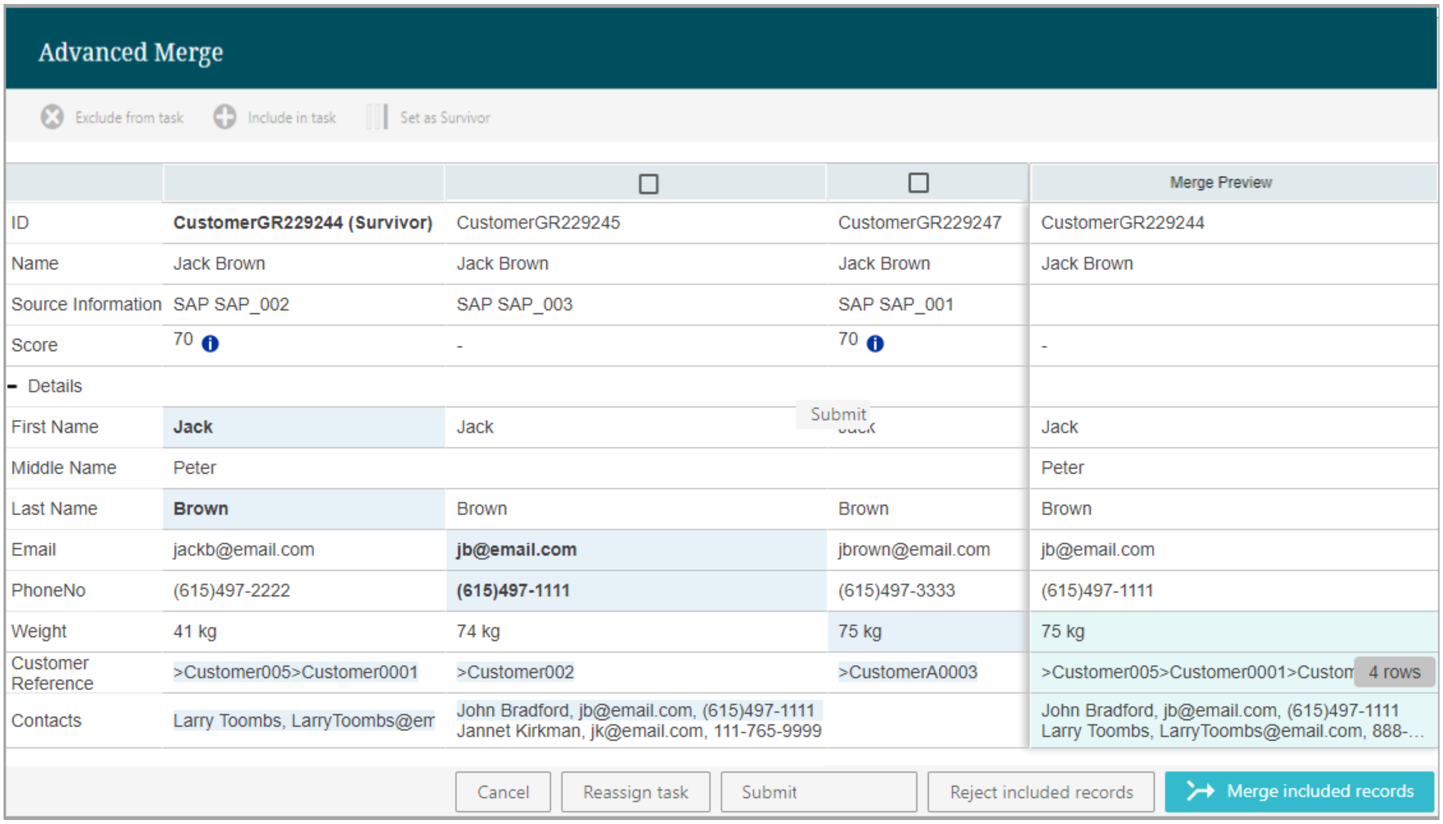
Golden Record Advanced Merge Dialog

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

The Golden Record Advanced Merge dialog is accessed from the Golden Record Clerical Review Task List screen. Advanced Merge allows users to view the relevant details of each golden record in the selected task and perform a number of actions such as: rejecting specific golden records as duplicates, reassigning the task to another user, submitting the task to the next state in the workflow, or merging the selected records. The values available are determined by the value headers configured. For information on the Golden Record Clerical Review Task List screen, refer to the **Configuring the Golden Record Clerical Review Task List** topic.

Important: Advanced Merge only merges the first 20 records in a task. Once those 20 are merged, all other potential duplicate records are removed from the task list without being merged or rejected. When the merge event processor runs next, it identifies the remaining potential duplicates and re-enters them into the Golden Record Clerical Review Task List as a new task. For more information on the merge event processor, refer to the **Configuring the Merge Event Processor** topic.

This topic includes details on **Configuration**, **Header Configuration**, and **Using the View and Buttons**.



		□	□	Merge Preview
ID	**CustomerGR229244 (Survivor)**	CustomerGR229245	CustomerGR229247	CustomerGR229244
Name	Jack Brown	Jack Brown	Jack Brown	Jack Brown
Source Information	SAP SAP_002	SAP SAP_003	SAP SAP_001	
Score	70 ⓘ	-	70 ⓘ	-
- Details				
First Name	**Jack**	Jack	Submit Jack	Jack
Middle Name	Peter			Peter
Last Name	**Brown**	Brown	Brown	Brown
Email	jackb@email.com	**jb@email.com**	jbrown@email.com	jb@email.com
PhoneNo	(615)497-2222	**(615)497-1111**	(615)497-3333	(615)497-1111
Weight	41 kg	74 kg	75 kg	75 kg
Customer Reference	>Customer005>Customer0001	>Customer002	>CustomerA0003	>Customer005>Customer0001>Custom 4 rows
Contacts	Larry Toombs, LarryToombs@em	John Bradford, jb@email.com, (615)497-1111 Jannet Kirkman, jk@email.com, 111-765-9999		John Bradford, jb@email.com, (615)497-1111 Larry Toombs, LarryToombs@email.com, 888-...

Note: The **Reassign task, Submit,** and **Reject included records** action buttons in the Advanced Merge dialog are inherited from the parent screen configuration, Golden Record Clerical Review Task List.

Prerequisites

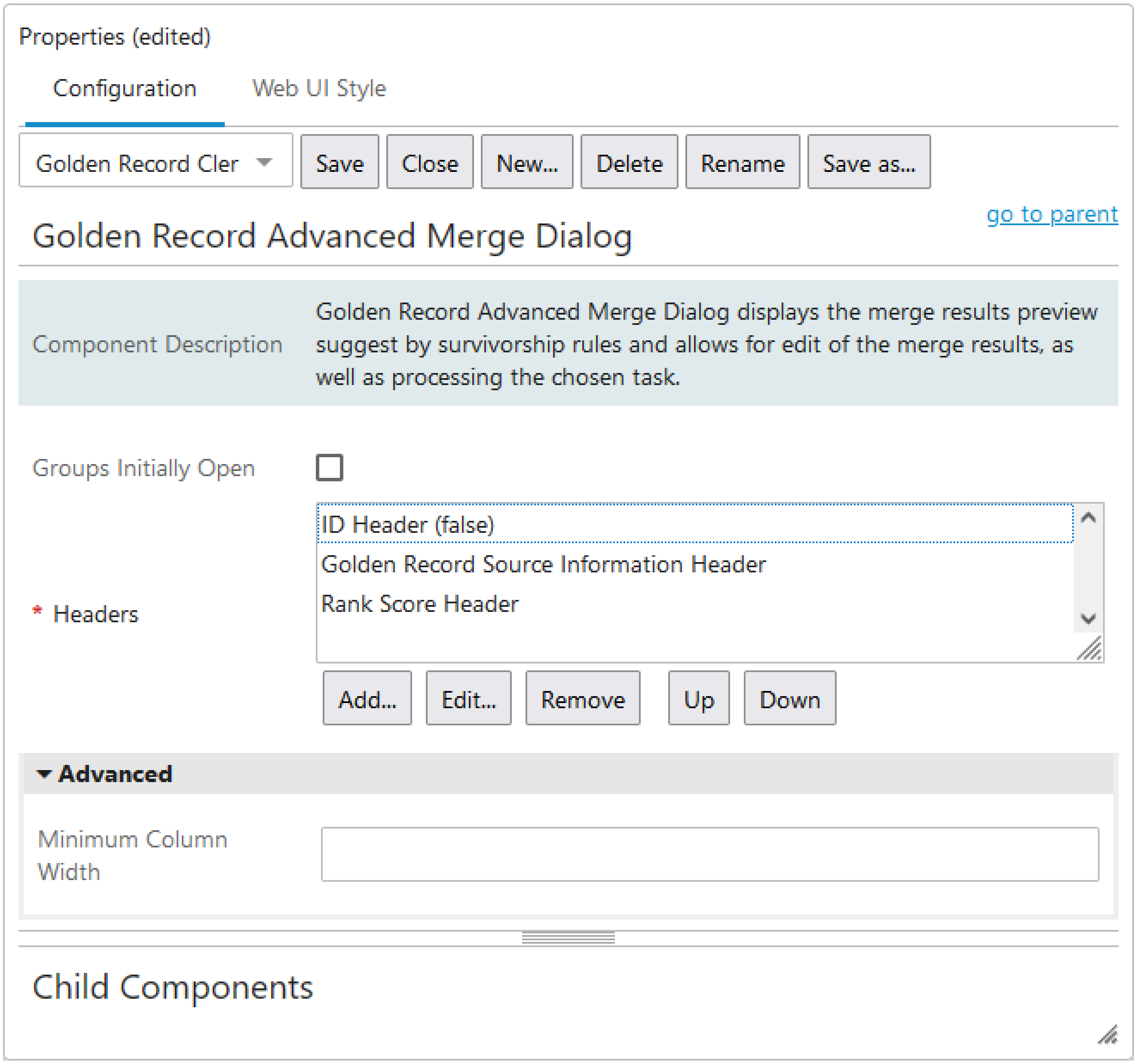
It is expected that anyone configuring the Golden Record Advanced Merge Dialog is familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** topic of the **Web User Interfaces** documentation.

Configuration

Use the following steps to add and configure the screen.

Important: For maximum efficiency in reviewing content, configuring the Dynamic Table Layout component is required. To optimize the view for object comparisons, when configuring this component uncheck the 'Allow Wrap Of Header Title' and 'Allow Wrap of Cell Content' parameters. This applies to both the Golden Record Clerical Review Task List and Advanced Merge Dialog. For more information, refer to the **Dynamic Table Layout** topic of the **Web User Interfaces** documentation.

1. In the designer, open a relevant Golden Record Clerical Review Task List screen.
2. Under the Child Components section, add a **Golden Record Advanced Merge From Grid** action.
3. Double click the Golden Record Advanced Merge From Grid action to display the configuration of the child component.
4. For the Component parameter, select **Golden Record Advanced Merge Dialog** and click the 'go to component' link.



Properties (edited)

Golden Record Cler

Golden Record Advanced Merge Dialog

Component Description: Golden Record Advanced Merge Dialog displays the merge results preview suggest by survivorship rules and allows for edit of the merge results, as well as processing the chosen task.

Groups Initially Open ☐

* Headers:
ID Header (false)
Golden Record Source Information Header
Rank Score Header

▾ **Advanced**

Minimum Column Width

Child Components

5. For the **Groups Initially Open** parameter, when checked, attribute groups are already open when the dialog displays.

6. For the **Headers** parameter (mandatory), click **Add...** and select a header to add to the dialog, and click the **Add** button on the Add Component dialog. An additional configuration dialog is displayed if required. For information on configuring the recommended headers, refer to the **Headers** section below.

   Recommended headers for this dialog are: Name, Attribute Value, Attribute Value Group, and Advanced Merge Reference Header.

   Additionally, by default, the following headers are pre-configured on newly created Advanced Merge Dialog components: ID, Golden Record Source Information, and Rank Score.

> **Note:** When available, set the 'Enable Links' parameter to 'false' since it is ignored if set to 'true'. Read-only parameters are also ignored.

7. For the **Minimum Column Width** parameter, enter the preferred width of the columns in pixels.

Header Configuration

The following section includes details about configuring the headers available on the Golden Record Advanced Merge Dialog.

ID Header

The 'ID Header' displays the IDs of objects in the table. Configuration options include:

- Dimensions - Specify height / width of the of the header cell.
- Enable Link - Check the box if the values under this header should act as links.
- Label - Enter a label for the header.
- Table Sorting - Select a method of sorting the values in the header.
- Context Help - Enter help text for the component to display. Only works if the 'Enable Link' box is checked.

ID values cannot be edited in the corresponding Merge Preview field.

Name Header

The 'Name Header' displays the names of objects in the table. Configuration options include:

- Dimensions - Specify height / width of the of the header cell.
- Label - Enter a label for the header.
- Table Sorting - Select a method of sorting the values in the header.

Name values can be edited in the corresponding Merge Preview field.

Golden Record Source Information Header

The 'Golden Record Source Information Header' displays source system and record information for objects in the table. Configuration options include:

- Label - Enter a label for the header.
- Preferred Column Width - Enter the width of the column in pixels.

Golden Record Source Information values cannot be edited in the corresponding Merge Preview field.

Rank Score Header

The 'Rank Score Header' displays the 'match score' of golden records as they compare to the surviving golden record. Configuration options include:

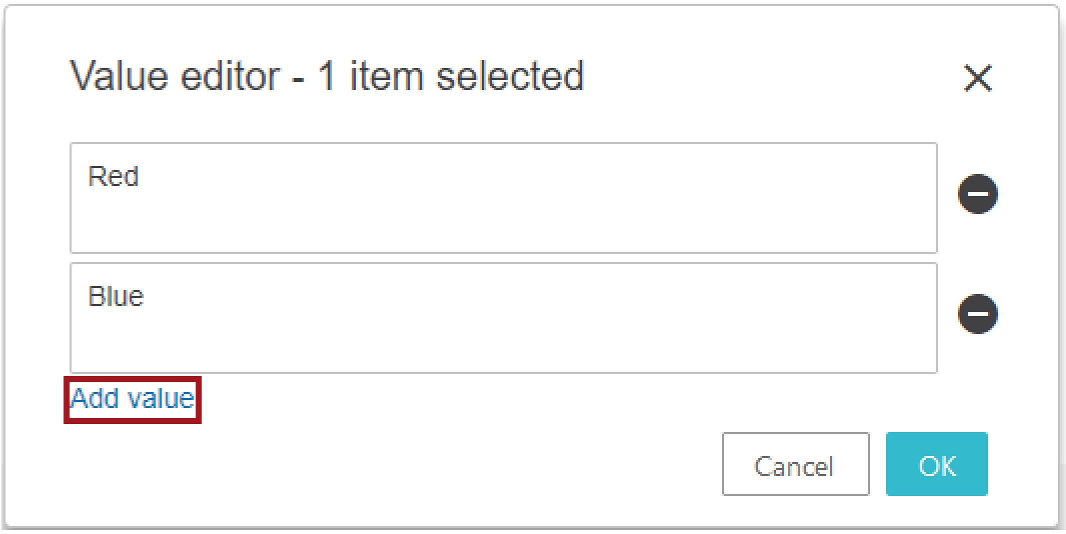
  - Dimensions - Specify height / width of the of the header cell.
  - Label - Enter a label for the header.
  - Table Sorting - Select a method of sorting the values in the header.

Attribute Value Header

The 'Attribute Value Header' displays the attribute values of objects in the table. Configuration options include:

  - Attribute - Click the ellipsis button (...) and browse or search for the relevant attribute to display values for.
  - Dimensions - Specify height / width of the of the header cell.
  - Label - Enter a label for the header.
  - Mandatory - Specify whether or not an attribute value are considered mandatory. If so, the header will appear italicized and with an asterisk (*) in the table.
  - Read Only - When checked, values under this header cannot be edited.
  - Table Sorting - Select a method of sorting the values in the header.
  - Enable Locale Formatting - When checked, 'ISO Date' and 'ISO Date and Time' values are formatted according to locale.
  - Show Invalid Inherited Values - When checked, the table displays inherited values even if the attribute is not valid for the object.
  - Show LOV IDs - When checked, relevant LOVs display their IDs next to the corresponding values.
  - No Wrap - When checked, values do not wrap within the cells. This setting is overridden by the Dynamic Table Layout Settings. For more information, refer to the **Main Properties** topic in the **Web User Interfaces** documentation.
  - Context Help - Enter help text to display.
  - Display Context help - When checked, display context help text for attributes.

- Attribute values can be maintained in the corresponding Merge Preview field.
- For a multi-valued attribute, double click the corresponding Merge Preview cell and click the **Add value** link in the 'Value editor' dialog.



Attribute Value Group Header

The 'Attribute Value Group Header' displays a group of attributes' values for objects in the table. Configuration options include:

- Attribute Group - Click the ellipsis button (...) and browse or search for the relevant attribute group to display values for.
- Blacklist Attribute Group - Specify which attribute groups **not** to display attributes, even if they also appear in the attribute group specified above.
- Dimensions - Specify height / width of the header cell.
- Included Nested Groups - When checked, attributes from nested parameter groups are included.
- Label - Specify a label for the header.
- Mandatory - Specify whether or not an attribute value are considered mandatory. If so, the header will appear italicized and with an asterisk (*) in the table.
- Read Only - When checked, values under this header cannot be edited.
- Show LOV IDs - When checked, relevant LOVs display their IDs next to the corresponding values.
- Table Sorting - Select a method of sorting the values in the header.
- Enable Locale Formatting - When checked, 'ISO Date' and 'ISO Date and Time' values are formatted according to locale.
- No Wrap - When checked, values do not wrap within the cells. This setting is overridden by the Dynamic Table Layout Settings. For more information, refer to the **Main Properties** topic in the **Web User Interfaces** documentation.

Note: This parameter is automatically set upon saving the component as the Golden Record Advanced Merge is designed to compare data, and thus should not wrap text.

  - Context Help - Enter help text for the component to display.
  - Display Context help - When checked, display context help text for attributes.

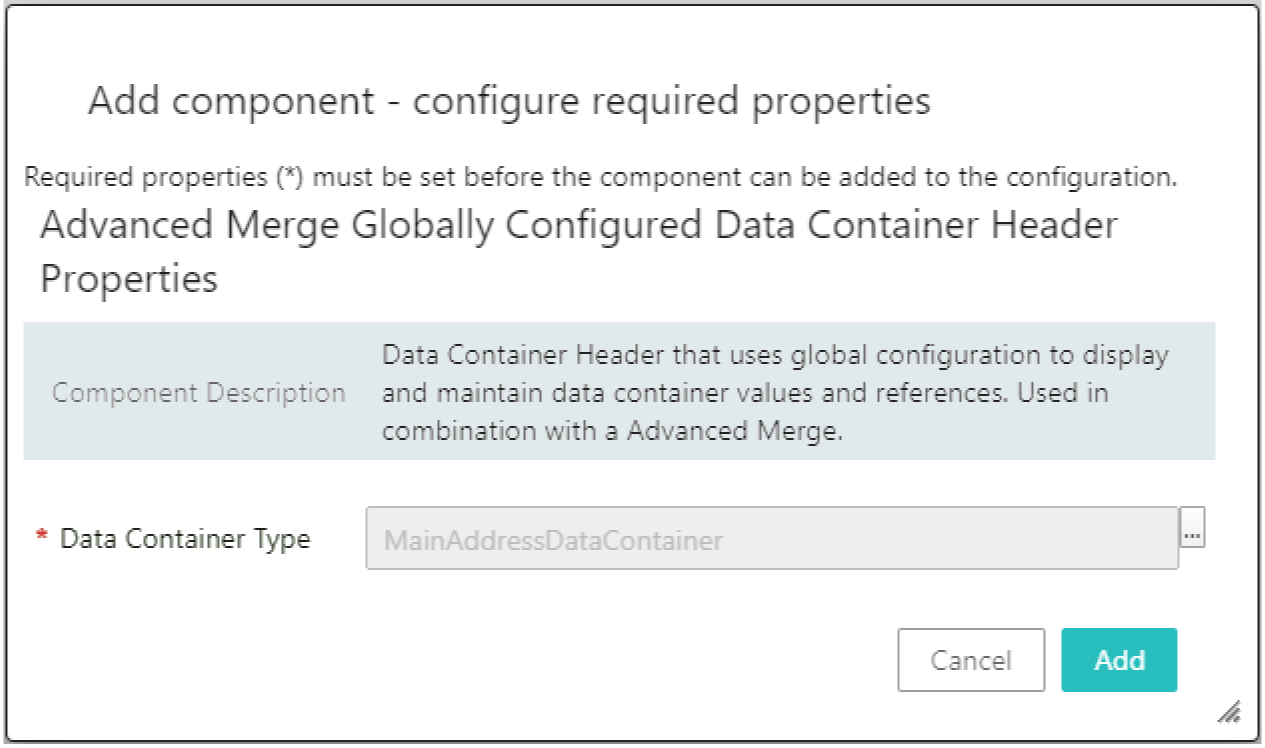
- Attribute values can be maintained in the corresponding Merge Preview field.
- For a multi-valued attribute, double click the corresponding Merge Preview cell and click the **Add value** link in the 'Value editor' dialog.

Advanced Merge Globally Configured Data Container Header

The 'Advanced Merge Globally Configured Data Container Header' displays data container values.

Note: Avoid using the Data Container Attribute Value Group Header and the Data Container Attribute Value Header.

Once added, specify the data container type to display data. Under the **Data Container Type** parameter, click the ellipsis button (**...**) and browse or search for the relevant data container type to display attributes and references from.



Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

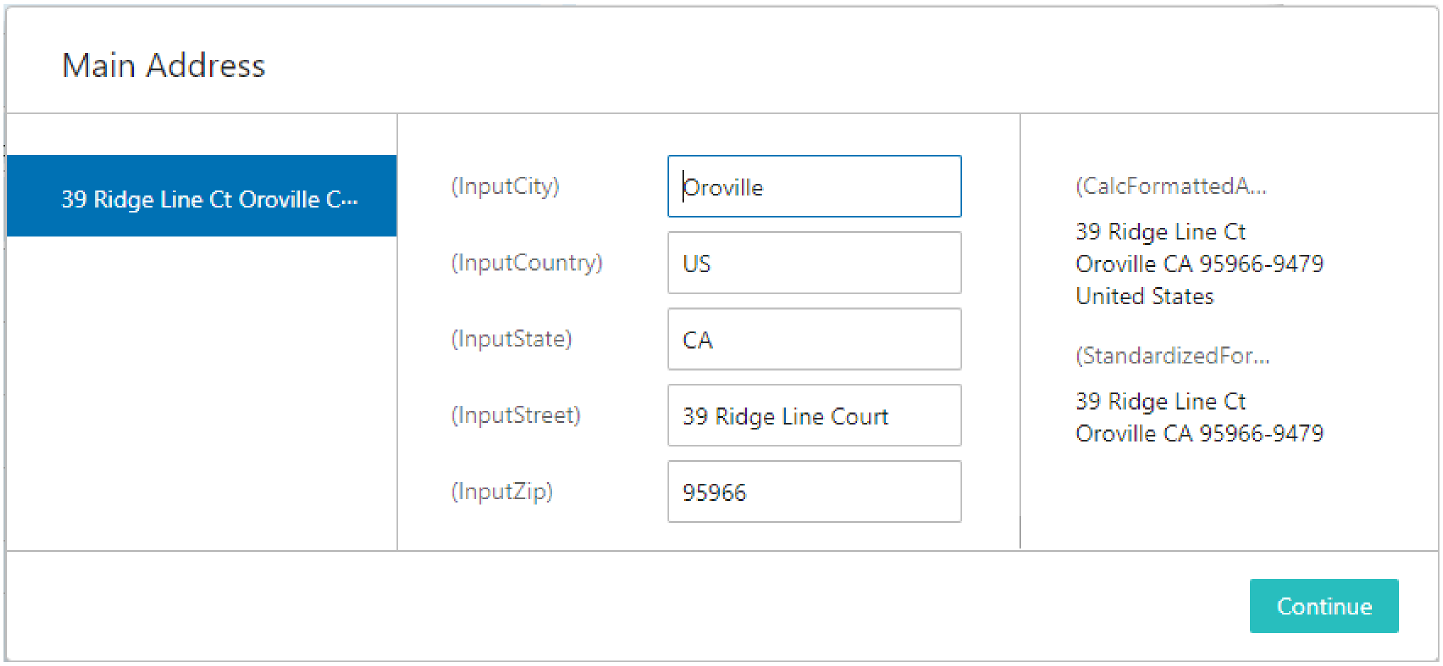
Advanced Merge Globally Configured Data Container Header Properties

Component Description	Data Container Header that uses global configuration to display and maintain data container values and references. Used in combination with a Advanced Merge.
* Data Container Type	MainAddressDataContainer

To configure which attribute values / references display in this component and under what conditions they are displayed, refer to the **Global Data Container Representations** topic of the **Web User Interfaces** documentation.

Data containers can be manually selected from various golden records to build a complete set of Data Containers in the Merge Preview.

- To add / maintain data containers from the Advanced Merge dialog, double click the corresponding cell under Merge Preview.
- The editor dialog has the same functionality as the Globally Configured Data Container component. For more information on creating / maintaining data containers using the Globally Configured Data Container component, refer to the **Data Containers in Web UI** topic of the **Web User Interfaces** documentation.



Main Address

39 Ridge Line Ct Oroville C...

(InputCity) Oroville

(InputCountry) US

(InputState) CA

(InputStreet) 39 Ridge Line Court

(InputZip) 95966

(CalcFormattedA...

39 Ridge Line Ct
Oroville CA 95966-9479
United States

(StandardizedFor...

39 Ridge Line Ct
Oroville CA 95966-9479

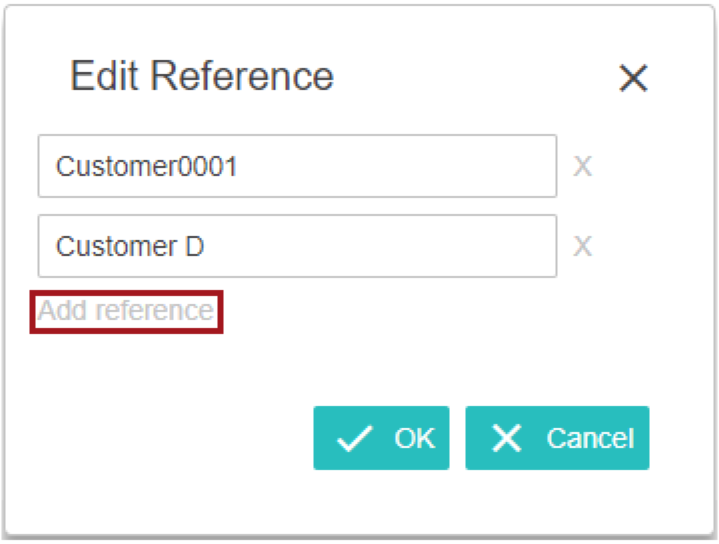
Advanced Merge Reference Header

The 'Advanced Merge Reference Header' displays the references to target objects in the table. Configuration options include:

- Label - Specify a label for the header.
- Reference Type - Click the ellipsis button (...) and browse or search for the relevant reference type to display.

References to target objects can be edited and manually selected from various golden records to build a complete set of references to target objects in the Merge Preview. If the reference type is multi-valued, multiple references can be added via the Merge Preview.

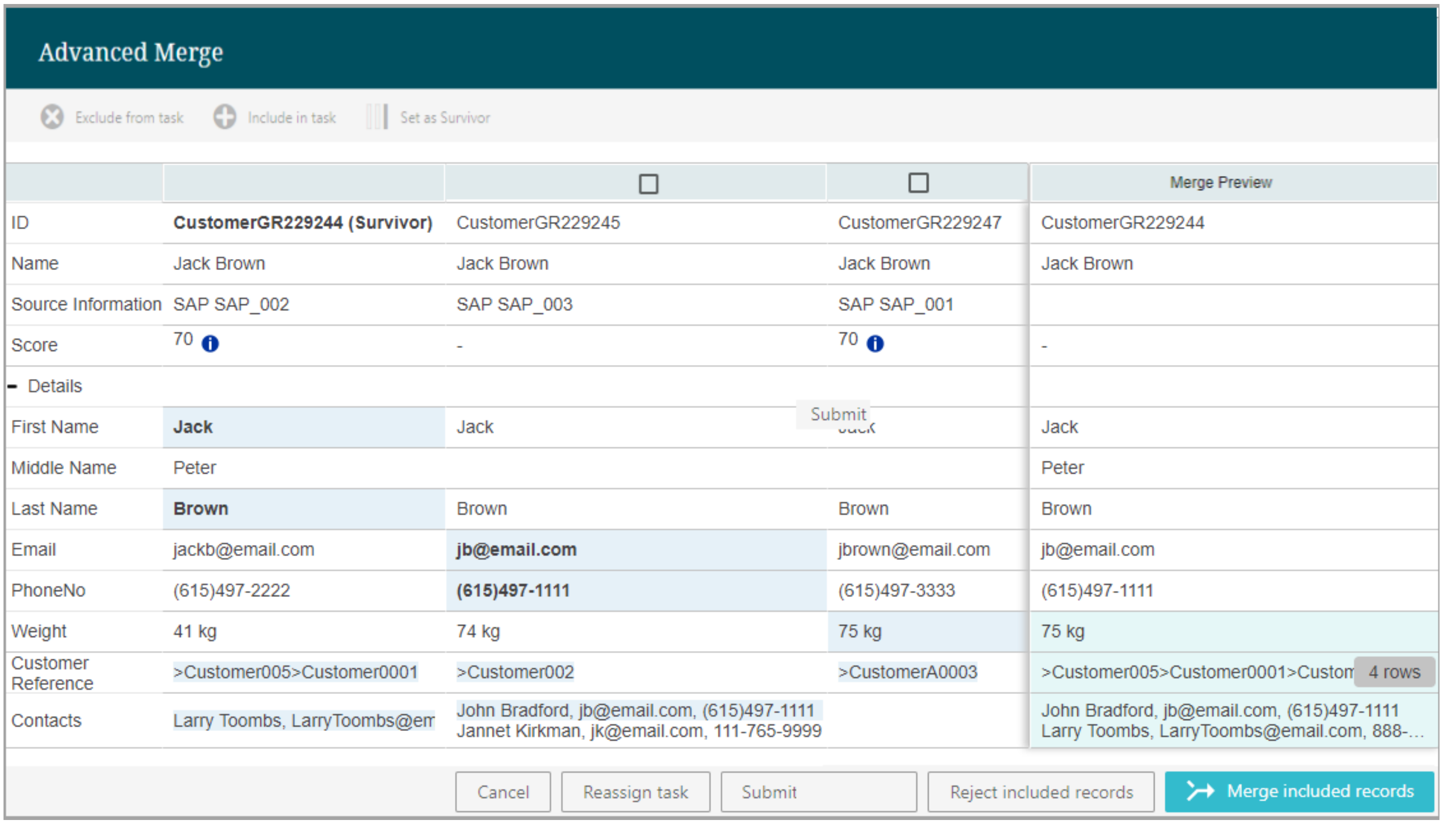
- To add a value for a reference type, double click the corresponding Merge Preview cell and click the **Add reference** link in the 'Edit Reference' dialog.
- In the node selection dialog, search for entities you want to reference via entity ID or Source Record ID.



Using the View and Buttons

When merging, individual records on the task can be excluded / included from the merge and the automatically assigned surviving record can be changed by the user. Specific values from each record can be selected for promotion to the surviving record. Attribute values can also be entered in the Merge Preview column. The end result is displayed in the Merge Preview on the right side of the dialog.

The dialog is separated into these sections: toolbar at the top, golden records, merge preview, and a toolbar at the bottom.

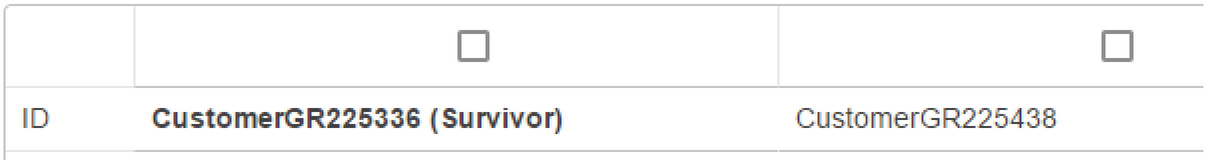


		☐	☐	Merge Preview
ID	**CustomerGR229244 (Survivor)**	CustomerGR229245	CustomerGR229247	CustomerGR229244
Name	Jack Brown	Jack Brown	Jack Brown	Jack Brown
Source Information	SAP SAP_002	SAP SAP_003	SAP SAP_001	
Score	70 ⓘ	-	70 ⓘ	-
- Details				
First Name	**Jack**	Jack	Jack	Jack
Middle Name	Peter			Peter
Last Name	**Brown**	Brown	Brown	Brown
Email	jackb@email.com	**jb@email.com**	jbrown@email.com	jb@email.com
PhoneNo	(615)497-2222	**(615)497-1111**	(615)497-3333	(615)497-1111
Weight	41 kg	74 kg	75 kg	75 kg
Customer Reference	>Customer005>Customer0001	>Customer002	>CustomerA0003	>Customer005>Customer0001>Custom 4 rows
Contacts	Larry Toombs, LarryToombs@em	John Bradford, jb@email.com, (615)497-1111 Jannet Kirkman, jk@email.com, 111-765-9999		John Bradford, jb@email.com, (615)497-1111 Larry Toombs, LarryToombs@email.com, 888-…

1. The top toolbar allows users to take actions on the selected records.
   - Click the **Exclude from task** button to exclude the selected records from the merge. The values of the record are disabled to signify the record's exclusion.

   - Click the **Include in task** button to include the selected records in the merge. This option is only available for records that have been excluded.

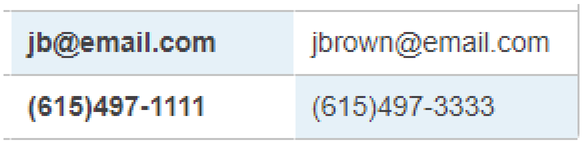
   - Click the **Set as Survivor** button to designate the selected record as the surviving record. The ID of the surviving record is marked to signify its status.



	☐	☐
ID	**CustomerGR225336 (Survivor)**	CustomerGR225438

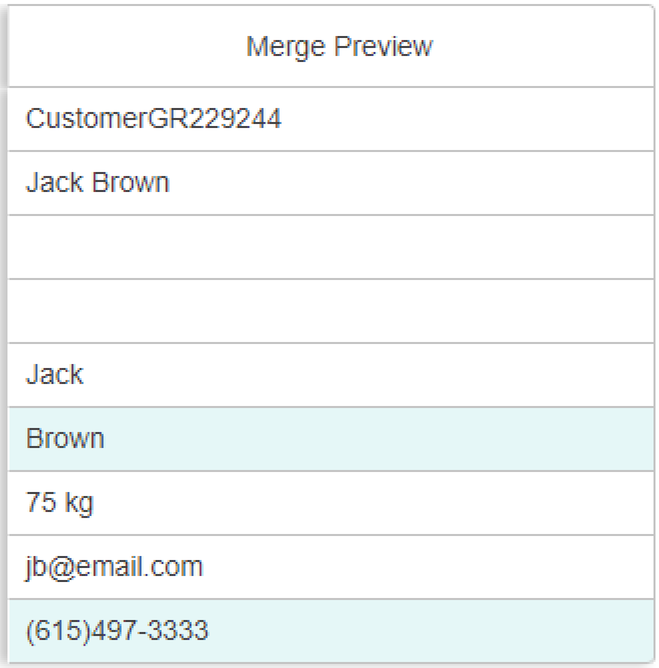
2. The golden record view in the center of the dialog lists all mapped headers.
   - Individual values selected for promotion to the surviving record are highlighted in blue.

   - Values promoted based on survivorship rules appear in bold.



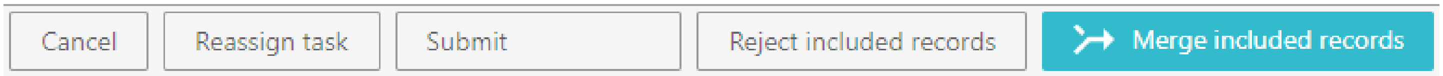
jb@email.com	jbrown@email.com
(615)497-1111	(615)497-3333

3. The **Merge Preview** column displays all of the values that will survive after the merge.
   - Values that have been manually selected for promotion to the surviving record are highlighted in blue.
   - Attribute, data container, and reference to target object values added manually are highlighted in green.



Merge Preview
CustomerGR229244
Jack Brown
Jack
Brown
75 kg
jb@email.com
(615)497-3333

4. The bottom toolbar allows users to perform the following actions:
   - **Cancel** the dialog.
   - **Reassign task** to another user.
   - **Submit** the task to the next state in the workflow.
   - **Reject included records** as the duplicates and close the task. The Exclude / Include buttons can be used to reject matches on a per-record basis.
   - **Merge included records** into one surviving record. All records are automatically included in the merge unless they have been manually excluded via the **Exclude from task** action button.

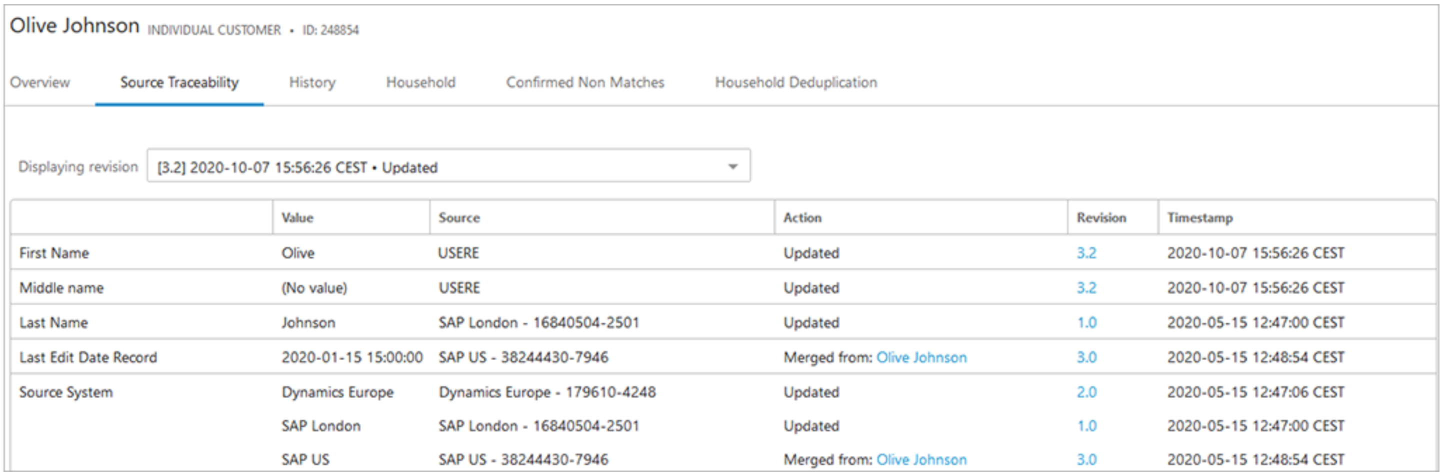


Golden Record Source Traceability Screen

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

Note: This screen can only display the revision history of golden records generated via Match and Merge.

The Golden Record Source Traceability Screen displays the revision history of a golden record. It can be configured with header rows to display the values of attributes, attribute groups, data container attributes, and reference types. This allows the user to track changes to individual aspects of a golden record, to view the system from which the new values originated, and to verify when the changes were made.



Olive Johnson INDIVIDUAL CUSTOMER • ID: 248854

Displaying revision [3.2] 2020-10-07 15:56:26 CEST • Updated

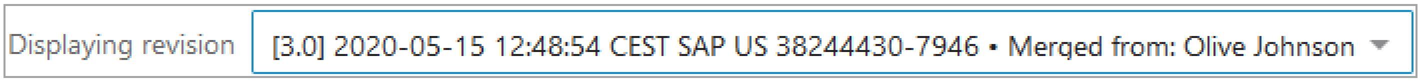
	Value	Source	Action	Revision	Timestamp
First Name	Olive	USERE	Updated	3.2	2020-10-07 15:56:26 CEST
Middle name	(No value)	USERE	Updated	3.2	2020-10-07 15:56:26 CEST
Last Name	Johnson	SAP London - 16840504-2501	Updated	1.0	2020-05-15 12:47:00 CEST
Last Edit Date Record	2020-01-15 15:00:00	SAP US - 38244430-7946	Merged from: Olive Johnson	3.0	2020-05-15 12:48:54 CEST
Source System	Dynamics Europe	Dynamics Europe - 179610-4248	Updated	2.0	2020-05-15 12:47:06 CEST
	SAP London	SAP London - 16840504-2501	Updated	1.0	2020-05-15 12:47:00 CEST
	SAP US	SAP US - 38244430-7946	Merged from: Olive Johnson	3.0	2020-05-15 12:48:54 CEST

The columns that display **cannot** be configured and include:

- Value – displays the value of the attribute / reference.
- Source – displays the source system and source record ID from which the revision originated.
- Action – displays the type of event that caused the revision and a link to the golden record for the merge / unmerge action, as defined below:
    - Updated – an update from a source system or a manual update from a user.
    - Merged from – a merge from another golden record.
    - Merged into – a merge into another golden record.
    - Unmerged from – an unmerge from another golden record.
    - Unmerged into – an unmerge into another golden record.
- Revision – a link to the golden record version from which the revision originated.
- Timestamp – the timestamp of the revision.

Note: The merge / unmerge actions are only present for merges and unmerges done after upgrading to release 10.0 (now unsupported) or a newer version. Prior to that this information was not stored and ‘Updated’ displays in the column.

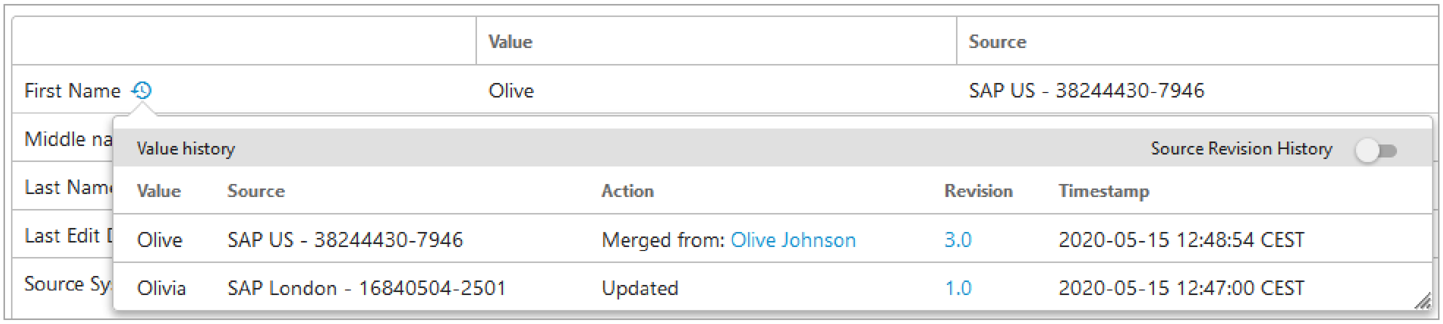
View previous versions of the golden record via the 'Displaying version' dropdown.



Displaying revision [3.0] 2020-05-15 12:48:54 CEST SAP US 38244430-7946 • Merged from: Olive Johnson

Value Traceability Popup

Click the header values to display a popup with revision history for the individual values.

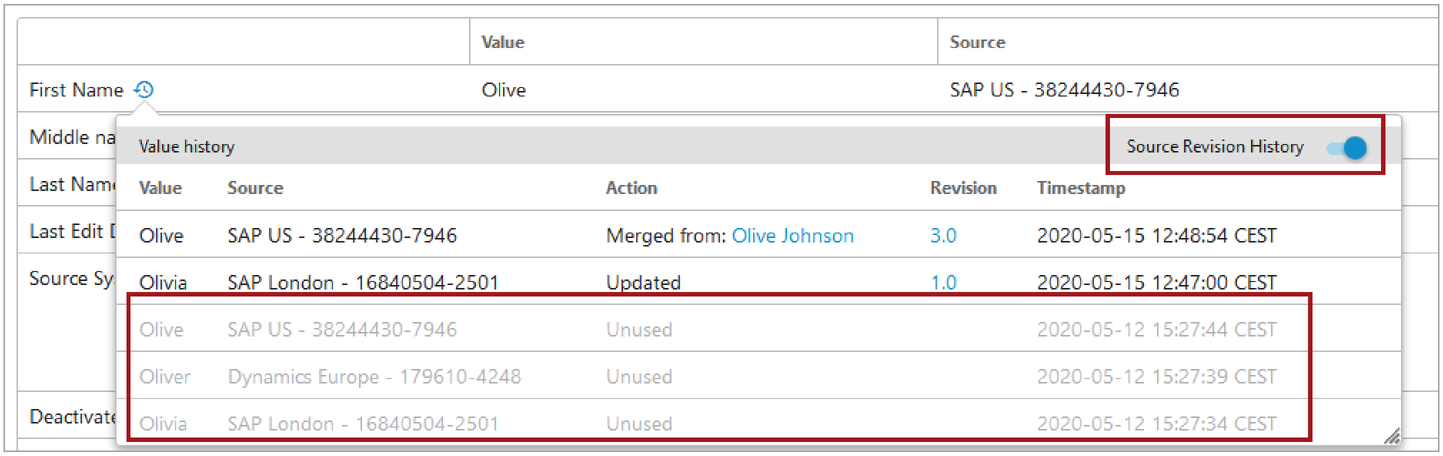


	Value	Source
First Name	Olive	SAP US - 38244430-7946
Middle na		
Last Nam		
Last Edit [		
Source Sy		

Value history — Source Revision History

Value	Source	Action	Revision	Timestamp
Olive	SAP US - 38244430-7946	Merged from: Olive Johnson	3.0	2020-05-15 12:48:54 CEST
Olivia	SAP London - 16840504-2501	Updated	1.0	2020-05-15 12:47:00 CEST

If the golden record object type is configured to keep the source data, the ‘Source Revision History' toggle is shown. Activate the toggle to display all current and historical source data that did not survive on the golden record in a gray color. For these values, 'Unused' is displayed in the Action column.



	Value	Source
First Name	Olive	SAP US - 38244430-7946
Middle na		
Last Nam		
Last Edit [		
Source Sy		
Deactivate		

Value history — Source Revision History

Value	Source	Action	Revision	Timestamp
Olive	SAP US - 38244430-7946	Merged from: Olive Johnson	3.0	2020-05-15 12:48:54 CEST
Olivia	SAP London - 16840504-2501	Updated	1.0	2020-05-15 12:47:00 CEST
Olive	SAP US - 38244430-7946	Unused		2020-05-12 15:27:44 CEST
Oliver	Dynamics Europe - 179610-4248	Unused		2020-05-12 15:27:39 CEST
Olivia	SAP London - 16840504-2501	Unused		2020-05-12 15:27:34 CEST

For more information, refer to the **Match and Merge Traceability** topic.

Configuration

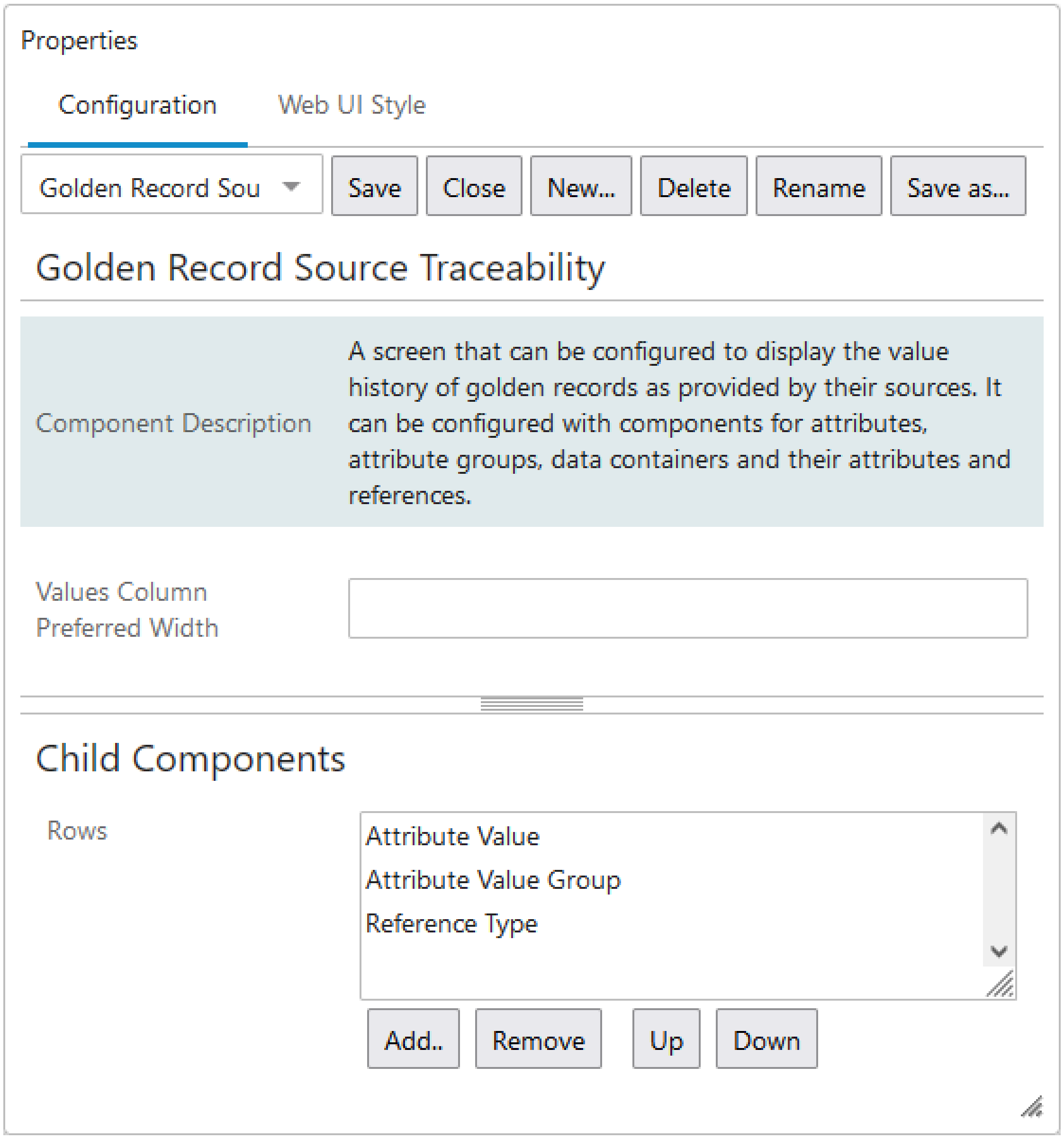
The Golden Record Source Traceability Screen allows users to customize which value headers appear on the source traceability table.

Prerequisites

It is expected that anyone configuring the Golden Record Source Traceability Screen component is familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** topic of the **Web User Interfaces** documentation.

Use these steps to create and configure a new Source Traceability screen:

1. Open the designer and click **New...**.
2. Select 'Golden Record Source Traceability', enter a Screen ID, and click **Add**.



Golden Record Sou

Golden Record Source Traceability

Component Description: A screen that can be configured to display the value history of golden records as provided by their sources. It can be configured with components for attributes, attribute groups, data containers and their attributes and references.

Values Column Preferred Width:

Child Components

Rows:
Attribute Value
Attribute Value Group
Reference Type

3. Configure the following parameters:
   - For the **Value Column Preferred Width** parameter, enter the preferred width of the value columns in pixels.
   - For the Child Components **Rows** parameter, click **Add...** and select a value header component from the list that appears.

     - Attribute Value - displays the value of the specified attribute on the golden record and requires additional configuration.
     - Attribute Value Group - displays the values of the specified group's attributes on the golden record and requires additional configuration. Attributes added to a selected Attribute Group are automatically included and displayed.

        - Data Container Attribute Value - requires a calculated attribute to display the value for the specified data container attribute on the golden record and requires additional configuration.
        - Reference Type - displays the value of the specified reference type and requires additional configuration.
    - Once a value header is selected and configured, click **Add**.

4. Configure the Golden Record Source Traceability Screen as a Tab Page on a Node Details Screen, as defined in the **Tab Pages** topic of the **Web User Interfaces** documentation.

Configuring the Unmerge Wizard

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

The Unmerge Wizard Screen uses built-in logic to unmerge matched golden records. For information on unmerge and the required configuration, refer to the **Match and Merge Traceability** topic.

Regardless of the way the unmerge is started, unmerging records is managed via the Unmerge wizard. Unmerging is only available for object types that are included in the 'Golden Record Object Types' parameter on the ‘Matching - Merge Golden Record’ component model.

- Ad hoc unmerging is intended for users who are knowledgeable about the data and want to start the unmerge wizard. This is defined in the **Unmerging Golden Records** topic.
- Workflow-based unmerging is intended to add a level of control to the unmerge process by initiating a merged record into the initial state of the unmerge workflow where a knowledgeable user can decide to continue or exit the unmerge process. This is defined in the **Creating an Unmerge Golden Record Clerical Review Workflow** topic.

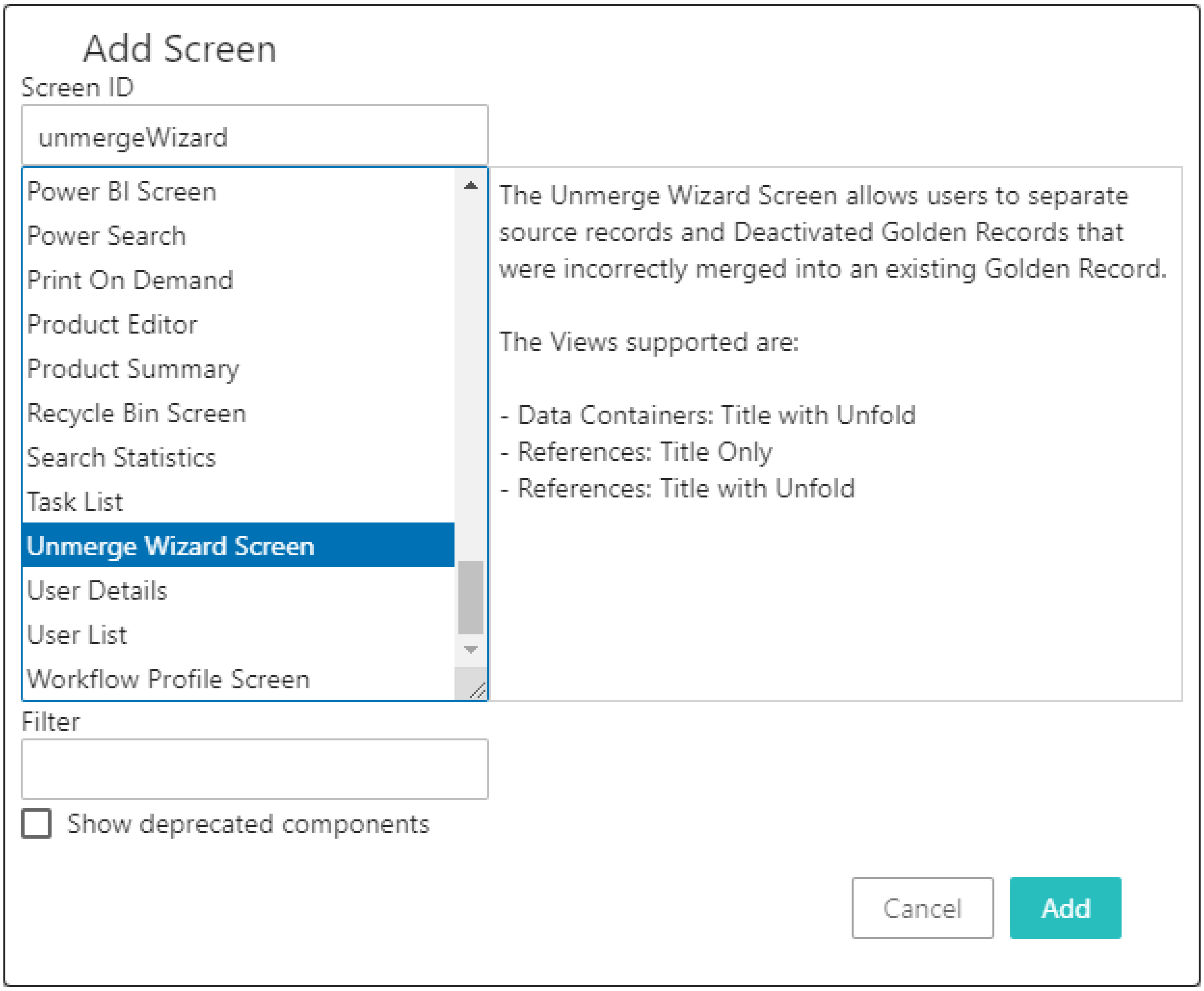
Prerequisites

It is expected that anyone configuring the Unmerge Wizard Screen component is familiar with the Web UI designer as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** topic of the **Web User Interfaces** documentation.

Configuration

Use the following steps to create and configure the Unmerge Wizard.

1. Create a new 'Unmerge Wizard Screen' as defined in the **Creating a New Screen** section of the **Design Mode Basics** topic of the **Web User Interfaces** documentation.



Add Screen

Screen ID: unmergeWizard

The Unmerge Wizard Screen allows users to separate source records and Deactivated Golden Records that were incorrectly merged into an existing Golden Record.

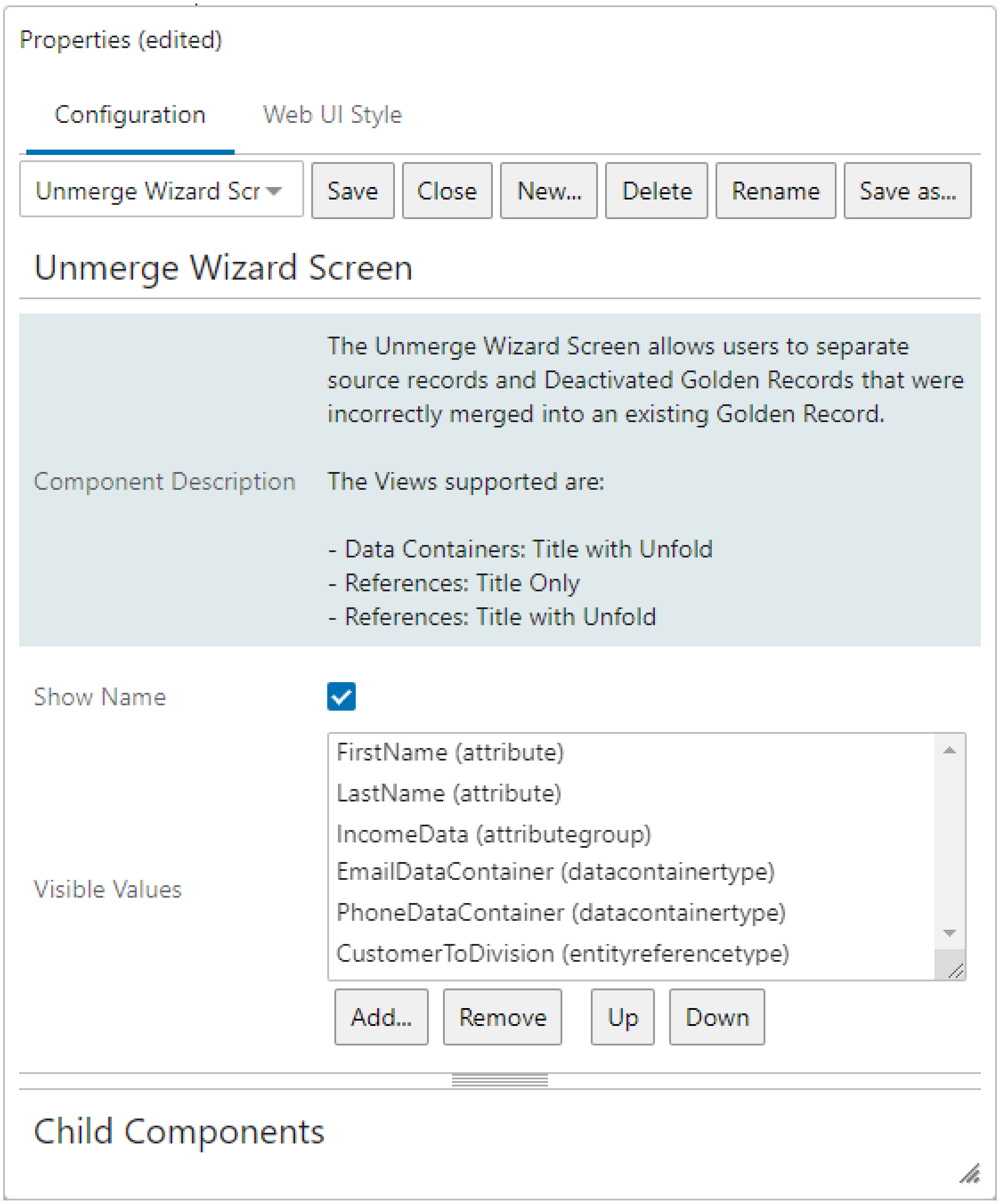
The Views supported are:

- Data Containers: Title with Unfold
- References: Title Only
- References: Title with Unfold

Filter:

☐ Show deprecated components

2. On the Unmerge Wizard Screen configure the following parameters.



Unmerge Wizard Screen

Component Description: The Unmerge Wizard Screen allows users to separate source records and Deactivated Golden Records that were incorrectly merged into an existing Golden Record.

The Views supported are:

- Data Containers: Title with Unfold
- References: Title Only
- References: Title with Unfold

Show Name: ☑

Visible Values:
FirstName (attribute)
LastName (attribute)
IncomeData (attributegroup)
EmailDataContainer (datacontainertype)
PhoneDataContainer (datacontainertype)
CustomerToDivision (entityreferencetype)

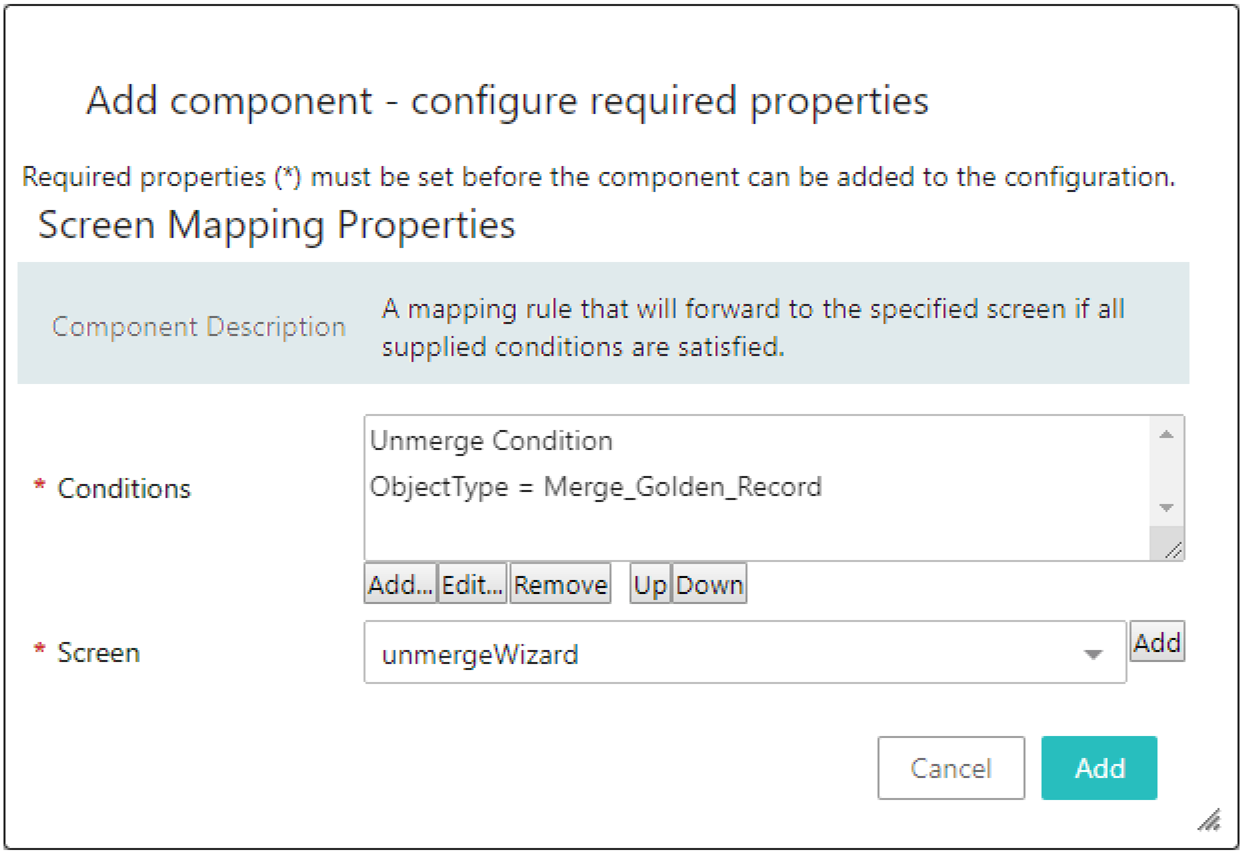
- For the **Show Name** parameter, when checked the 'Name' of the node displays.
- For the **Visible Values** parameter, add the elements to be displayed on the Unmerge Wizard Screen when unmerging. Click the 'Add...' button and select the desired elements from the list below.

> **Note:** All valid data types, even those not visible, are part of the unmerge operation.

- **Attributes** - displays the selected attributes.
- **Attribute Groups** - displays the selected attribute group including attributes in nested attribute groups. Data containers and references that are part of the attribute groups are not supported by this selection and must be configured separately.
- **Data Containers** - displays all attributes and references valid for the selected data container as configured in the [MAIN] -> [Global Representation List].
- **References** - displays all attributes valid for the reference valid for the selected data container as configured in the [MAIN] -> [Global Representation List].

The order of the Visible Values determines the ordering in the screen. To change the order, select an entry and click the 'Up' or 'Down' button.

3. Click **Save**
4. Add the screen mapping as defined in the **Mappings** topic in the **Web User Interfaces** documentation.
    - Navigate to the ---MAIN--- properties, on the Mappings parameter click the Add button.
    - For the **Conditions** parameter, add the 'Unmerge Condition' and the appropriate golden record object type condition.
    - For the **Screen** parameter, select your Unmerge Wizard Screen.
    - Click the **Add** button.



Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Screen Mapping Properties

Component Description: A mapping rule that will forward to the specified screen if all supplied conditions are satisfied.

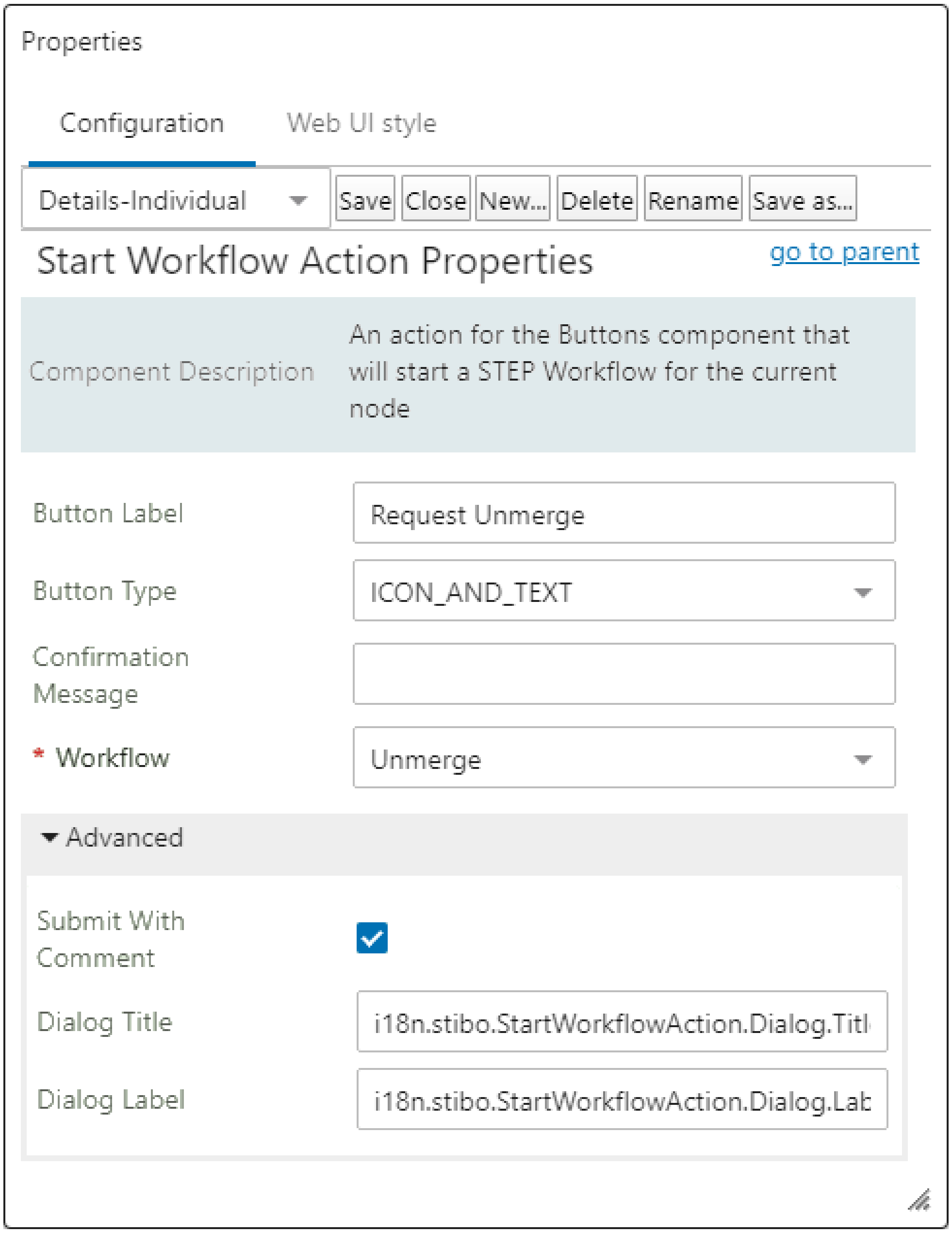
* Conditions: Unmerge Condition
ObjectType = Merge_Golden_Record

* Screen: unmergeWizard

5. Click **Save**.

6. If data stewards use an unmerge workflow, add an unmerge button on the entity details screen(s). For more information, refer to the **Creating an Unmerge Golden Record Clerical Review Workflow** topic.

   - Navigate to the entity details screen.
   - For the **Button Label** parameter, add descriptive text such as 'Request Unmerge' as shown below.
   - For the **Workflow** parameter, select the unmerge workflow.
   - In the Advanced section, on the **Submit with Comment** parameter, when checked, users can provide information from workflow states to aid in unmerging. The comment enters the workflow as a process note and is displayed on the unmerge screen along with other process notes.



Start Workflow Action Properties

Component Description: An action for the Buttons component that will start a STEP Workflow for the current node

Button Label: Request Unmerge

Button Type: ICON_AND_TEXT

Confirmation Message:

* Workflow: Unmerge

Submit With Comment: ☑

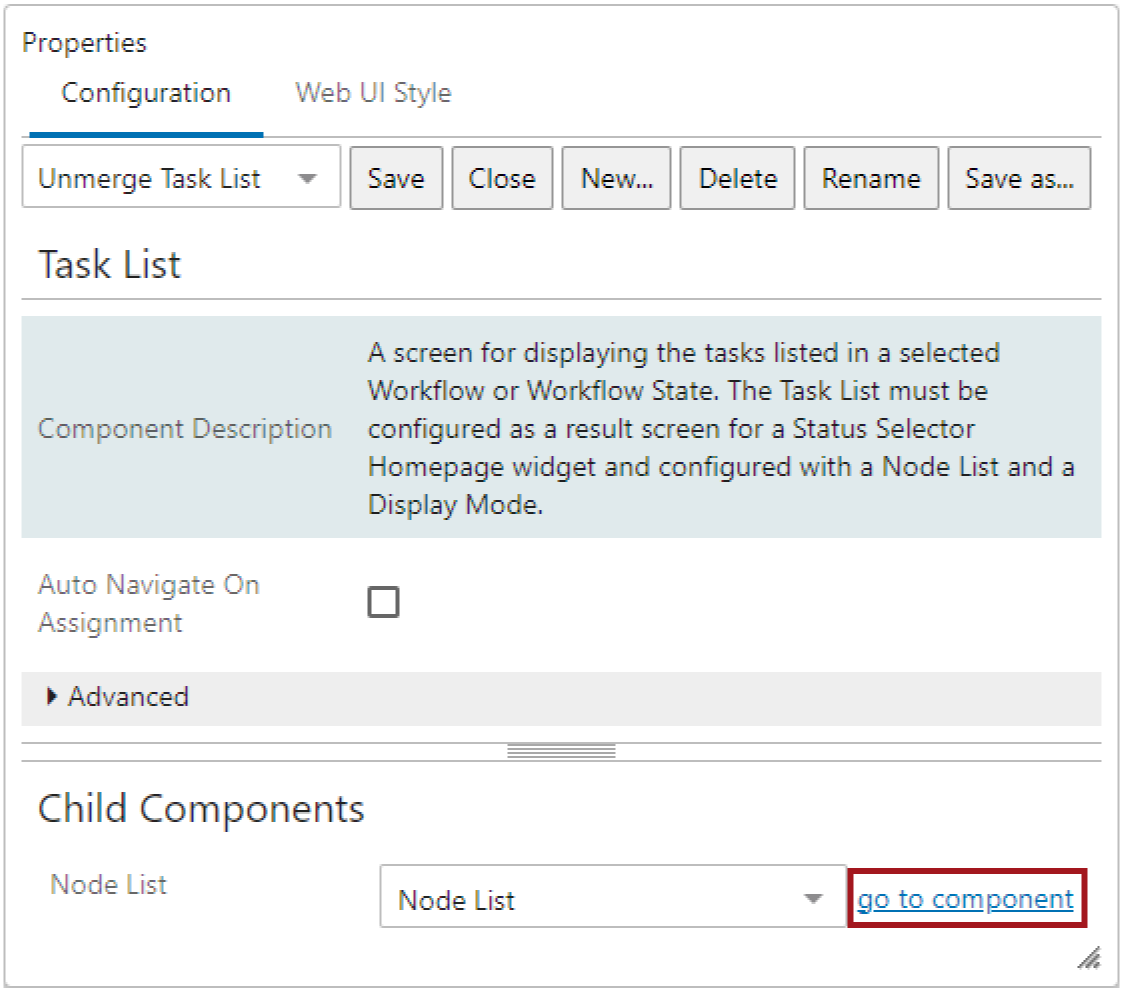
Dialog Title: i18n.stibo.StartWorkflowAction.Dialog.Titl

Dialog Label: i18n.stibo.StartWorkflowAction.Dialog.Lab

7. Click **Save**.

8. If data stewards use an unmerge workflow, add an unmerge button and/or edit button on the task list screen (s). For more information, refer to the **Creating an Unmerge Golden Record Clerical Review Workflow** topic. Unmerge can also be performed without a workflow using an ‘Unmerge Action’ button on a node details screen.

   - On the Task List screen, click the 'go to the component' link on the Node List child component. Steps for creating a new screen are outlined in the **Creating a New Screen** section of the **Design Mode Basics** topic in the **Web User Interfaces** documentation.



Unmerge Task List

Task List

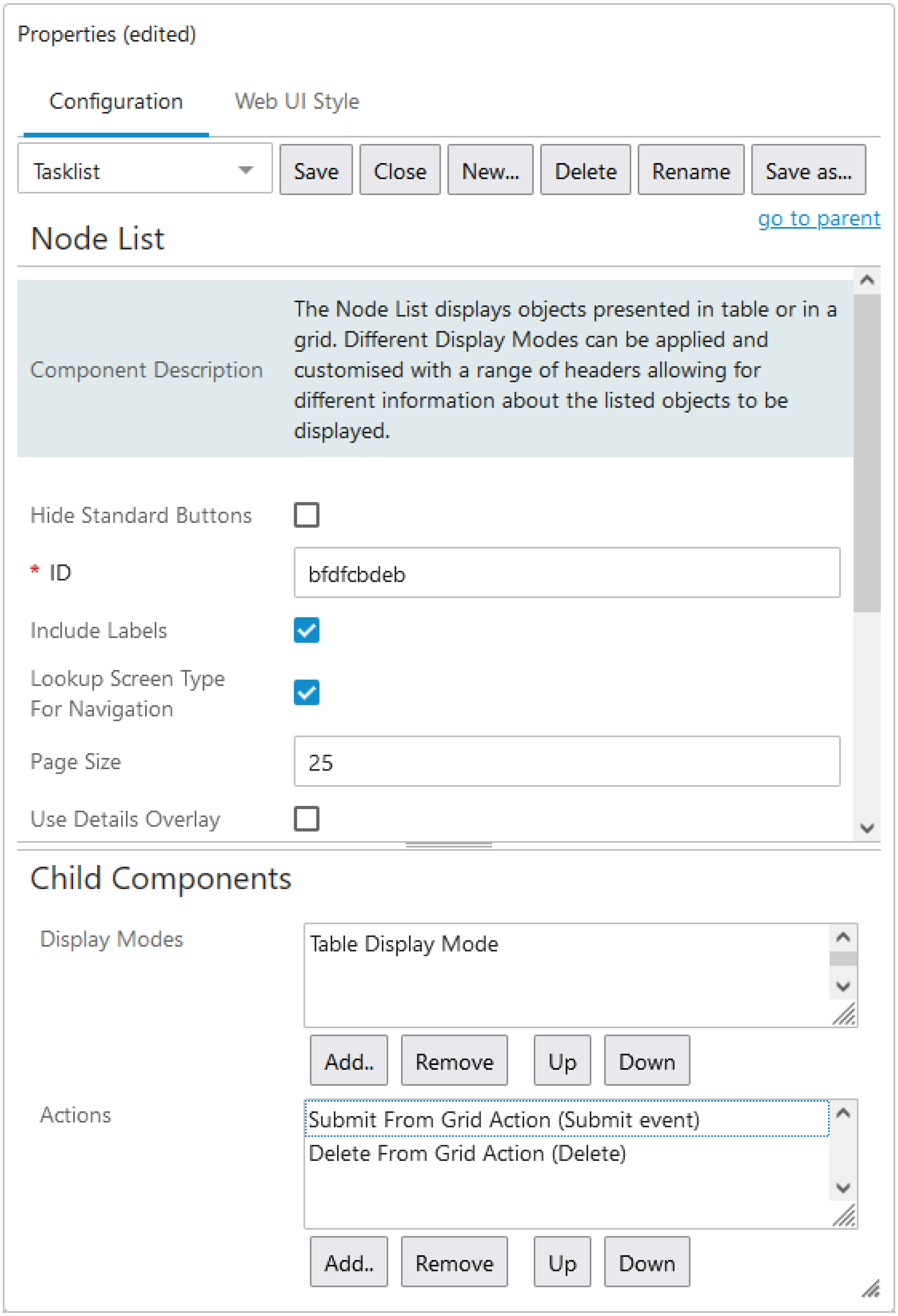
Component Description	A screen for displaying the tasks listed in a selected Workflow or Workflow State. The Task List must be configured as a result screen for a Status Selector Homepage widget and configured with a Node List and a Display Mode.
Auto Navigate On Assignment	☐

Child Components

Node List: Node List

   - In the Node List properties Child Components section, on the Display Mode parameter, click Add, select **Table Display Mode**, and include the attributes to be shown in the Task List.

   - In the 'Actions' field, add the following:
     - **Submit From Grid Action -** moves an entity to the next step in the workflow.
     - **Delete From Grid Action -** takes an entity out of the workflow.



Properties (edited)

Node List

Component Description: The Node List displays objects presented in table or in a grid. Different Display Modes can be applied and customised with a range of headers allowing for different information about the listed objects to be displayed.

Hide Standard Buttons: □

* ID: bfdfcbdeb

Include Labels: ☑

Lookup Screen Type For Navigation: ☑

Page Size: 25

Use Details Overlay: □

Child Components

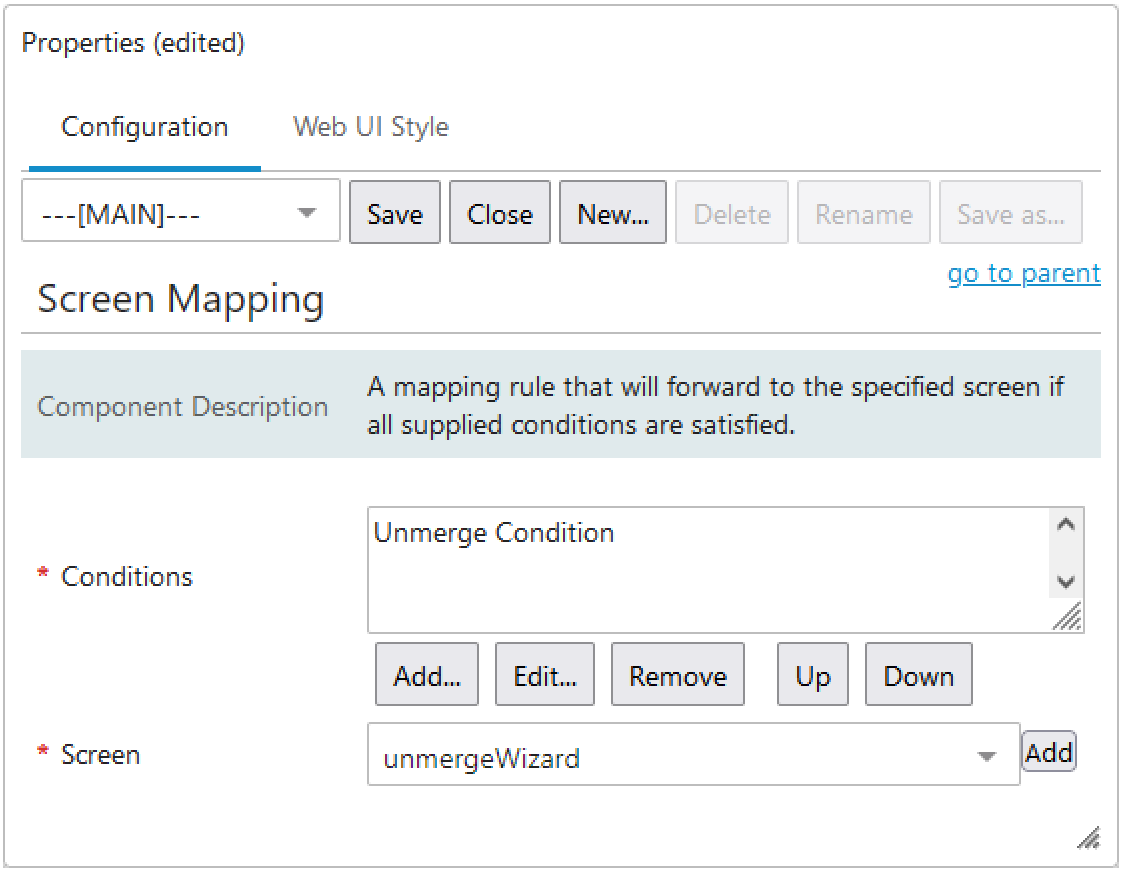
Display Modes: Table Display Mode

Actions:
Submit From Grid Action (Submit event)
Delete From Grid Action (Delete)

9. Click **Save**.

10. Add the screen mapping as defined in the **Mappings** topic in the **Web User Interfaces** documentation.

    - Navigate to the ---MAIN--- properties, on the Mappings parameter click the Add button.
    - For the **Conditions** parameter, add the unmerge Flow and State.
    - For the **Screen** parameter, select your Unmerge Wizard Screen.
    - Click the **Add** button.



Properties (edited)

Screen Mapping

Component Description: A mapping rule that will forward to the specified screen if all supplied conditions are satisfied.

* Conditions: Unmerge Condition

* Screen: unmergeWizard

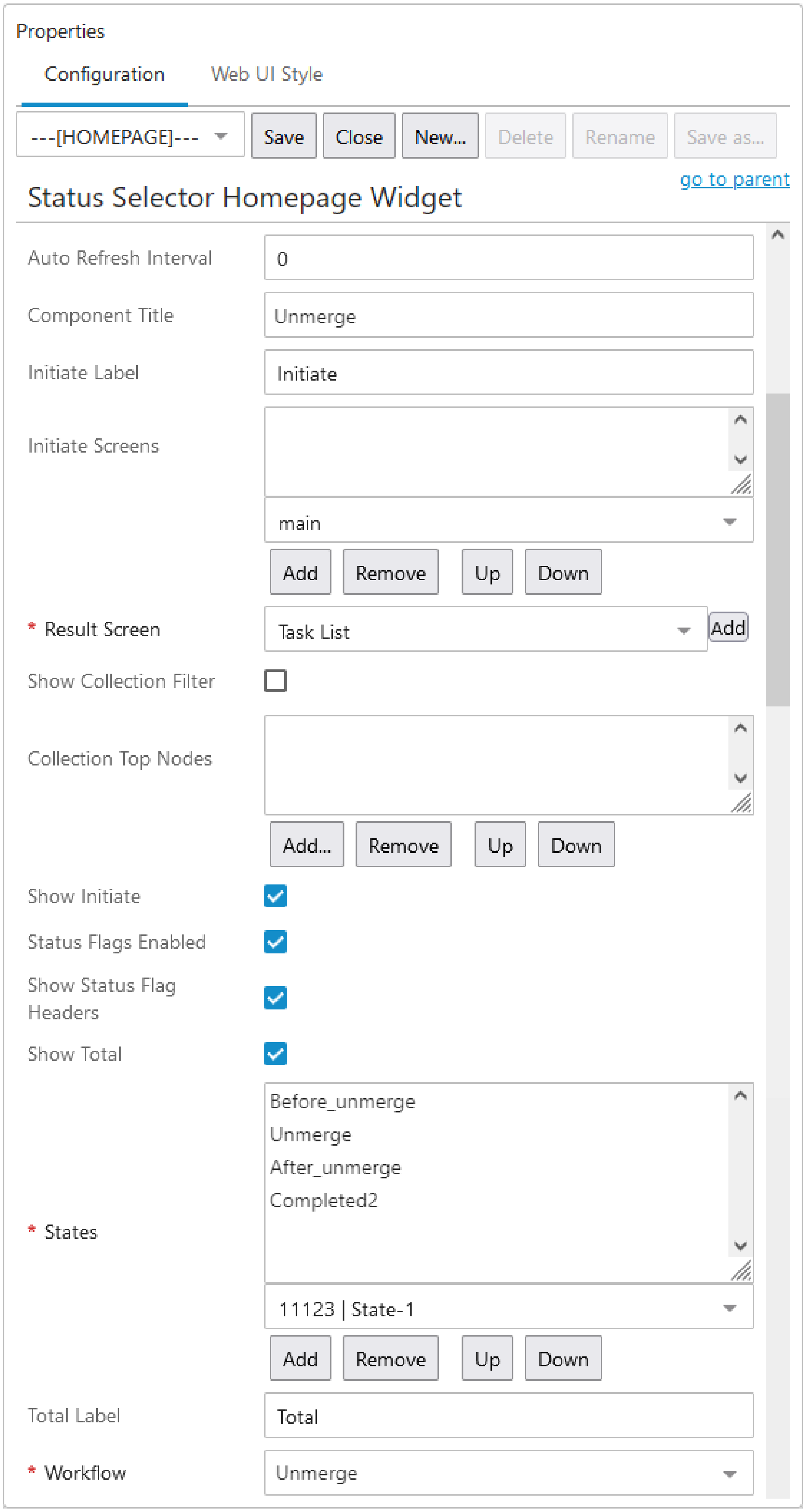
11. Click **Save**.

12. If data stewards use an unmerge workflow, add the unmerge workflow to homepage widget. For more information, refer to the **Creating an Unmerge Golden Record Clerical Review Workflow** topic. Unmerge can also be performed without a workflow using an 'Unmerge Action' button on a node details screen.

    - Add a Status Selector Homepage Widget as defined in the **Homepage Widgets** topic of the **Web User Interfaces** documentation.

    - Set the following parameters on the Status Selector Homepage Widget Properties as shown below:

- **Result Screen -** add the Unmerge Tasklist as created in this topic.
- **States -** add the states as configured in the unmerge workflow.
- **Workflow -** select the created unmerge workflow.



Status Selector Homepage Widget

Auto Refresh Interval: 0

Component Title: Unmerge

Initiate Label: Initiate

Initiate Screens: main

* Result Screen: Task List

Show Collection Filter: □

Collection Top Nodes:

Show Initiate: ☑

Status Flags Enabled: ☑

Show Status Flag Headers: ☑

Show Total: ☑

* States: Before_unmerge
Unmerge
After_unmerge
Completed2

11123 | State-1

Total Label: Total

* Workflow: Unmerge

13. Click **Save** and **Close**.

Using the Unmerge Wizard

This functionality is used by a Match and Merge solution. For more information, refer to the **Match and Merge** topic and the **Configuring Match and Merge** topic.

The unmerge process run by the Unmerge Wizard includes:

- the **Distribute Source Records** step, where a user determines how values from source records should be used (or left unused) on the golden records, including moving source records, reactivating golden records, and creating new golden records.
- the **Select Survivorship Values** step, where a user accepts or overrides the values for the modified golden records.

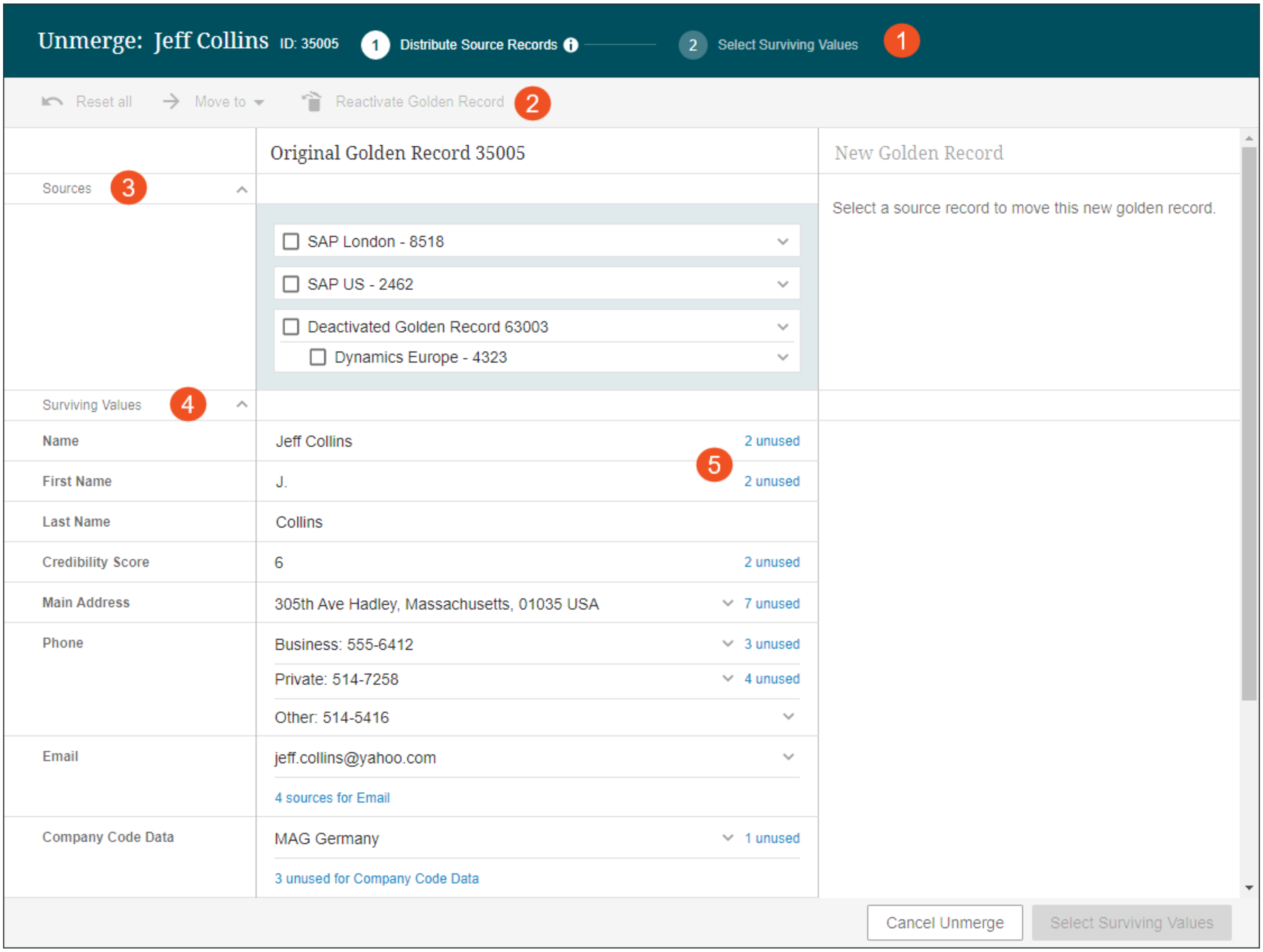
This topic includes:

- The **Managing Source Records and Golden Records** section which includes steps to move a source record and steps to reactivate a golden record.
- The **Finalizing an Unmerge** section which includes steps to complete the unmerge process.

Manage Source Records and Golden Records

Regardless of the way the unmerge is started, unmerging records is managed via the Unmerge wizard. Unmerging is only available for object types that are included in the 'Golden Record Object Types' parameter on the ‘Matching - Merge Golden Record’ component model.

- Ad hoc unmerging is intended for users who are knowledgeable about the data and want to start the unmerge wizard. This is defined in the **Unmerging Golden Records** topic.
- Workflow-based unmerging is intended to add a level of control to the unmerge process by initiating a merged record into the initial state of the unmerge workflow where a knowledgeable user can decide to continue or exit the unmerge process. This is defined in the **Creating an Unmerge Golden Record Clerical Review Workflow** topic.



Original Golden Record 35005

New Golden Record

Select a source record to move this new golden record.

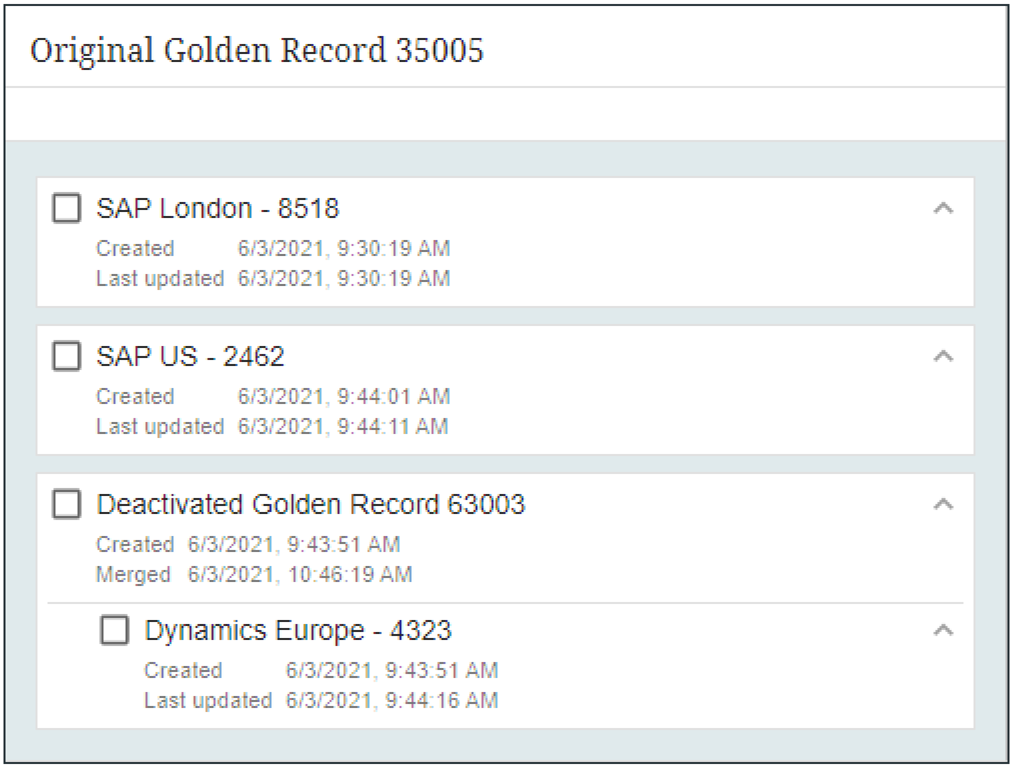
- ☐ SAP London - 8518
- ☐ SAP US - 2462
- ☐ Deactivated Golden Record 63003
  - ☐ Dynamics Europe - 4323

Surviving Values	Original Golden Record 35005	
Name	Jeff Collins	2 unused
First Name	J.	2 unused
Last Name	Collins	
Credibility Score	6	2 unused
Main Address	305th Ave Hadley, Massachusetts, 01035 USA	7 unused
Phone	Business: 555-6412	3 unused
	Private: 514-7258	4 unused
	Other: 514-5416	
Email	jeff.collins@yahoo.com	
	4 sources for Email	
Company Code Data	MAG Germany	1 unused
	3 unused for Company Code Data	

The unmerge screen includes the following elements:

1. **Unmerge Steps -** the current step of the Unmerge process is highlighted.
2. **Action Bar -** actions are enabled based on the status of the selected records:
   - Reset all: reverts the screen back to the original state. All actions / changes that were made are lost.
   - Move to: moves the selected source record(s) to another golden record or creates a new golden record. Refer to the **Moving a Source Record** section below.
   - Reactivate Golden Record: reactivates the selected golden record that was previously merged into another golden record. All source records associated with this record are also moved. Refer to the **Reactivating a Golden Record** section below.
3. **Sources -** records used to determine the original record. This section shows all deactivated golden records that were merged into the golden record being unmerged. As shown below, expand a record to display additional information.

If the system uses source record IDs when importing, those source records are shown in combination with the deactivated golden records. All source records shown are actively 'assigned' to the golden record being unmerged, while those that were previously actively assigned to the deactivated golden records are shown below them. This representation shows that the values might move along if the deactivated golden records are reactivated. For details on reactivation, refer to the **Reactivating a Golden Record** section below.



Original Golden Record 35005

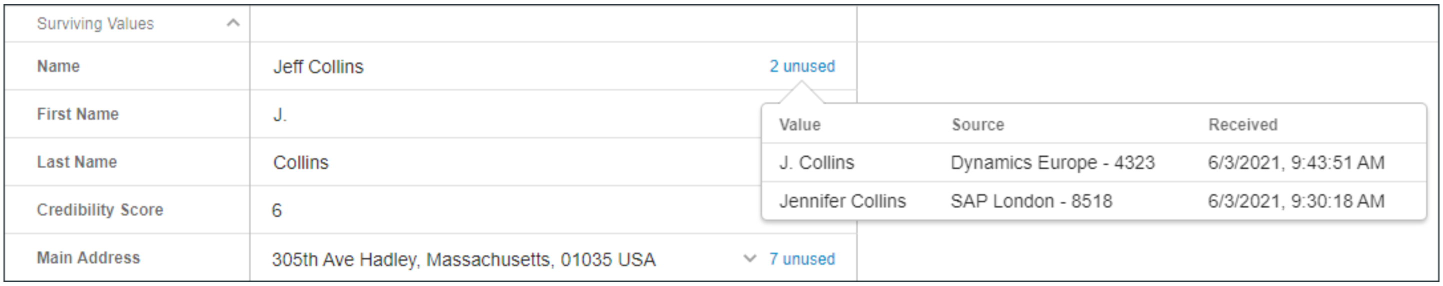
□ SAP London - 8518
Created 6/3/2021, 9:30:19 AM
Last updated 6/3/2021, 9:30:19 AM

□ SAP US - 2462
Created 6/3/2021, 9:44:01 AM
Last updated 6/3/2021, 9:44:11 AM

□ Deactivated Golden Record 63003
Created 6/3/2021, 9:43:51 AM
Merged 6/3/2021, 10:46:19 AM

□ Dynamics Europe - 4323
Created 6/3/2021, 9:43:51 AM
Last updated 6/3/2021, 9:44:16 AM

4. **Surviving Values -** a preview of the configured surviving data values after a completed unmerge operation.
5. **Unused Values -** values are determined based on data from deactivated golden records and source data. Since the data on the deactivated golden record can include several sources (and may be identical to the surviving value), traceability is required for a complete view of unused values. For more information, refer to the **Match and Merge Traceability** topic.



Surviving Values		
Name	Jeff Collins	2 unused
First Name	J.	
Last Name	Collins	
Credibility Score	6	
Main Address	305th Ave Hadley, Massachusetts, 01035 USA	7 unused

Value	Source	Received
J. Collins	Dynamics Europe - 4323	6/3/2021, 9:43:51 AM
Jennifer Collins	SAP London - 8518	6/3/2021, 9:30:18 AM

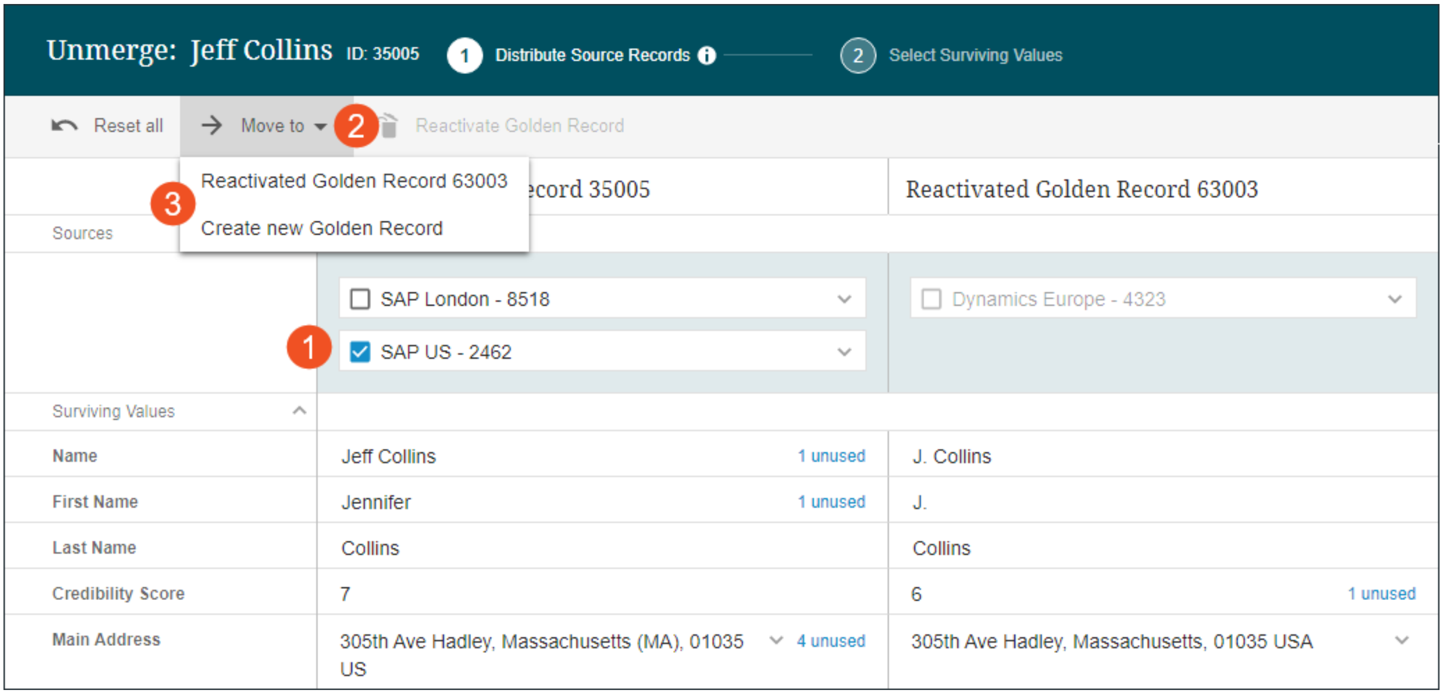
- Blue link text on a surviving value row (the Name row in the image above) shows the number of unselected attribute values associated with the record that did not survive. Click the link to display a popup that includes details.

- Blue link text below a surviving value row (the Main Address row in the image above) shows the unused references (grouped by reference target) and data containers (grouped by data container key). If no key is defined, the link shows the available sources in a list.

Moving a Source Record

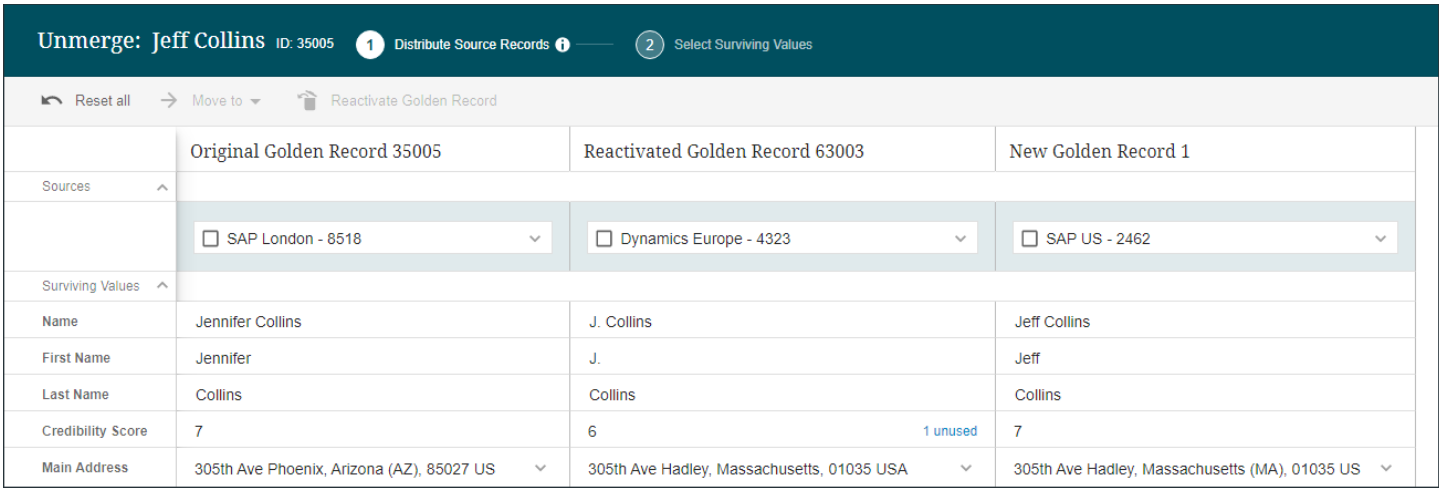
Use the following steps to move source records to another golden record or to a new golden record.

1. On the Distribute Source Records step, select one or more source records.
2. Click the **Move To** action button.
3. Choose the desired golden record for the selected source record(s).


Unmerge: Jeff Collins ID: 35005
1 Distribute Source Records
2 Select Surviving Values
Reset all
Move to
Reactivate Golden Record
Reactivated Golden Record 63003
Create new Golden Record
1
2
3

	ecord 35005	Reactivated Golden Record 63003
Sources	□ SAP London - 8518 ☑ SAP US - 2462	□ Dynamics Europe - 4323
Surviving Values		
Name	Jeff Collins 1 unused	J. Collins
First Name	Jennifer 1 unused	J.
Last Name	Collins	Collins
Credibility Score	7	6 1 unused
Main Address	305th Ave Hadley, Massachusetts (MA), 01035 US 4 unused	305th Ave Hadley, Massachusetts, 01035 USA

Unmerge displays the results for all golden records.



Unmerge: Jeff Collins ID: 35005 — 1 Distribute Source Records — 2 Select Surviving Values

Reset all · Move to · Reactivate Golden Record

	Original Golden Record 35005	Reactivated Golden Record 63003	New Golden Record 1
Sources			
	☐ SAP London - 8518	☐ Dynamics Europe - 4323	☐ SAP US - 2462
Surviving Values			
Name	Jennifer Collins	J. Collins	Jeff Collins
First Name	Jennifer	J.	Jeff
Last Name	Collins	Collins	Collins
Credibility Score	7	6 1 unused	7
Main Address	305th Ave Phoenix, Arizona (AZ), 85027 US	305th Ave Hadley, Massachusetts, 01035 USA	305th Ave Hadley, Massachusetts (MA), 01035 US

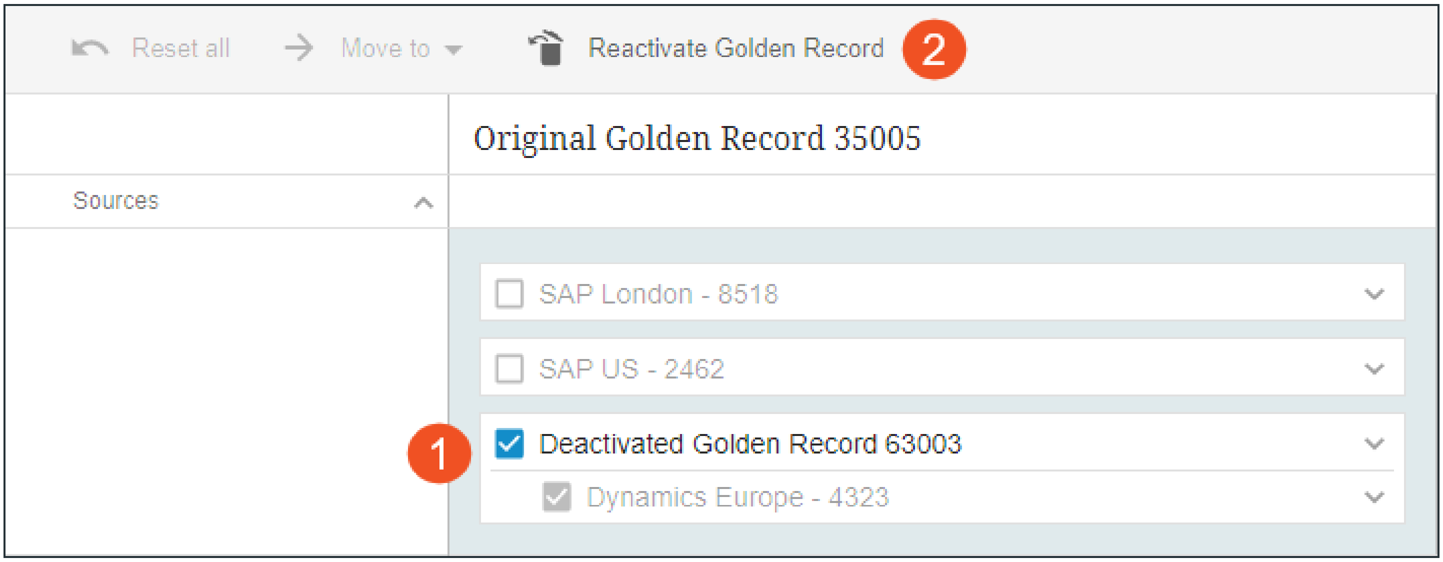
4. Click the **Select Surviving Values** button to continue with the unmerge or click **Cancel Unmerge** to close the dialog without making changes.

5. Complete the unmerge as defined in the **Finalizing an Unmerge** section below.

Reactivating a Golden Record

Use the following steps to reactivate a golden record.

1. On the Distribute Source Records step, select a deactivated source record and click the **Reactivate Golden Record** toolbar action button.



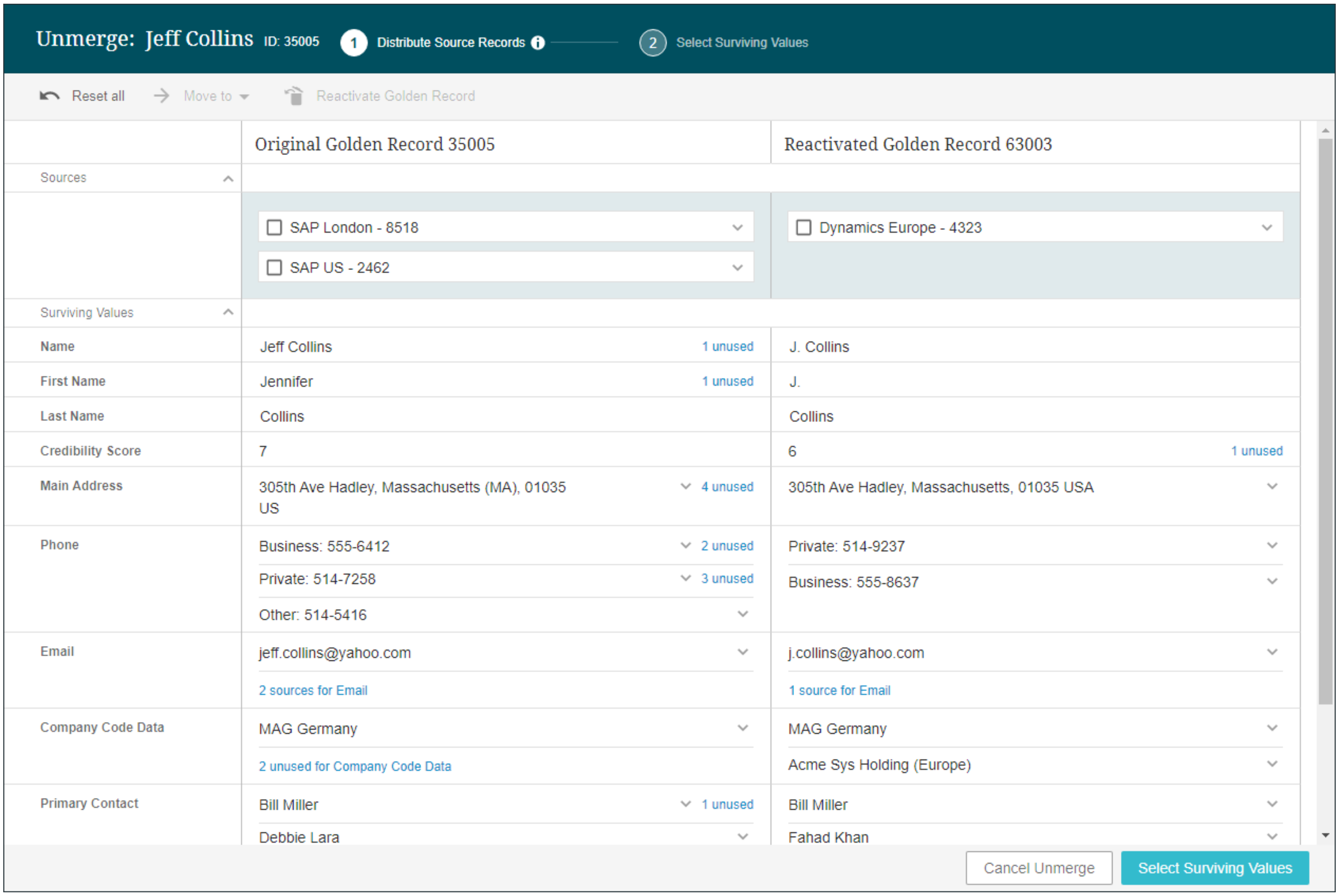
☐ SAP London - 8518

☐ SAP US - 2462

☑ Deactivated Golden Record 63003

☑ Dynamics Europe - 4323

2. Review the assigned values, the unused values, and the unused references calculated by Unmerge for the original selected golden record and the one being reactivated.


Unmerge: Jeff Collins ID: 35005
1 Distribute Source Records
2 Select Surviving Values
Reset all
Move to
Reactivate Golden Record
Cancel Unmerge
Select Surviving Values

	Original Golden Record 35005	Reactivated Golden Record 63003
Sources		
	☐ SAP London - 8518 ☐ SAP US - 2462	☐ Dynamics Europe - 4323
Surviving Values		
Name	Jeff Collins 1 unused	J. Collins
First Name	Jennifer 1 unused	J.
Last Name	Collins	Collins
Credibility Score	7	6 1 unused
Main Address	305th Ave Hadley, Massachusetts (MA), 01035 US 4 unused	305th Ave Hadley, Massachusetts, 01035 USA
Phone	Business: 555-6412 2 unused Private: 514-7258 3 unused Other: 514-5416	Private: 514-9237 Business: 555-8637
Email	jeff.collins@yahoo.com 2 sources for Email	j.collins@yahoo.com 1 source for Email
Company Code Data	MAG Germany 2 unused for Company Code Data	MAG Germany Acme Sys Holding (Europe)
Primary Contact	Bill Miller 1 unused Debbie Lara	Bill Miller Fahad Khan

3. Click the **Select Surviving Values** button to continue with the merge or click **Cancel Unmerge** to close the dialog without making changes.

4. Complete the unmerge as defined in the **Finalizing an Unmerge** section below.

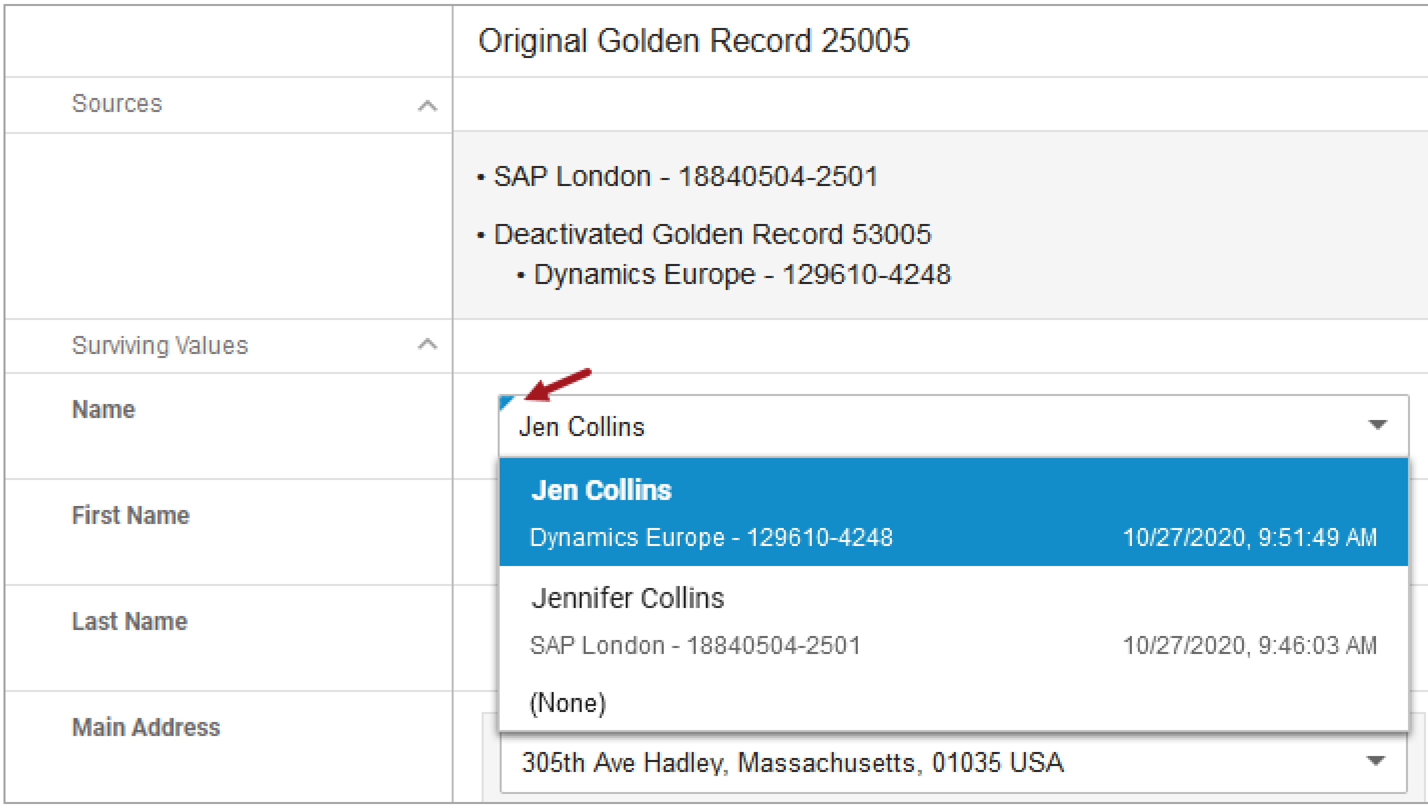
Finalizing an Unmerge

Once the new golden record is created, either from a merged record or a reactivated golden record, clicking the 'Select Surviving Values' button on the Distribute Source Records step displays the **Selecting Surviving Values** step.

Complete the merge process:

1. Verify the desired values are displayed for the golden records as follows:

   - A marker in the top left corner of a field indicates that multiple values exist. Click the dropdown to view the options, select '(None)' to erase the value.



Original Golden Record 25005

Sources

- SAP London - 18840504-2501
- Deactivated Golden Record 53005
  - Dynamics Europe - 129610-4248

Surviving Values

Name: Jen Collins

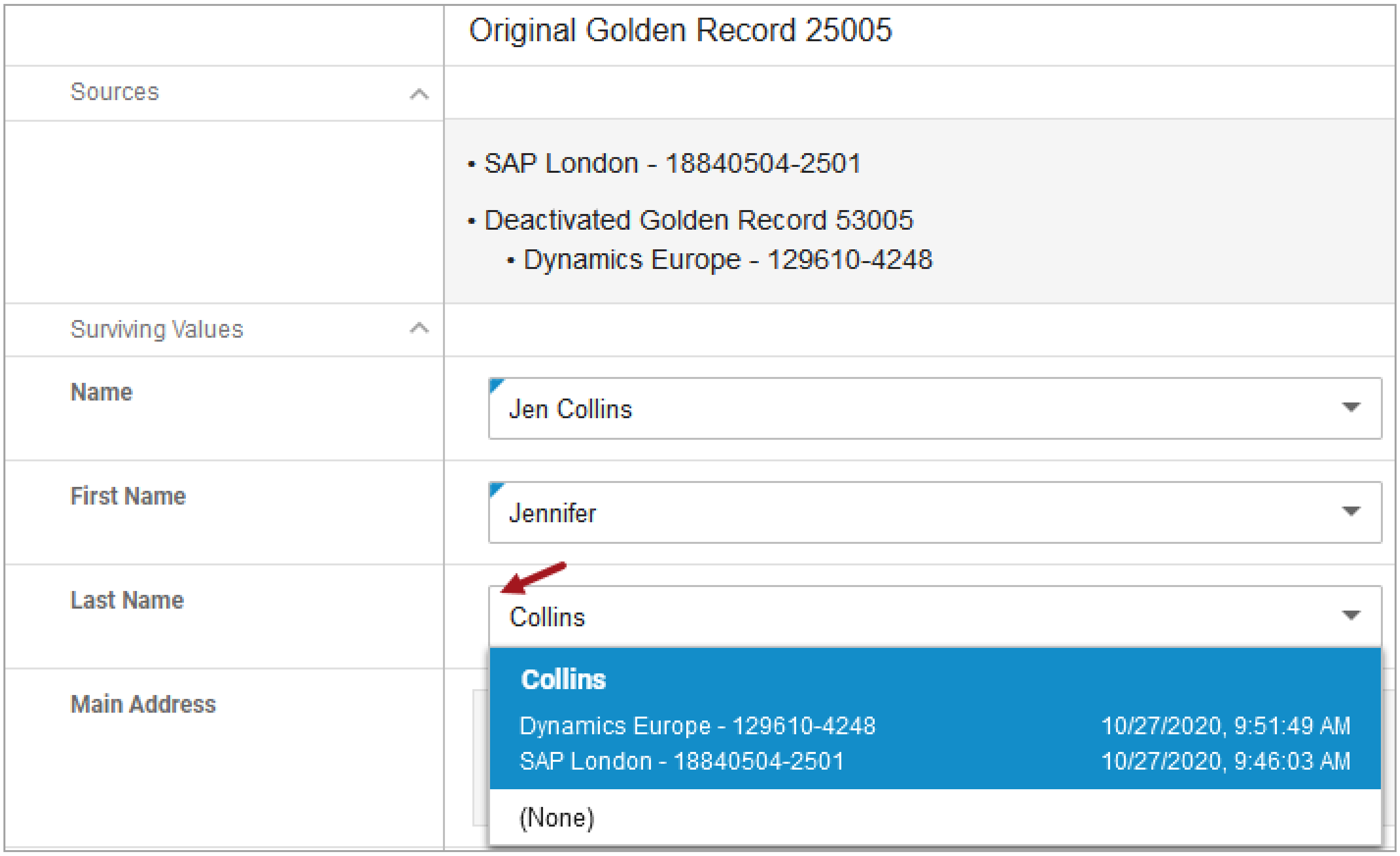
- Jen Collins — Dynamics Europe - 129610-4248 — 10/27/2020, 9:51:49 AM
- Jennifer Collins — SAP London - 18840504-2501 — 10/27/2020, 9:46:03 AM
- (None)

First Name

Last Name

Main Address: 305th Ave Hadley, Massachusetts, 01035 USA

- On a field without a marker only has the value displayed. Click the dropdown and select '(None)' to erase the value.



Original Golden Record 25005

Sources

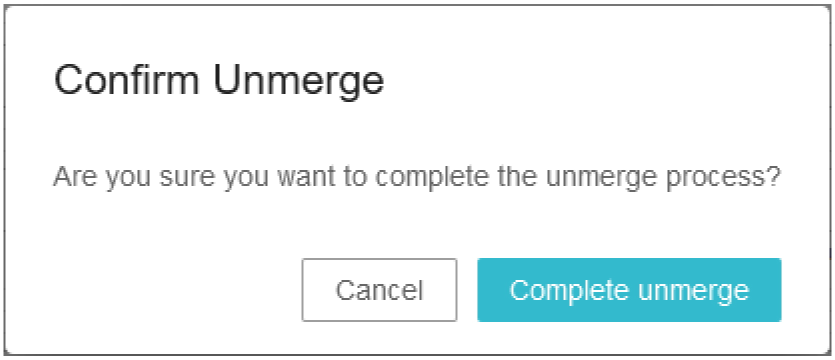
- SAP London - 18840504-2501
- Deactivated Golden Record 53005
  - Dynamics Europe - 129610-4248

Surviving Values

Name	Jen Collins
First Name	Jennifer
Last Name	Collins
Main Address	

Collins	
Dynamics Europe - 129610-4248	10/27/2020, 9:51:49 AM
SAP London - 18840504-2501	10/27/2020, 9:46:03 AM
(None)	

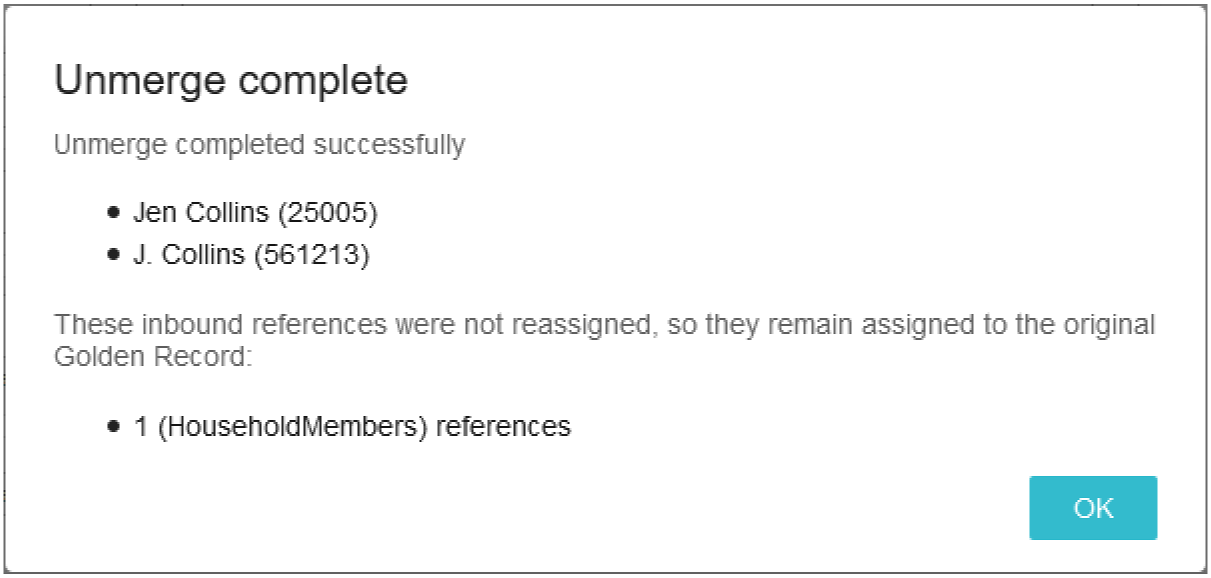
2. On the **Selecting Surviving Values** step:
   - Click **Back** to return to the Distribute Source Records step.
   - Click **Cancel Unmerge** to close the dialog without making changes.
   - Click **Complete Unmerge** to confirm the changes.
3. On the Confirm Unmerge dialog:
   - Click **Cancel** to return to the wizard.
   - Click **Complete unmerge** to continue.



Confirm Unmerge

Are you sure you want to complete the unmerge process?

4. On the Unmerge Complete dialog:
   - Review the actions taken and note any references not reassigned to be resolved below.
   - Click **OK** to close the dialog.
   - Manually review all reference types noted above and resolve as required.



Unmerge complete

Unmerge completed successfully

- Jen Collins (25005)
- J. Collins (561213)

These inbound references were not reassigned, so they remain assigned to the original Golden Record:

- 1 (HouseholdMembers) references

Match and Merge Flow Details

The level of detail for the following selected match and merge flows are intended to assist integrators and administrators in understanding and troubleshooting a match and merge solution.

This topic includes the following flow charts:

- Inbound Record Flow
- Event Processor Flow
- Merge Flow

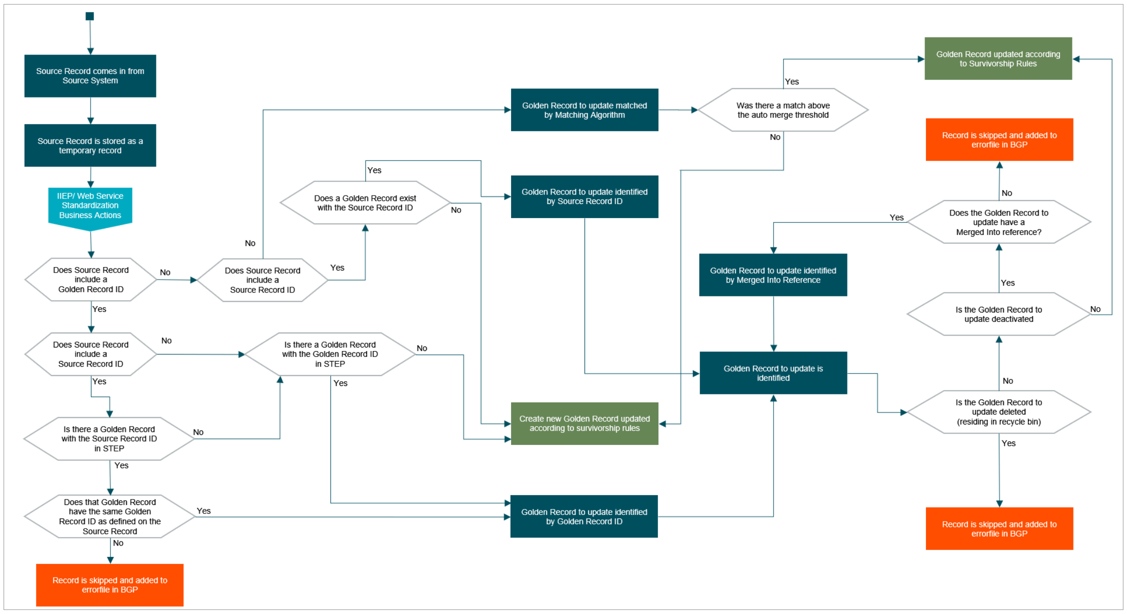
Inbound Record Flow

The **Configuring the Match Data Exchange Method** topic describes that the inbound source records can come from either and inbound integration endpoint or a web service. The flow of these is quite similar.

The **Information Flow** section of the **Match and Merge** topic explains that the incoming source record is matched against the existing golden records, and if a match is found, the information from the source record is merged into the relevant golden record using survivorship rules. If no match is found, or if the match is uncertain, a new golden record is created.

The actual identification flow is detailed in the image below. This identification flow is the same for asynchronous Integration Endpoints and synchronous web service endpoints.

View this topic in online help to explore this flowchart.


Source Record comes in from Source System
Source Record is stored as a temporary record
IIEP/ Web Service Standardization Business Actions
Does Source Record include a Golden Record ID
No
Yes
Does Source Record include a Source Record ID
No
Yes
Does a Golden Record exist with the Source Record ID
Yes
No
Does Source Record include a Source Record ID
No
Yes
Is there a Golden Record with the Golden Record ID in STEP
No
Yes
Is there a Golden Record with the Source Record ID in STEP
No
Yes
Does that Golden Record have the same Golden Record ID as defined on the Source Record
Yes
No
Record is skipped and added to errorfile in BGP
Golden Record to update matched by Matching Algorithm
Was there a match above the auto merge threshold
Yes
No
Golden Record to update identified by Source Record ID
Create new Golden Record updated according to survivorship rules
Golden Record to update identified by Golden Record ID
Golden Record updated according to Survivorship Rules
Record is skipped and added to errorfile in BGP
Does the Golden Record to update have a Merged Into reference?
No
Yes
Golden Record to update identified by Merged Into Reference
Golden Record to update is identified
Is the Golden Record to update deactivated
Yes
No
Is the Golden Record to update deleted (residing in recycle bin)
No
Yes
Record is skipped and added to errorfile in BGP

During an import, source records listed in the imported file are created as temporary STEP objects, which can be acted on by business rules, with some limitations. The references to and from the temporary object are not fully established, and the golden record ID is a temporary ID. A permanent ID is assigned later.

The diagram above details the decision to either create a new Golden Record or identify the existing Golden Record to be updated. This constitutes the 'matching' part of match and merge, even though in many systems the majority of source records are identified by an ID.

After identifying where the information in the source record belongs, the survivorship rules part of the merge determines which (if any) source values get promoted to existing golden records. When the import process is complete, the temporary object is either discarded or saved as part of the source traceability, depending on the 'Keep Source Records for Golden Record Object Types' setting in the Matching - Merge Golden Record component model, as defined in the **Configuring the Matching - Merge Golden Record Component Model** topic.

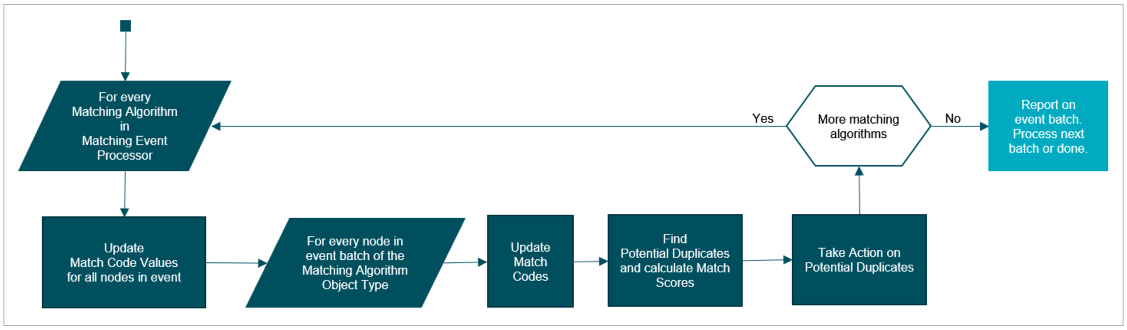
Note: During both matching and merging, the incoming source record is accessed as a temporary STEP object. As a consequence, business functions of both matching and merging are run before references to other objects are established. It is not possible to query for references to and from the source record during the import. Furthermore, match codes can only depend on values on the current object, and survivorship rules are only allowed to update values on the current object.

Event Processor Flow

The recommendation is to have one matching event processor process events across a multitude of matching algorithms if event triggering can be correctly shared between them. As a result, the matching event processor includes a number of matching algorithms and loops over them.

- For every **matching algorithm**, the event processor loops through the nodes in the event batch and updates all match code values across the event batch for that matching algorithm.
- For every **such event node**, the event processor updates the match codes on the current object (again). That ensures that match codes for the entire batch are not outdated and also that if anything is updated during the event batch processing, the match code on the current event node is current.
- For every **event node**, after the match codes are updated, the match scores are calculated, and the matching algorithm is applied.

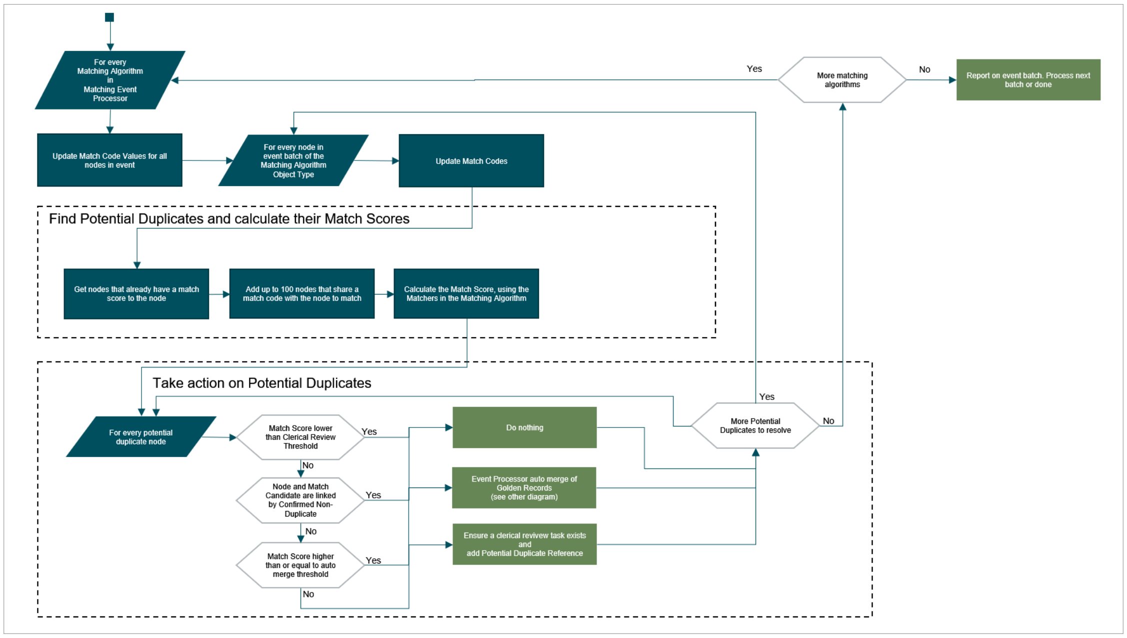
View this topic in online help to explore this flowchart.


For every Matching Algorithm in Matching Event Processor
Yes
More matching algorithms
No
Report on event batch. Process next batch or done.
Update Match Code Values for all nodes in event
For every node in event batch of the Matching Algorithm Object Type
Update Match Codes
Find Potential Duplicates and calculate Match Scores
Take Action on Potential Duplicates

Note: These flow diagrams and the descriptions in this topic, do not describe the consequence of using multiple match and merge match algorithms on the same golden record object type separated by Category, as defined in the **Configuring Matching Algorithms** topic.

When potential duplicates are identified, their match scores are calculated, and action is taken, the flow is a bit more advanced, as shown below.

View this topic in online help to explore this flowchart.


For every Matching Algorithm in Matching Event Processor
Yes
More matching algorithms
No
Report on event batch. Process next batch or done
Update Match Code Values for all nodes in event
For every node in event batch of the Matching Algorithm Object Type
Update Match Codes
Find Potential Duplicates and calculate their Match Scores
Get nodes that already have a match score to the node
Add up to 100 nodes that share a match code with the node to match
Calculate the Match Score, using the Matchers in the Matching Algorithm
Take action on Potential Duplicates
For every potential duplicate node
Match Score lower than Clerical Review Threshold
Yes
No
Node and Match Candidate are linked by Confirmed Non-Duplicate
Yes
No
Match Score higher than or equal to auto merge threshold
Yes
No
Do nothing
Event Processor auto merge of Golden Records (see other diagram)
Ensure a clerical revivew task exists and add Potential Duplicate Reference
More Potential Duplicates to resolve
Yes
No

As illustrated above, when more than 100 objects share the same match code they are not guaranteed to be compared using the matchers of the matching algorithm.

When the score exceeds the auto merge threshold, the Event Processor invokes the **Merge Flow** shown in the following section.

Merge Flow

The merge flow is invoked with standard survivor selection and with manual survivorship selection from Advanced Merge.

Standard Survivor

In the standard version, the survivor is selected based on the Merge Keep First Handler, if configured. Otherwise, the oldest Golden Record is the survivor.

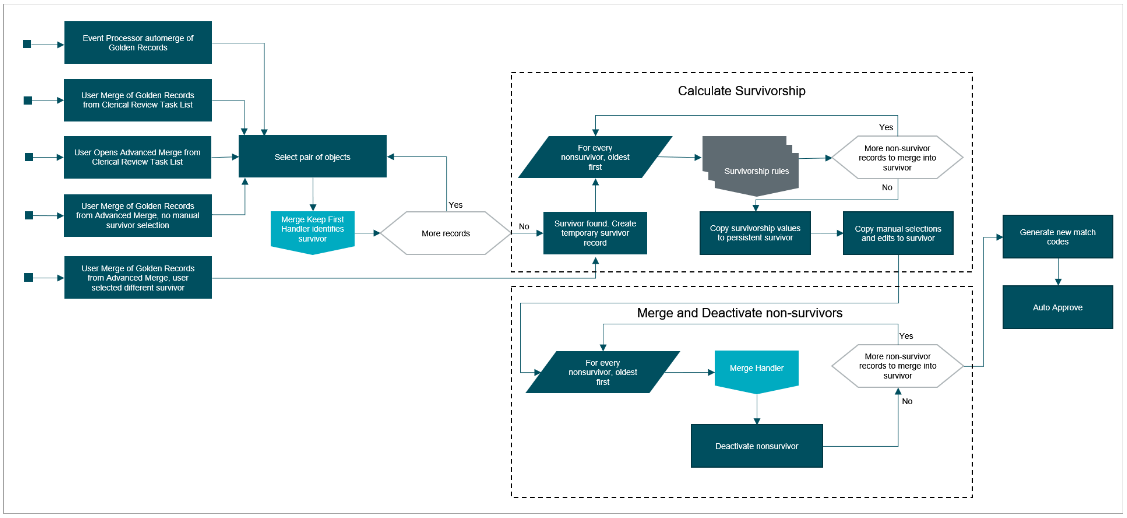
The standard version is invoked:

- Automatically by the event processor, when the match score of two potential duplicate records exceeds the auto merge threshold.
- Manually, when the user chooses the merge action directly on the clerical review task list.
- To populate the Advanced Merge UI with the initial merge result.

When clicking a merge button in the Advanced Merge dialog, the user can manually:

- Select a survivor which bypasses the Merge Keep First Handler survivor selection.
- Select surviving values from source records which are applied to the survivor.
- Add values that should survive, which are applied to the survivor.

View this topic in online help to explore this flowchart.


Event Processor automerge of Golden Records
User Merge of Golden Records from Clerical Review Task List
User Opens Advanced Merge from Clerical Review Task List
User Merge of Golden Records from Advanced Merge, no manual survivor selection
User Merge of Golden Records from Advanced Merge, user selected different survivor
Select pair of objects
Merge Keep First Handler identifies survivor
More records
Yes
No
Calculate Survivorship
For every nonsurvivor, oldest first
Survivorship rules
More non-survivor records to merge into survivor
Yes
No
Survivor found. Create temporary survivor record
Copy survivorship values to persistent survivor
Copy manual selections and edits to survivor
Merge and Deactivate non-survivors
For every nonsurvivor, oldest first
Merge Handler
More non-survivor records to merge into survivor
Yes
No
Deactivate nonsurvivor
Generate new match codes
Auto Approve

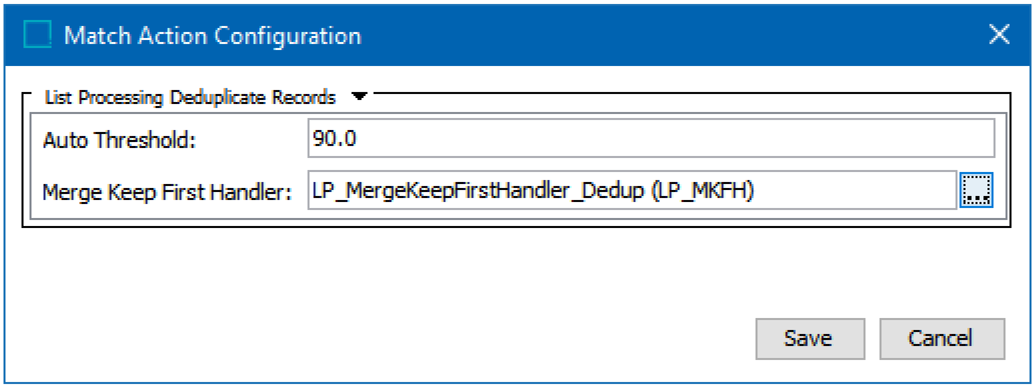
Survivorship rules are run on a temporary object which has some consequences when writing JavaScript survivorship rules. JavaScript survivorship rules must deal with references towards the survivor pointing to the permanent object, while updates from already-run survivorship rules are only available on the temporary object.

The **Deactivate nonsurvivor** step in the flow diagram performs the following actions:

- Removes all **Confirmed Duplicate** relations from the non-survivor
- Removes all **Confirmed Non-duplicate** relations from the non-survivor
- Deactivates the non-survivor Golden Record – that is, sets the deactivated attribute to deactivated. If using the starter package, deactivated corresponds to Value = "Yes" with Value ID = true.
- Re-targets references that pointed to the non-survivor to now point to the survivor
  - This can fail, for example, if the reference is a data container key on a data container, in which case the update could otherwise result in the survivor having several data containers with the same key.
  - The re-target may also discover the reference is already present, which results in no change.
- Ensures a major revision on the non-survivor and on the survivor, with appropriate revision comments describing the merge
- Adds **Merged Into Relation** from the non-survivor to survivor
- Removes unmerged from references from the non-survivor
- Copies non-survivor source information to the survivor
- Moves Source Record traceability tracking to the survivor
- Removes all Match Codes from the non-survivor. Match codes are never created on deactivated objects, ensuring they are not part of matching in the future.
- Adjusts clerical review workflow tasks to account for the deactivation. If the now-deactivated record was the last potential duplicate, the review task needs to be closed, even if the merge was initiated from Event Processor. This may also result in a new task being created for the survivor, and that could now match entirely new potential duplicates.
- Removes all match scores for the non-survivor

List Processing Deduplicate Records

The List Processing Deduplicate Records match action is used exclusively for removing duplicates during List Processing. Users can determine the matching threshold for when records are regarded as duplicates and merged into one record. A business condition can be used as the Merge Keep First Handler to determine if the first or the second duplicate will survive. If needed, survivorship rules configured after this match action can be used to ensure the survival of certain attribute values from non-surviving records.


Match Action Configuration
List Processing Deduplicate Records
Save
Cancel

Auto Threshold: 90.0

Merge Keep First Handler: LP_MergeKeepFirstHandler_Dedup (LP_MKFH)

For more information, refer to the **List Processing Remove Duplicates Operations** topic in the **Data Preparation** documentation.

Golden Records Survivorship Rules

Survivorship rules determine the outcome of merging two records by declaring which values survive for each attribute, reference, and data container on the golden record. The application of survivorship differs slightly across match actions, but the overall principles remain the same. When merging records, the surviving attribute values are selected by survivorship rules.

When merging two existing golden records – which can happen when updating information on one record results in both records being the same real-world object – one of the records must survive and the other must be deactivated. The default is to allow the record with the oldest STEP revision to persist and to deactivate the youngest record. This behavior can be overridden as defined in the **Creating a Merge Keep First Handler** topic.

When selecting which values survive, the basic strategy is often to either trust some sources more than other sources or to preserve the most recent updates. These kinds of rules are called **Most Recent** and **Trusted Source**. A set of general rules can be configured, but if special business logic is required, a business action survivorship rule can be implemented to apply surviving values to golden records.

Survivorship rules are defined independently for an object's name, references, data containers, and attributes / attribute groups. It is possible to apply different survivorship rules to groups of attributes or attributes and references individually, so that, for example, the value of one attribute follows a 'trusted source' rule while the value of other attributes follow a 'most recent' rule.

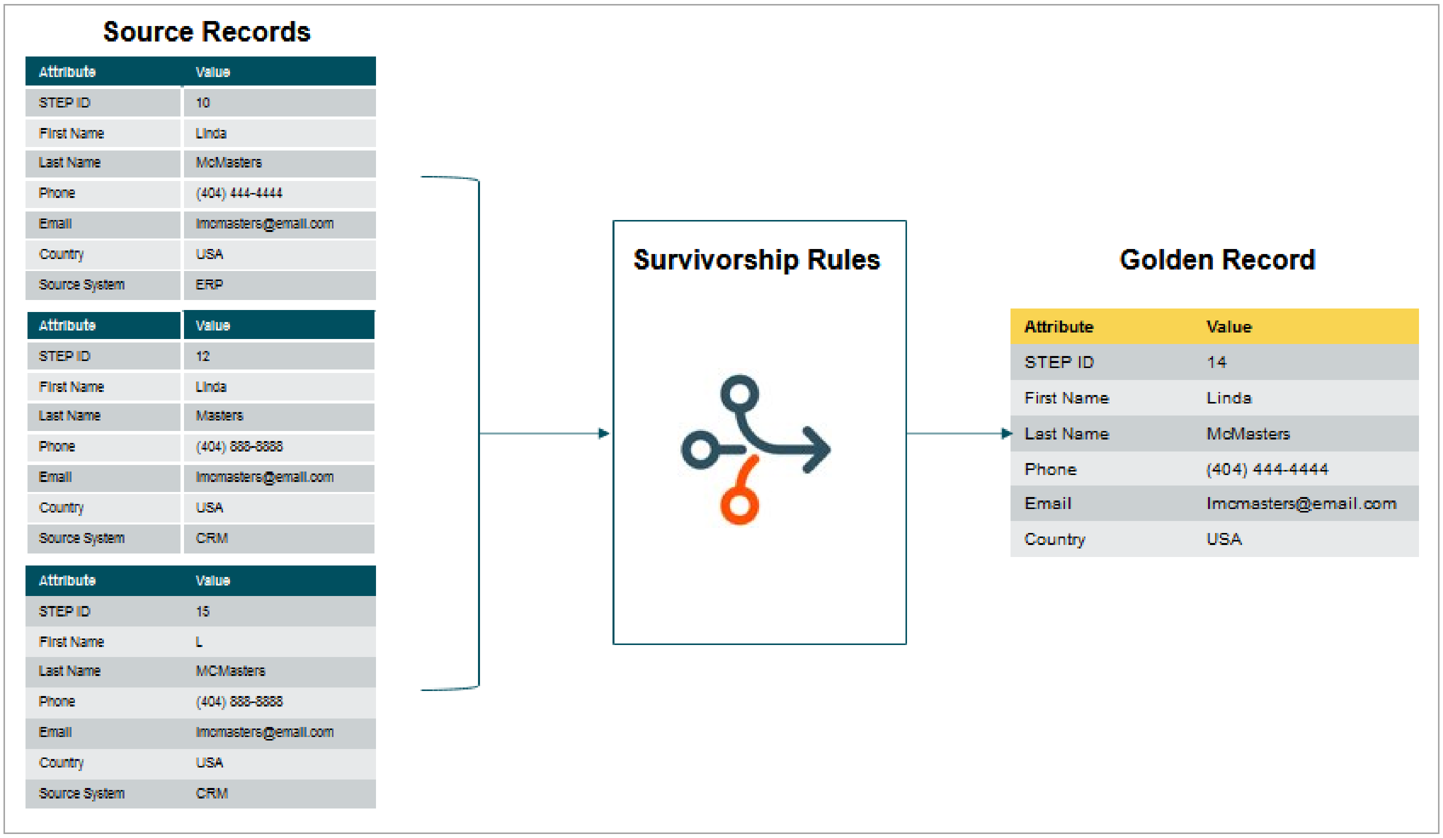
For more information, refer to the following topics:

- **Survivorship with Match and Link**
- **Survivorship with Match and Merge**
- **Configuring Survivorship Rules**

Survivorship in Match and Link

In a match and link solution, source records are products or entities that already exist in STEP. The golden record is a new product or entity, created and populated by the survivorship rules.

When survivorship rules run in a match and link solution, the number of sources is unknown; there could be one or many sources. This lack of information is especially important to remember if writing business action survivorship rules.



Source Records

Attribute	Value
STEP ID	10
First Name	Linda
Last Name	McMasters
Phone	(404) 444-4444
Email	lmcmasters@email.com
Country	USA
Source System	ERP

Attribute	Value
STEP ID	12
First Name	Linda
Last Name	Masters
Phone	(404) 888-8888
Email	lmcmasters@email.com
Country	USA
Source System	CRM

Attribute	Value
STEP ID	15
First Name	L
Last Name	MCMasters
Phone	(404) 888-8888
Email	lmcmasters@email.com
Country	USA
Source System	CRM

Golden Record

Attribute	Value
STEP ID	14
First Name	Linda
Last Name	McMasters
Phone	(404) 444-4444
Email	lmcmasters@email.com
Country	USA

Match and link survivorship rules are only ever run in the context of an event processor; they are not used when merging source records.

Golden records should not be merged in a match and link solution as that conflicts with the general rule that the golden record is not to be directly edited.

Trusted Source

To use the trusted source survivorship rule, information about the source, e.g., the object's originating system / supplier, must be available on the source objects. This attribute is defined in the general Matching component model as the 'Data Source Attribute.' Typically, this attribute is a mandatory LOV-based description attribute that does not allow users to add values. For more information, refer to the **Configuring Matching Component Model** topic.

Information from a source outside the list of trusted sources is not copied to the golden record during a trusted source survivorship rule evaluation. Information on a record without a source attribute is not copied to the golden record by trusted source survivorship rules.

For more information, refer to the **Configuring Survivorship Rules** topic.

Most Recent

The 'Most Recent' survivorship rule strategy takes the most recent data from a golden record's source objects.

The most recent can be qualified either by the revision date in STEP or by a 'Last Edited' date attribute. The date attribute option allows promotion of data based on the time of edit in source systems.

For more information, refer to the **Golden Records Survivorship Rules** topic.

Business Action Rule

Solutions commonly include special rules for survivorship that can be implemented via business actions that run as survivorship rules.

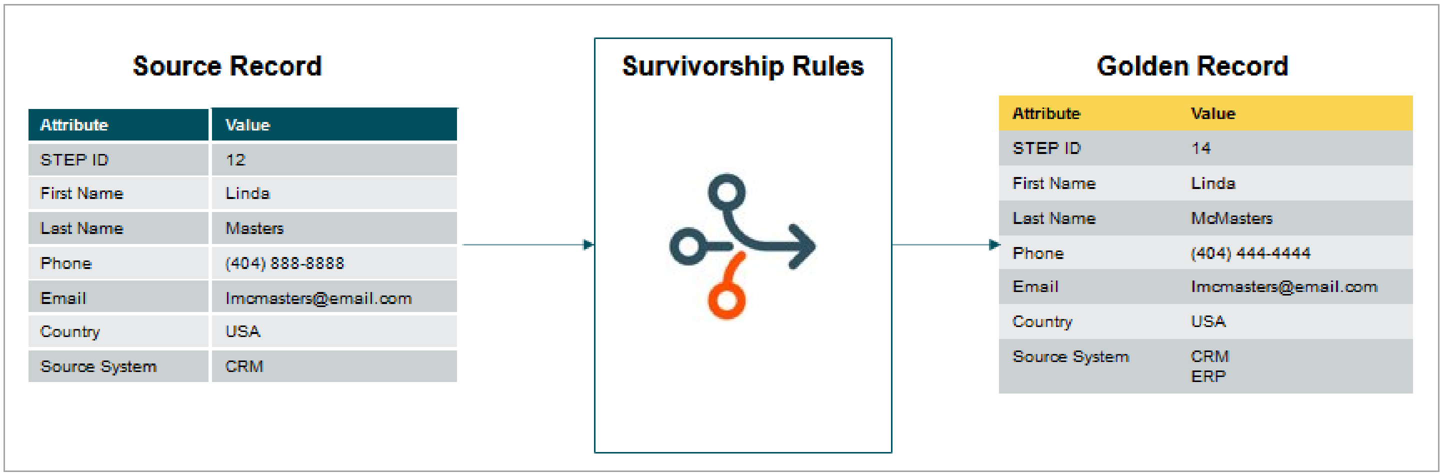
Note: A survivorship rule should never update values outside the golden record.

For more information, refer to the **Business Actions** topic in the **Business Rule** documentation.

Survivorship in Match and Merge

In match and merge, survivorship rules promote information from exactly one source to exactly one target by comparing information from the source with information from the target and writing the relevant updates to the target.

- In the match and merge IIEP and match and merge web service endpoint, information is promoted from incoming entities to existing or newly created golden records.
- In the matching event processing and in the clerical review Web UI, information is promoted from non-surviving golden records to surviving golden records as those records are merged.
- In the unmerge Web UI actions, as the association between source records and golden records are changed, the content of the resulting golden records is resolved.



Source Record

Attribute	Value
STEP ID	12
First Name	Linda
Last Name	Masters
Phone	(404) 888-8888
Email	lmcmasters@email.com
Country	USA
Source System	CRM

Golden Record

Attribute	Value
STEP ID	14
First Name	Linda
Last Name	McMasters
Phone	(404) 444-4444
Email	lmcmasters@email.com
Country	USA
Source System	CRM ERP

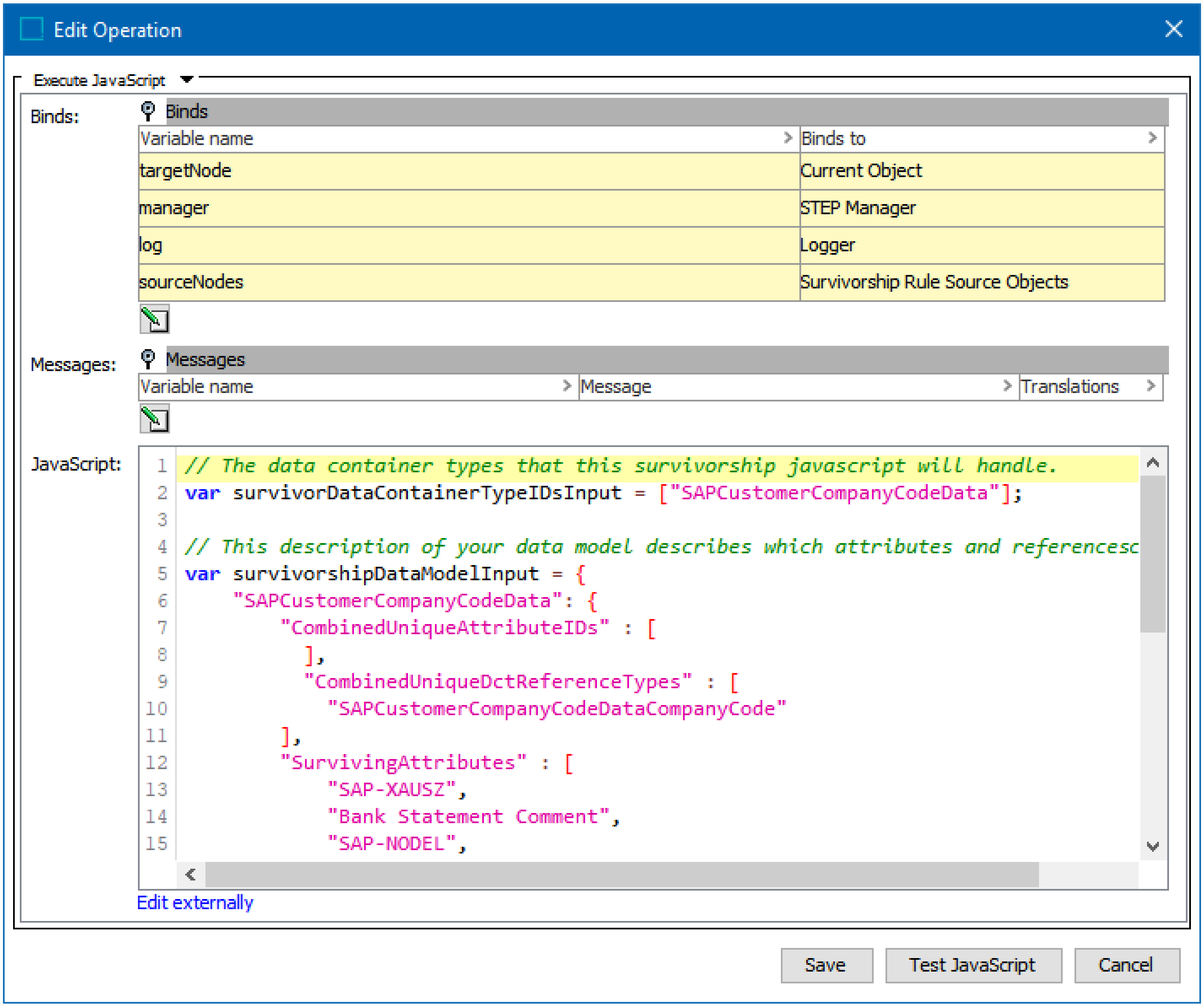
Keep in mind the difference between initial modifications of a golden record and an update to a golden record. An initial modification is when the source system supplies source records without knowing the golden record within STEP. This is an unconnected source, and it needs the normal trusted source priority to work. An update modification is when the source system supplies a source record while knowing which golden record to merge it with. This is a connected source, as it has picked up on a golden record feedback loop from STEP. In this case, these connected sources are treated equally.

Important: Survivorship on values for **Externally Maintained Attributes** is not recommended since survivorship logic depends on revision traceability. Externally maintained values may not figure correctly in the traceability view in Web UI, may be survived wrongly, and in some cases, this can lead to errors in survivorship rules when writing the values.

For more information, refer to the **Configuring Survivorship Rules** topic.

Survivorship in Match and Merge - Business Actions

Solutions commonly include special rules for survivorship and can be implemented via business actions that run as survivorship rules.



Variable name	Binds to
targetNode	Current Object
manager	STEP Manager
log	Logger
sourceNodes	Survivorship Rule Source Objects

Variable name	Message	Translations

```
// The data container types that this survivorship javascript will handle.
var survivorDataContainerTypeIDsInput = ["SAPCustomerCompanyCodeData"];

// This description of your data model describes which attributes and referencesc
var survivorshipDataModelInput = {
    "SAPCustomerCompanyCodeData": {
        "CombinedUniqueAttributeIDs" : [
          ],
          "CombinedUniqueDctReferenceTypes" : [
            "SAPCustomerCompanyCodeDataCompanyCode"
        ],
        "SurvivingAttributes" : [
            "SAP-XAUSZ",
            "Bank Statement Comment",
            "SAP-NODEL",
```

Note: A survivorship rule should only update values owned by the golden record itself. This includes attributes on the golden record, data containers on the golden record, and references from the golden record. This does not include references to the golden record, as reference are typically owned by their source node.

In match and merge, the survivorship rules always compare one source object against the golden record at a time. When merging multiple golden records from the clerical review task list, or unmerging multiple golden records in the unmerge screen, all survivorship rules are applied between the survivor and one source record before being compared to the next source record.

Business action survivorship rules in match and merge can use the following binds defined in the online help **Resource Materials** documentation:

- **Survivorship Rule Source Objects Bind** topic
- **Match and Merge Survivorship Context Bind** topic

For more information, refer to the **Business Actions** topic in the **Business Rules** documentation

Important: In match and merge survivorship rules, source records are often only available as non-persistent objects. Many of the operations available in the API are not applicable to non-persistent objects and will fail. Examples of operations that cannot be used successfully are approval and workflow-related operations. Operations related to reading and modifying attributes values, references, and data containers can be used successfully.

In match and merge, it is not possible to implement a trusted source pattern with the business action survivorship rule as the source information for an existing value on the golden record is not available in the JavaScript API.

Match and Merge Web Service Endpoint

When using a SOAP web service endpoint for a match and merge solution (defined in the **Web Service Endpoint - Match and Merge** topic of the **Data Exchange** documentation), JavaScript business actions can be used for survivorship rules. By default, when multiple JavaScript survivorship rules are to be run during matching, if a rule fails with an exception, rules that already completed without error are not rolled back and the rules following the one that failed are not run at all.

To change this functionality and ensure all changes are rolled back on a SOAP web service, a property can be added to the sharedconfig.properties file that sets parallel configuration to commit after each job. Because this setting can negatively impact performance, you must contact Stibo Systems Support for activation.

Survivorship in Match and Merge - Unmerge

Survivorship rules in unmerge run to:

- Suggest the values to survive on the golden record that were present before unmerge but exist after a number of sources have been removed from it.
- Suggest the values to survive on a new golden record created by unmerging a number of sources.
- Suggest the values to survive on a reactivated golden record after moving a number of sources to it.

For more information on unmerge, refer to **Match and Merge Clerical Review - Unmerge** topic.

Updating a Golden Record Created through Unmerging

The unmerge process is done when erroneously flagged duplicates are merged together. In this use case, the corrected golden record, created by removing the false sources, is updated based on the values selected for survivorship.

1. In the unmerge UI, a user removes a number of source records and golden records that do not belong to the record.
2. The algorithm removes values originating from the removed sources since those values no longer belong on the golden record.
3. The algorithm attempts to restore the cleaned values from revision history, applying the value as it was before it was set to the now cleaned value. This step does not happen for multivalued references and data containers.
4. Finally, the algorithm applies survivorship for all available source records to the golden record. These applications of survivorship rules will function as 'Match and Merge Survivorship update - when import merges with existing record.'

Using Survivorship Rules within the Unmerge Process

If using a golden record that was created from unmerging individual sources, the process uses survivorship rules like in the previous golden record updating scenario.

1. In the unmerge UI, the user removes a number of sources form a golden record to create this new golden record.
2. The unmerge algorithm sorts the source records associated with the new golden record by the time of editing the records and applies the changes, starting with the oldest source.
3. The survivorship of the oldest source, when applied, works like the 'Match and Merge Survivorship when Import creates new record' operation.
4. The newer source records, when applied, work like 'Match and Merge Survivorship update - when import merges with existing record' operation.

Unmerging a Golden Record from Another Golden Record

1. In the unmerge UI, the user removes a falsely merged golden record from another golden record using the unmerge UI.
2. The unmerged golden record is reactivated and it is assumed to have the attribute values it had when it was merged.
3. The algorithm removes any values that originated from removed sources since those values no longer belong on the reactivated golden record.
4. The algorithm attempts to restore the cleaned values from revision history, applying the value as it was before it was set to the now cleaned value. This step does not happen for multivalued references and data containers.
5. The algorithm applies survivorship rules for all available source records to the golden record. The application of survivorship rules functions as 'Match and Merge Survivorship update - when import merges with existing record' operation.

Configuring Survivorship Rules

Survivorship rules are stored on a matching algorithm and define which source data is added to the golden record for the selected merging or linking solution. For more information, refer to the **Golden Records Survivorship Rules** topic.

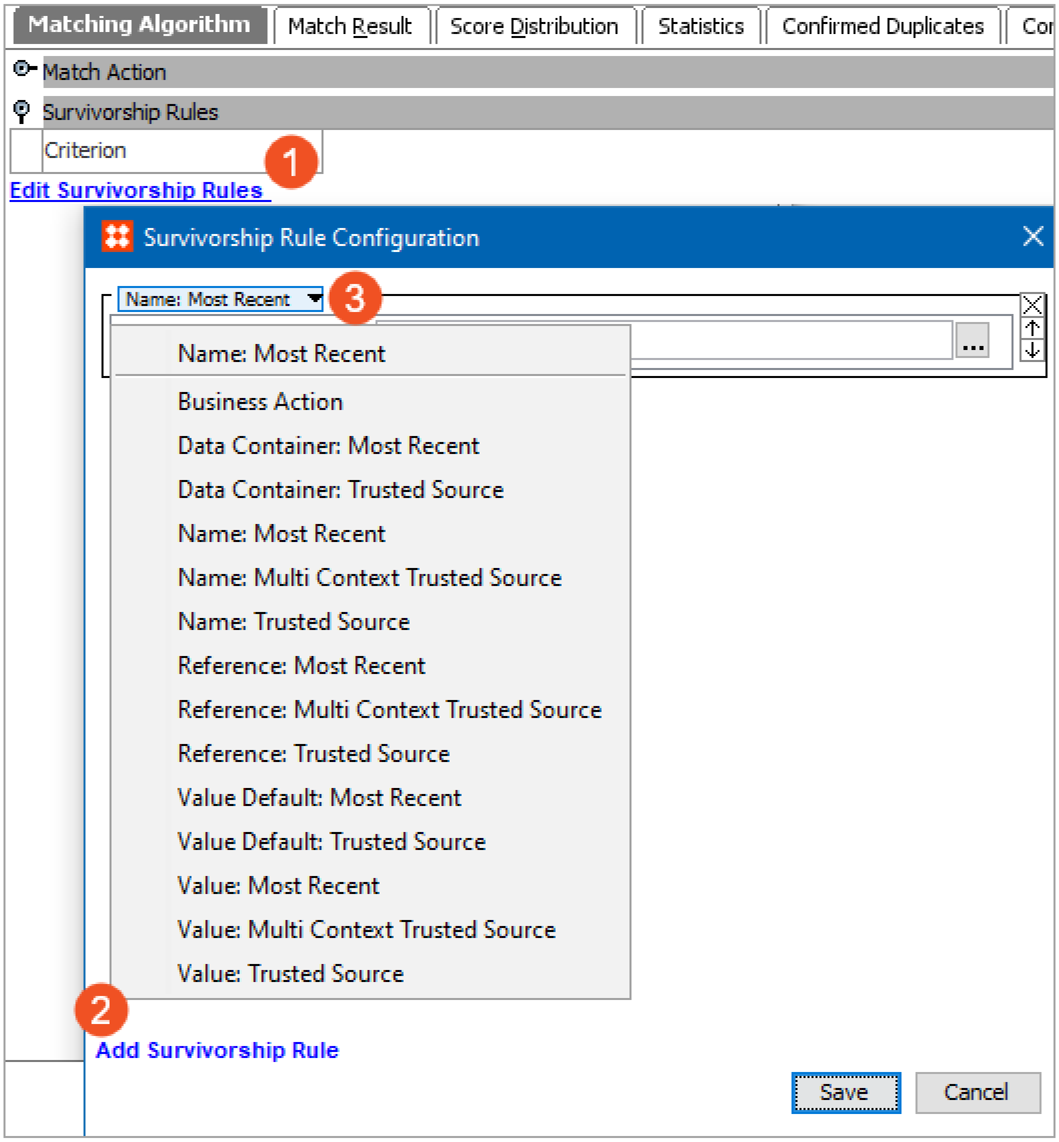
For details on how to the ensure the most trusted and up-to-date values survive, review the **Considerations** section below.

Configuration

To configure a survivorship rule:

1. On the Matching Algorithm tab, open the 'Survivorship Rules' flipper and click the **Edit Survivorship Rules** link.

2. On the 'Survivorship Rule Configuration' dialog, click the **Add Survivorship Rule** link, and select the required rule from the dropdown.



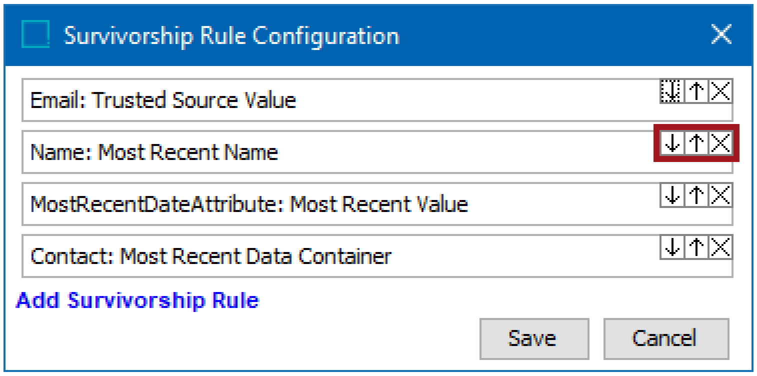
3. Provide the required parameter information for the selected rule. Parameter details are included in the following topics:

   - Business Action Rule
   - Data Container Rules
   - Name Rules
   - Reference Rules
   - Value Rules

   Additionally, for both 'match and link' and 'match and merge' solutions, review the **Creating a Merge**

Keep First Handler topic for directions to override default behavior on the surviving record.

4. If necessary, edit the order of the rules which are executed in order from top-to-bottom.
   - Click the down and up buttons to change the order of the rules.
   - Click the **X** button to remove a rule.


Survivorship Rule Configuration

Email: Trusted Source Value

Name: Most Recent Name

MostRecentDateAttribute: Most Recent Value

Contact: Most Recent Data Container

For more information on matching algorithms, refer to the **Configuring Matching Algorithms** topic.

Considerations

For most survivorship rules, understanding of the time aspect is important. For Trusted Source, you still need to know which of the values from the most trusted source is the most up-to-date value.

> **Important:** Survivorship on values for **Externally Maintained Attributes** is not recommended since survivorship logic depends on revision traceability. Externally maintained values may not figure correctly in the traceability view in Web UI, may be survived wrongly, and in some cases, this can lead to errors in survivorship rules when writing the values.

- **Trusted source** survivorship rules trust some source systems over others. The rule is configured with a list of the available source systems in the sequence of trust. The systems with lower trust rankings do not overwrite values set by higher trust systems.

  > **Note:** Since STEP is considered the ultimate trusted source for trusted source rules, manual edits on a golden record are never overwritten.

  The source system information is an integral part of match and merge and is defined in the component model. For more information on how source information is tracked in match and merge, refer to the **Match and Merge Traceability** topic.

  > **Important:** Information from a source outside the list of trusted sources is regarded as untrusted and as such, that information is not copied to the golden record during trusted source survivorship.

For match and merge trusted source, when the value on a trusted source is deleted:

  - The trusted value is not deleted from the golden record.
  - The value from a lesser trusted source is not applied to the golden record.
  - Values from lesser trusted sources are not available during the survivorship evaluation.

- **Most Recent** survivorship rule strategy lets the most recent data from all contributing records survive to the final golden record and can be qualified either by the revision date in STEP or by a 'Last Edited' date attribute.

  - Using a 'Last Edited' date attribute makes it possible to promote data based on the time of edit in source systems. For an Attribute Group survivorship rule, the Last Edit Date attribute cannot be part of the attribute group.
  - When no 'Last Edited' date attribute is selected in a survivorship rule, the STEP revision date is used.

  For manual edits, the revision date is always used to determine the most recent value. This logic applies to the object name, attribute values, references, data containers, attribute values on data containers, attribute values on references, and references on data containers.

  In the 'Match and Merge Importer' IIEP as well as in the 'Match and Merge Web Service Endpoint,' the deletion of attribute values on existing golden records can be promoted by sending an empty value element in the STEP XML. For example, the following STEPXML would void the 'FirstName' attribute value:

  ```
  <Value AttributeID=”FirstName”></Value>
  ```

Survivorship Business Action Rule

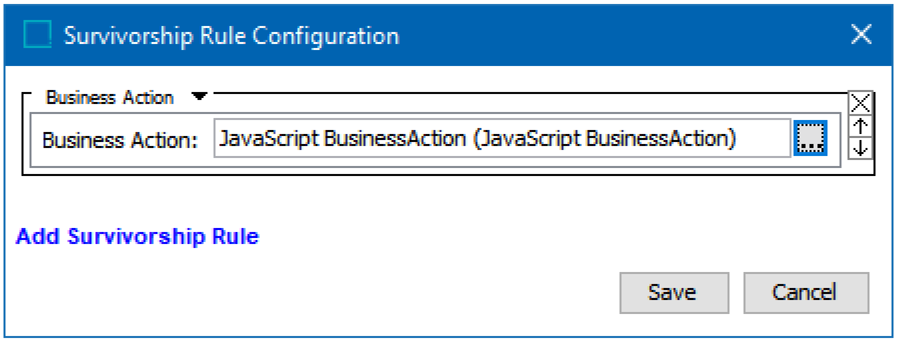
On a matching algorithm, the following rule is available to define promotion to a golden record via a business action.

Important: For both a match and merge solution or a match and link solution, survivorship rules should only update the surviving entity and not other entities referencing the survivor.

Business Action

Valid for strategies: merge or link

Specifies the business action used to promote data to the golden record. Data is promoted to a golden record across all contexts (the evaluation is performed in the context and workspace selected on the algorithm) and only the data for which survivorship rules exist will be promoted. Inherited and calculated values are not used.



Business Action: JavaScript BusinessAction (JavaScript BusinessAction)

Business Action - Click the ellipsis button (**...**) to specify a business action to run on golden records when survivorship rules are applied.

When using a JavaScript business action with the merge golden record solution, you must use the 'Survivorship Rules Source Object' bind. This bind grants the script access to the temporary source objects so that relevant values can be promoted from them to the surviving golden records. For more information, refer to the **Survivorship Rule Source Objects Bind** topic in the online help **Resource Materials** documentation.

Note: For JavaScript survivorship rules, if the source has a reference to itself, that reference has already had its target moved to the surviving record before survivorship rules are run.

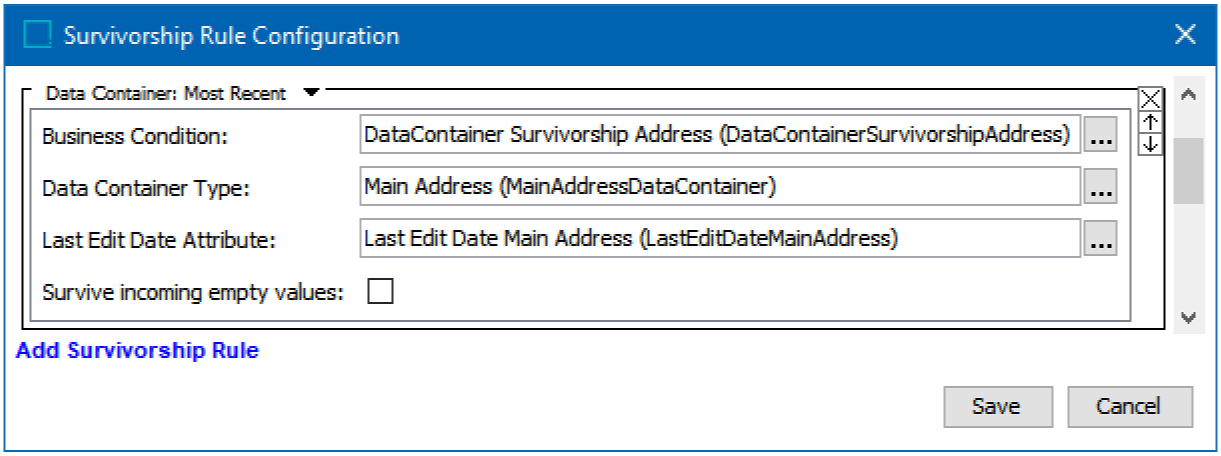
Survivorship Data Container Rules

On a matching algorithm, the following rules are available for promoting data container values to a golden record.

Data Container: Most Recent

Valid for strategies: merge or link

Specifies that the most recent data container instances and their attribute values are promoted to the golden records. The analysis is performed in the single context / workspace selected in the algorithm, and that data is promoted across all contexts / qualifiers.



Business Condition: DataContainer Survivorship Address (DataContainerSurvivorshipAddress)

Data Container Type: Main Address (MainAddressDataContainer)

Last Edit Date Attribute: Last Edit Date Main Address (LastEditDateMainAddress)

Survive incoming empty values: □

- **Business Condition** - The business condition is used on data containers to determine if the source data container instance represents an update to one of the existing target data containers. If no Data Container Key is configured, click the ellipsis button (...) and select a business condition that is valid for the golden record object type.
  - If the source record should always overwrite the golden record, the condition must return true.
  - Otherwise, this condition must be a JavaScript rule that uses the 'Pairs of Attributes' bind to compare data container instances on source records with data container instances on golden records when survivorship rules are applied. For more information and an example of the bind, refer to the **Pair of Attribute Values Bind** topic in the online help **Resource Materials** documentation.

Note: There is no need to configure a business condition if the data container type being merged has a configured Data Container Key. For more information on data container keys, refer to the **Data Container Keys** topic in the **System Setup** documentation.

- **Data Container Type:** Click the ellipsis button (...) and select the relevant data container type.
- **Last Edit Date Attribute -** When no attribute is selected, the most recent date is the STEP object revision timestamp when the given element of the survivorship rule entered STEP.

  Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

  - When the selected attribute is valid for this object, timestamp is taken from the object.
  - When the selected attribute is not valid for the object, the value is taken from the given element of the survivorship rule, for example, a data container object or a reference object.

> **Note:** Survivorship rules consider Last Edit Date attributes on the entities before considering Last Edit Date on the attributes within a data container. Additionally, for multi-value data container types, the newest date from all data containers of the specified type is considered.

- **Survive incoming empty values -** When selected, an imported empty value replaces an existing empty value in the data container. For example, the phone data container for a record has PhoneType value of 'Private Phone', PhoneNumber value of '555-8637', and the LastEditDatePhone value of '2021-06-21'. Importing the following XML:

```
<DataContainer>
  <Values>
    <Value AttributeID="PhoneType"></Value>
    <Value AttributeID="PhoneNumber">555-8637</Value>
    <Value AttributeID="LastEditDatePhone">2021-11-13 15:00:00</Value>
  </Values>
</DataContainer>
```

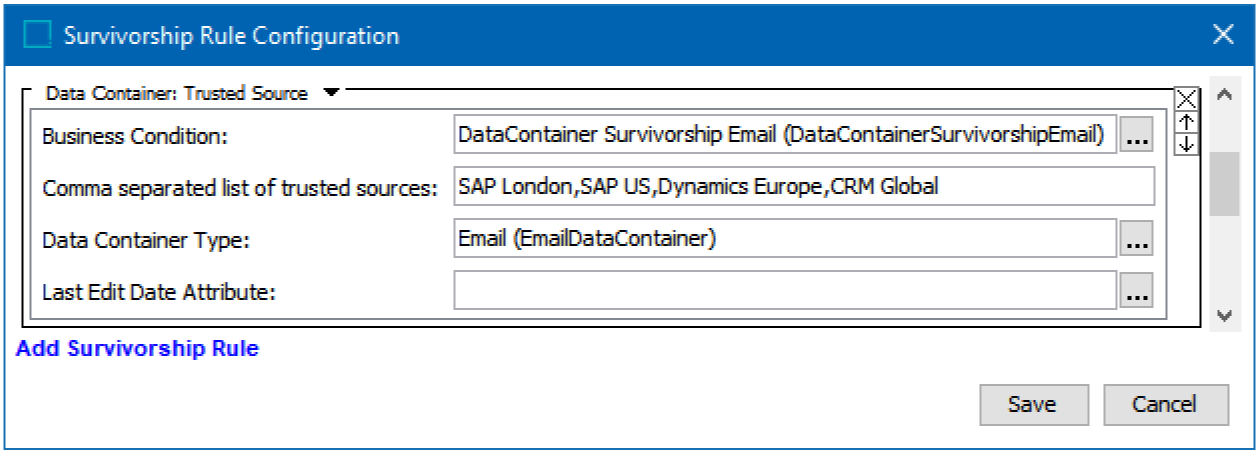
  results in the following outcome based on the checkbox setting:

  - Survive incoming empty values = checked, the phone data container PhoneType attribute is updated to blank (the empty value survives) and the LastEditDatePhone attribute value is updated to 2021-11-13 15:00:00.
  - Survive incoming empty values = not checked, the phone data container PhoneType attribute is not updated (the previous value 'Private Phone' remains) but the LastEditDatePhone is updated to 2021-11-13 15:00:00.

Data Container: Trusted Source

Valid for strategies: merge or link

Specifies data container instances and their attribute values that originate from the specified trusted source(s) are promoted to the golden records. The analysis is performed in the single context / workspace selected in the algorithm, and that data is promoted across all contexts / qualifiers.


Survivorship Rule Configuration
Data Container: Trusted Source

Business Condition: DataContainer Survivorship Email (DataContainerSurvivorshipEmail)

Comma separated list of trusted sources: SAP London,SAP US,Dynamics Europe,CRM Global

Data Container Type: Email (EmailDataContainer)

Last Edit Date Attribute:

Add Survivorship Rule
Save
Cancel

- **Business Condition** - If no Data Container Key is configured, click the ellipsis button (**...**) and select a business condition that is valid for the golden record object type.
  - If the source record should always overwrite the golden record, the condition must return true.
  - Otherwise, this condition must be a JavaScript rule that uses the 'Pairs of Attributes' bind to compare data container instances on source records with data container instances on golden records when survivorship rules are applied. For more information and an example of the bind, refer to the **Pair of Attribute Values Bind** topic in the online help **Resource Materials** documentation.

> **Note:** There is no need to configure a business condition if the data container type being merged has a configured Data Container Key. For more information on data container keys, refer to the **Data Container Keys** topic in the **System Setup** documentation.

- **Comma separated list of trusted sources** - Enter a comma-separated list of the case-sensitive Source System ID for all trusted sources, starting with the most trusted source, then the next-most, and so on. Content is taken from the first trusted source with data. If content does not exist for any of the trusted sources, nothing is promoted to the golden record and the existing golden record value is cleaned. For information on the Source System ID Attribute setting, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.
- **Data Container Type:** Click the ellipsis button (**...**) and select the relevant data container type.
- **Last Edit Date Attribute** - When no attribute is selected, the most recent date is the STEP object revision timestamp when the given element of the survivorship rule entered STEP.

  Optionally, click the ellipsis button (**...**) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

- When the selected attribute is valid for this object, timestamp is taken from the object.
- When the selected attribute is not valid for the object, the value is taken from the given element of the survivorship rule, for example, a data container object or a reference object.

> **Note:** Survivorship rules consider Last Edit Date attributes on the entities before considering Last Edit Date on the attributes within a data container. Additionally, for multi-value data container types, the newest date from all data containers of the specified type is considered.

Data Containers with Inconsistent Keys

Consider the following principles when configuring survivorship rules to account for data containers with inconsistent keys:

- Data containers with inconsistent keys do not survive a merge using standard survivorship rules, even if the key is incomplete or duplicated.
- When updating a target with a duplicate key, the data container with the lowest internal STEP ID survives.
- Survivorship rules cannot write an incomplete key.
- Survivorship rules cannot add a data container instance with a duplicated key.

For more information on data container keys, refer to the **Data Container Keys** topic in the **System Setup** documentation.

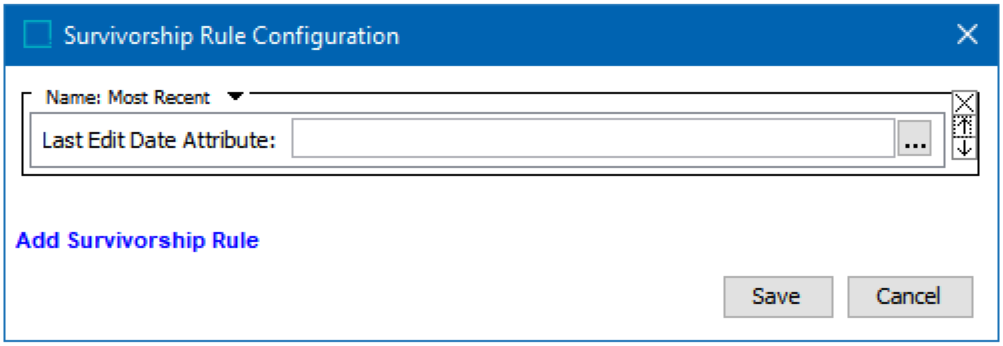
Survivorship Name Rules

On a matching algorithm, the following rules are available for promoting an object name to a golden record.

Name: Most Recent

Valid for strategies: merge or link

Specifies that 'Name' is taken from the source object with the most recent name. The analysis is performed in the single context / workspace selected in the algorithm, and that data is promoted across all contexts / qualifiers.



Survivorship Rule Configuration

Name: Most Recent

Last Edit Date Attribute:

- **Last Edit Date Attribute** - When no attribute is selected, the most recent date is the STEP object revision timestamp when the given element of the survivorship rule entered STEP.

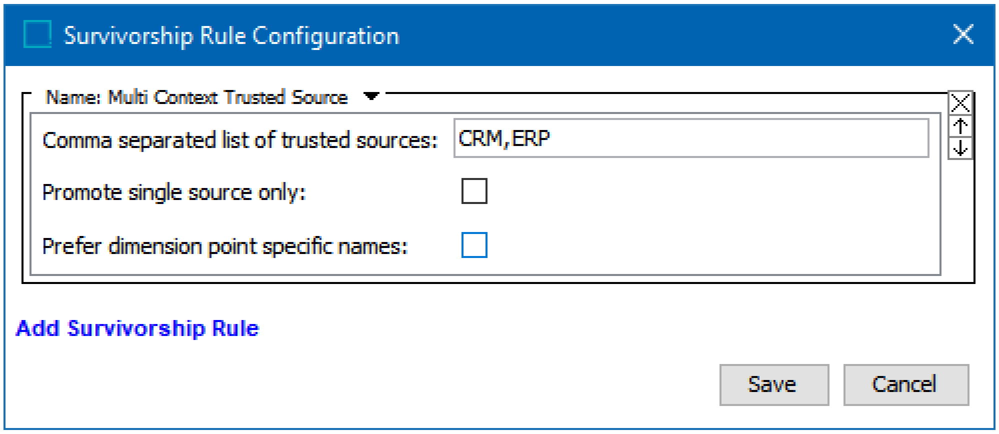
  Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

  - When the selected attribute is valid for this object, timestamp is taken from the object.
  - When the selected attribute is not valid for the object, the value is taken from the given element of the survivorship rule, for example, a data container object or a reference object.

Name: Multi Context Trusted Source

Valid for strategies: link only

Specifies that the name promoted to the golden record is from the most trusted source and considers data that is dimension dependent. The analysis is performed for all contexts / qualifiers (a set of one or more dimension points, like country and language) in STEP.



Survivorship Rule Configuration

Name: Multi Context Trusted Source

Comma separated list of trusted sources: CRM,ERP

Promote single source only: □

Prefer dimension point specific names: □

- **Comma separated list of trusted sources -** Enter a comma-separated list of the case-sensitive Source System ID for all trusted sources, starting with the most trusted source, then the next-most, and so on. Content is taken from the first trusted source with data. If content does not exist for any of the trusted sources, nothing is promoted to the golden record and the existing golden record value is cleaned. For information on the Source System ID Attribute setting, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.
- **Promote single source only -** When checked, content from the most trusted source is used for all contexts / qualifiers, which prevents empty values in the golden record as long as one of the trusted sources has content. For example, when only the French language / France country context has a value, that value would be written into other contexts that are blank.

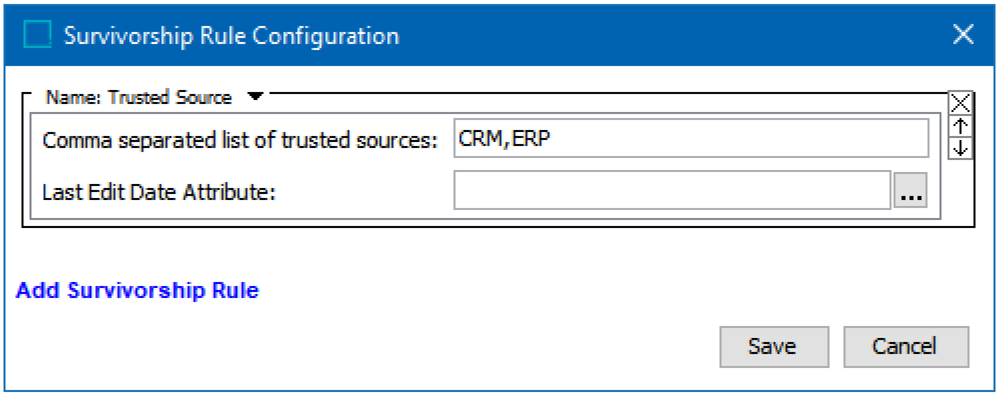
  When not checked, each context / qualifier supplies its own content, including empty values when found.
- **Prefer dimension point specific names -** When checked, only a local name is promoted.

  When not checked, available inherited content is promoted if a local name does not exit.

Name: Trusted Source

Valid for strategies: merge or link

Specifies that 'name' is taken from the most trusted source. The analysis is performed in the single context / workspace selected in the algorithm, and that data is promoted across all contexts / qualifiers.



Survivorship Rule Configuration

Name: Trusted Source

Comma separated list of trusted sources: CRM,ERP

Last Edit Date Attribute:

- **Comma separated list of trusted sources** - Enter a comma-separated list of the case-sensitive Source System ID for all trusted sources, starting with the most trusted source, then the next-most, and so on. Content is taken from the first trusted source with data. If content does not exist for any of the trusted sources, nothing is promoted to the golden record and the existing golden record value is cleaned. For information on the Source System ID Attribute setting, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.

- **Last Edit Date Attribute** - When no attribute is selected, the most recent date is the STEP object revision timestamp when the given element of the survivorship rule entered STEP.

  Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

  - When the selected attribute is valid for this object, timestamp is taken from the object.

  - When the selected attribute is not valid for the object, the value is taken from the given element of the survivorship rule, for example, a data container object or a reference object.

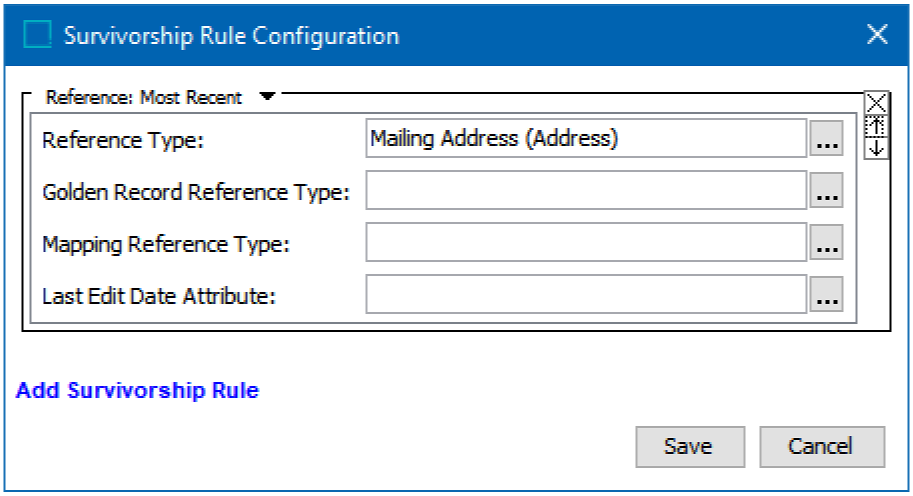
Survivorship Reference Rules

On a matching algorithm, the following rules are available for promoting references / links to a golden record.

Reference: Most Recent

Valid for strategies: merge or link

Specifies that the reference / link types promoted from the source object is the most recent reference / link. The analysis is performed in the single context / workspace selected in the algorithm, and that data is promoted across all contexts / qualifiers.



Reference Type: Mailing Address (Address)

Golden Record Reference Type:

Mapping Reference Type:

Last Edit Date Attribute:

- **Reference Type:** Required. Click the ellipsis button (...) to specify the valid reference / link type from the source objects you are handling. When this is the only field populated, a reference / link of the same type pointing to the same target will be promoted to the golden record.
- **Golden Record Reference Type -** Optionally, click the ellipsis button (...) to specify the reference type that links the target golden records and target source objects.

  If the objects that the source objects are pointing to also have golden records, you can configure the new golden record to point to this golden record rather than the source object's original target.
- **Mapping Reference Type -** Optionally, click the ellipsis button (...) to specify a reference / link type mapped to this reference / link type.

  When this parameter is not populated, the reference or link created for the golden record is of the same type as the source object's reference / link.

- **Last Edit Date Attribute -** When no attribute is selected, the most recent date is the STEP object revision timestamp when the given element of the survivorship rule entered STEP.

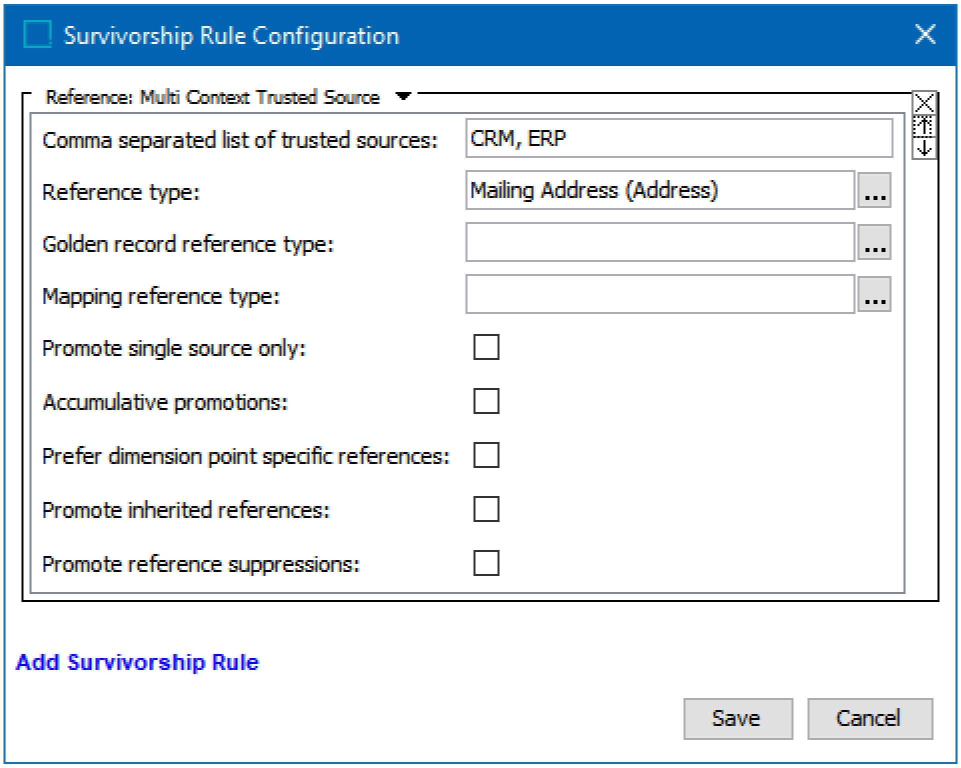
  Optionally, click the ellipsis button (**...**) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

  - When the selected attribute is valid for this object, timestamp is taken from the object.
  - When the selected attribute is not valid for the object, the value is taken from the given element of the survivorship rule, for example, a data container object or a reference object.

Reference: Multi Context Trusted Source

Valid for strategies: link

Specifies that the reference / link types promoted from the source object is the trusted source and considers data that is dimension dependent. The analysis is performed for all contexts / qualifiers (a set of one or more dimension points, like country and language) in STEP.


Survivorship Rule Configuration
Reference: Multi Context Trusted Source
Add Survivorship Rule
Save
Cancel

Comma separated list of trusted sources:	CRM, ERP
Reference type:	Mailing Address (Address) ...
Golden record reference type:	...
Mapping reference type:	...
Promote single source only:	□
Accumulative promotions:	□
Prefer dimension point specific references:	□
Promote inherited references:	□
Promote reference suppressions:	□

- **Comma separated list of trusted sources -** Enter a comma-separated list of the case-sensitive Source System ID for all trusted sources, starting with the most trusted source, then the next-most, and so on. Content is taken from the first trusted source with data. If content does not exist for any of the trusted sources, nothing is promoted to the golden record and the existing golden record value is cleaned. For information on the Source System ID Attribute setting, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.

- **Reference Type:** Required. Click the ellipsis button (...) to specify the valid reference / link type from the source objects you are handling. When this is the only field populated, a reference / link of the same type pointing to the same target will be promoted to the golden record.
- **Golden Record Reference Type** - Optionally, click the ellipsis button (...) to specify the reference type that links the target golden records and target source objects.

  If the objects that the source objects are pointing to also have golden records, you can configure the new golden record to point to this golden record rather than the source object's original target.
- **Mapping Reference Type** - Optionally, click the ellipsis button (...) to specify a reference / link type mapped to this reference / link type.

  When this parameter is not populated, the reference or link created for the golden record is of the same type as the source object's reference / link.
- **Promote single source only -** When checked, content from the most trusted source is used for all contexts / qualifiers, which prevents empty values in the golden record as long as one of the trusted sources has content. For example, when only the French language / France country context has a value, that value would be written into other contexts that are blank.

  When not checked, each context / qualifier supplies its own content, including empty values when found.
- **Accumulative promotions -** When checked, all references / links and their metadata from multiple source records are written to the golden record (including references from different contexts). In a case of a reference / link being single-valued with multiple references in a particular context, the most trusted will be promoted, but if there are multiple references in multiple contexts, all references will be promoted to golden record.

  When not checked, only references / links and their metadata from the most trusted source records are written to the golden record.

  > **Note:** If both the 'Accumulative promotions' and the 'Promote single source only' options are checked, then 'Promote single source only' takes precedence, and only references / links from that source are promoted.
- **Prefer dimension point specific references -** When checked, only local references / links are promoted.

  When not checked, available inherited content is promoted if a local reference / link does not exit.

  This option can be used in conjunction with the 'Accumulative promotions' option to determine which reference / link to promote when multiple source records have references / links to the same target object.

  > **Note:** If both the 'Prefer dimension point specific references' and the 'Promote single source only' options are checked, then 'Promote single source only' takes precedence, and only references from that source are promoted.

- **Promote inherited references -** When checked, inherited references / links are written to the golden record only if the golden record object type is valid for the selected reference type.

  When not checked, only local references are written to the golden record.

- **Promote reference suppressions -** When checked, suppressed references / links are written to the golden record.

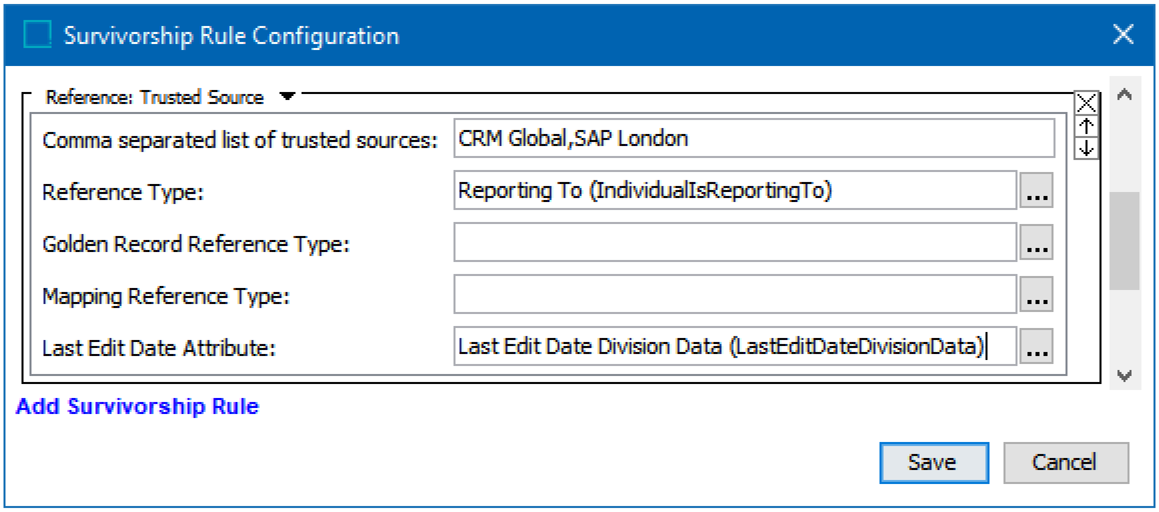
  When not checked, suppressed references / links are ignored.

Reference: Trusted Source

Valid for strategies: merge or link

Specifies that the reference / link types promoted from the source object is the most trusted reference / link. The analysis is performed in the single context / workspace selected in the algorithm, and that data is promoted across all contexts / qualifiers.

In a match and merge solution, when a source system updates a record in STEP, the trusted source picks up on the golden record feedback loop and serves as a connected source. For more information on connected sources, refer to the **Survivorship in Match and Merge** topic.



Comma separated list of trusted sources:	CRM Global,SAP London
Reference Type:	Reporting To (IndividualIsReportingTo)
Golden Record Reference Type:	
Mapping Reference Type:	
Last Edit Date Attribute:	Last Edit Date Division Data (LastEditDateDivisionData)

- **Comma separated list of trusted sources -** Enter a comma-separated list of the case-sensitive Source System ID for all trusted sources, starting with the most trusted source, then the next-most, and so on. Content is taken from the first trusted source with data. If content does not exist for any of the trusted sources, nothing is promoted to the golden record and the existing golden record value is cleaned. For information on the Source System ID Attribute setting, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.

- **Reference Type:** Required. Click the ellipsis button (...) to specify the valid reference / link type from the source objects you are handling. When this is the only field populated, a reference / link of the same type pointing to the same target will be promoted to the golden record.

- **Golden Record Reference Type -** Optionally, click the ellipsis button (...) to specify the reference type that links the target golden records and target source objects.

  If the objects that the source objects are pointing to also have golden records, you can configure the new golden record to point to this golden record rather than the source object’s original target.

- **Mapping Reference Type -** Optionally, click the ellipsis button (...) to specify a reference / link type mapped to this reference / link type.

  When this parameter is not populated, the reference or link created for the golden record is of the same type as the source object's reference / link.

- **Last Edit Date Attribute -** When no attribute is selected, the most recent date is the STEP object revision timestamp when the given element of the survivorship rule entered STEP.

  Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

  - When the selected attribute is valid for this object, timestamp is taken from the object.
  - When the selected attribute is not valid for the object, the value is taken from the given element of the survivorship rule, for example, a data container object or a reference object.

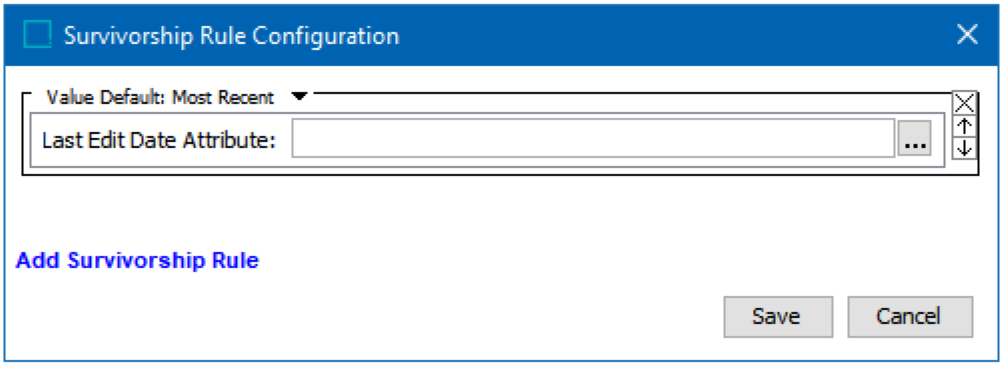
Survivorship Value Rules

On a matching algorithm, the following rules are available for promoting values to a golden record.

Value Default: Most Recent

Valid for strategies: merge or link

Specifies that the value is taken from the source with the most recent date for all attributes. The analysis is performed in the single context / workspace selected in the algorithm, and that data is promoted across all contexts / qualifiers.


Survivorship Rule Configuration
Value Default: Most Recent
Last Edit Date Attribute:
...
Add Survivorship Rule
Save
Cancel

Last Edit Date Attribute - When no attribute is selected, the most recent date is the STEP object revision timestamp when the given element of the survivorship rule entered STEP.

Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

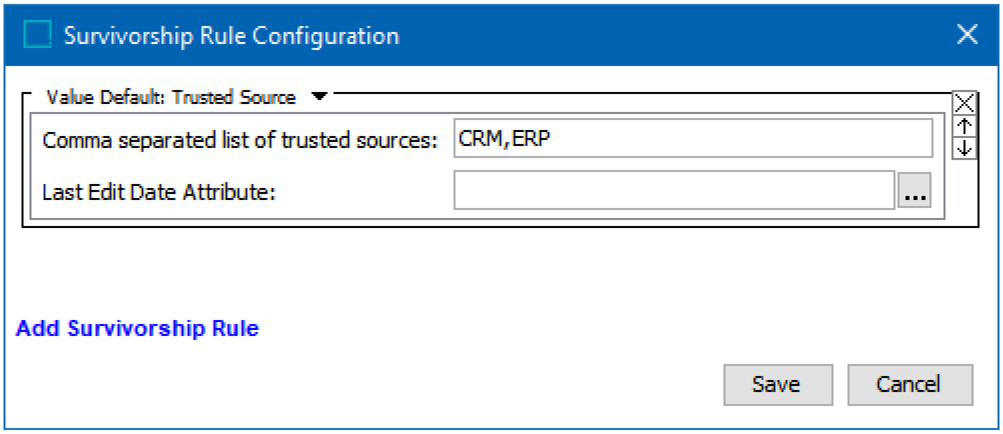
- When the selected attribute is valid for this object, timestamp is taken from the object.
- When the selected attribute is not valid for the object, the value is taken from the given element of the survivorship rule, for example, a data container object or a reference object.

Note: Attributes, used by live Unique Keys, will not be updated by Value Default survivorship rules

Value Default: Trusted Source

Valid for strategies: merge or link

Specifies that the value is taken from the most trusted source for all attributes. The analysis is performed in the single context / workspace selected in the algorithm, and that data is promoted across all contexts / qualifiers.



Survivorship Rule Configuration
Value Default: Trusted Source

Comma separated list of trusted sources: CRM,ERP

Last Edit Date Attribute:

...
Add Survivorship Rule
Save
Cancel

- **Comma separated list of trusted sources -** Enter a comma-separated list of the case-sensitive Source System ID for all trusted sources, starting with the most trusted source, then the next-most, and so on. Content is taken from the first trusted source with data. If content does not exist for any of the trusted sources, nothing is promoted to the golden record and the existing golden record value is cleaned. For information on the Source System ID Attribute setting, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.
- **Last Edit Date Attribute -** When no attribute is selected, the most recent date is the STEP object revision timestamp when the given element of the survivorship rule entered STEP.

  Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

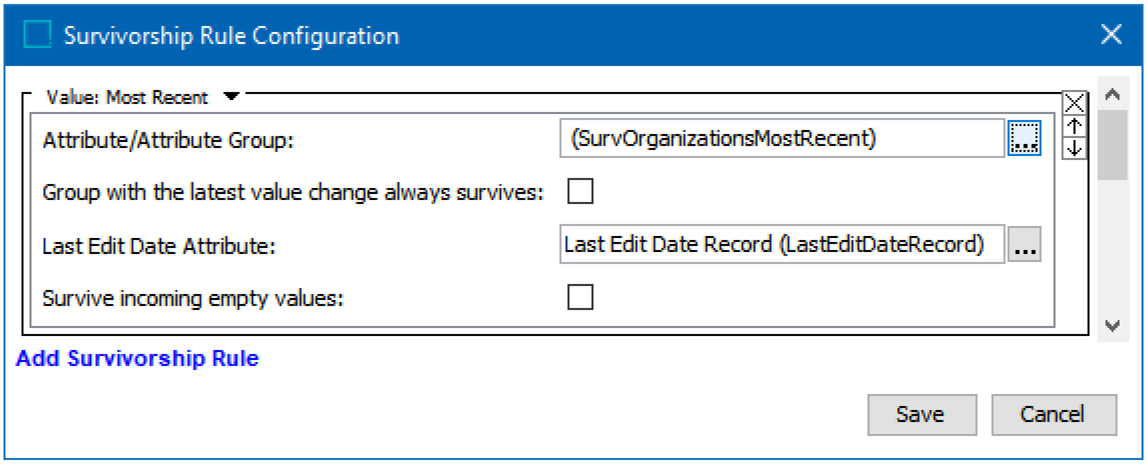
  - When the selected attribute is valid for this object, timestamp is taken from the object.
  - When the selected attribute is not valid for the object, the value is taken from the given element of the survivorship rule, for example, a data container object or a reference object.

Note: Attributes, used by live Unique Keys, will not be updated by Value Default survivorship rules

Value: Most Recent

Valid for strategies: merge or link

Specifies that value is taken from the source object with the most recent value. The analysis is performed in the single context / workspace selected in the algorithm, and that data is promoted across all contexts / qualifiers.



Attribute/Attribute Group: (SurvOrganizationsMostRecent)

Group with the latest value change always survives: □

Last Edit Date Attribute: Last Edit Date Record (LastEditDateRecord)

Survive incoming empty values: □

- **Attribute / Attribute Group** - Click the ellipsis button (...) and select a single attribute or all attributes in a specific group for which the rule applies.
- **Group with the latest value change always survives** - When checked, all values of an attribute group will survive when the group contains the attribute with the most recent timestamp among all compared attribute groups.
- **Last Edit Date Attribute** - When no attribute is selected, the most recent date is the STEP object revision timestamp when the given element of the survivorship rule entered STEP.

  Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

  - When the selected attribute is valid for this object, timestamp is taken from the object.
  - When the selected attribute is not valid for the object, the value is taken from the given element of the survivorship rule, for example, a data container object or a reference object.
- **Survive incoming empty values** - When selected, imported empty values replace existing values. For example, for a record with the first name attribute value of 'John' and a LastEditDate of 2021-06-21. Importing the following XML:

```
<Values>
  <Value AttributeID="FirstName"></Value>
  <Value AttributeID="LastEditDate">2021-11-12 13:51:01</Value>
</Values>
```

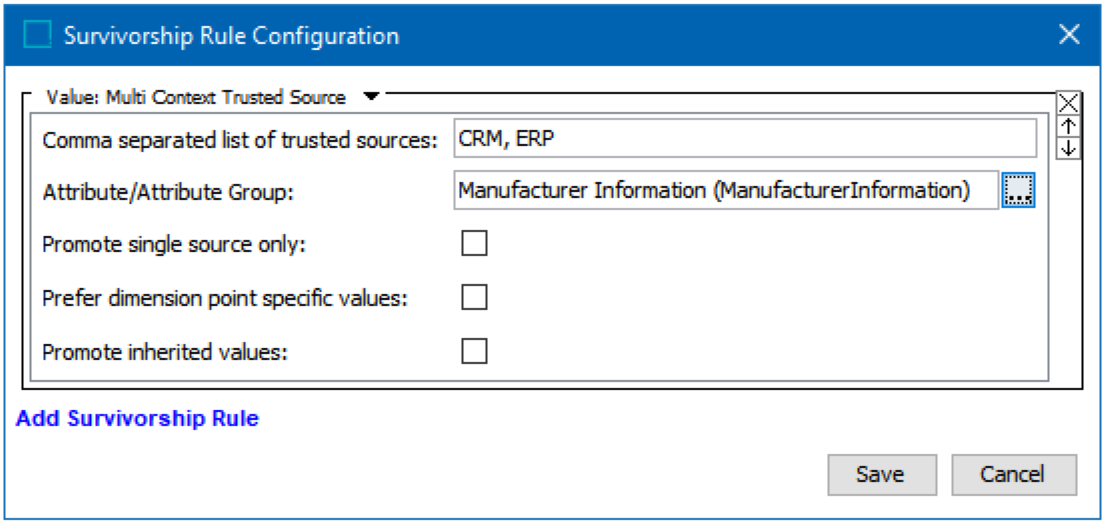
  results in the following outcome based on the checkbox setting:

  - Survive incoming empty values = checked, the first name value is updated to blank and the LastEditDate is updated to 2021-11-12 13:51:01.
  - Survive incoming empty values = not checked, the first name value is not updated but the LastEditDate is updated to 2021-11-12 13:51:01.

Value: Multi Context Trusted Source

Valid for strategies: link only

Specifies that the value is taken from the source object with the most trusted source and considers data that is dimension dependent. The analysis is performed for all contexts / qualifiers (a set of one or more dimension points, like country and language) in STEP.


Survivorship Rule Configuration
Value: Multi Context Trusted Source
Add Survivorship Rule
Save
Cancel

Comma separated list of trusted sources: CRM, ERP

Attribute/Attribute Group: Manufacturer Information (ManufacturerInformation)

Promote single source only: ☐

Prefer dimension point specific values: ☐

Promote inherited values: ☐

- **Comma separated list of trusted sources** - Enter a comma-separated list of the case-sensitive Source System ID for all trusted sources, starting with the most trusted source, then the next-most, and so on. Content is taken from the first trusted source with data. If content does not exist for any of the trusted sources, nothing is promoted to the golden record and the existing golden record value is cleaned. For information on the Source System ID Attribute setting, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.

- **Attribute / Attribute Group** - Click the ellipsis button (...) and select a single attribute or all attributes in a specific group for which the rule applies.

- **Promote single source only** - When checked, content from the most trusted source is used for all contexts / qualifiers, which prevents empty values in the golden record as long as one of the trusted sources has content. For example, when only the French language / France country context has a value, that value would be written into other contexts that are blank.

  When not checked, each context / qualifier supplies its own content, including empty values when found.

- **Prefer dimension point specific values** - When checked, only local values are promoted for the selected attribute / attribute group.

  When not checked, available inherited content is promoted if a local value does not exist for the selected attribute / attribute group.

> **Note:** If both the 'Prefer dimension point specific values' and the 'Promote single source only' options are checked, then 'Promote single source only' takes precedence, and only values from that source are promoted for the selected attribute / attribute group.

- **Promote inherited values -** When checked, inherited values are written to the golden record for the selected attribute / attribute group only if the golden record object type is valid.

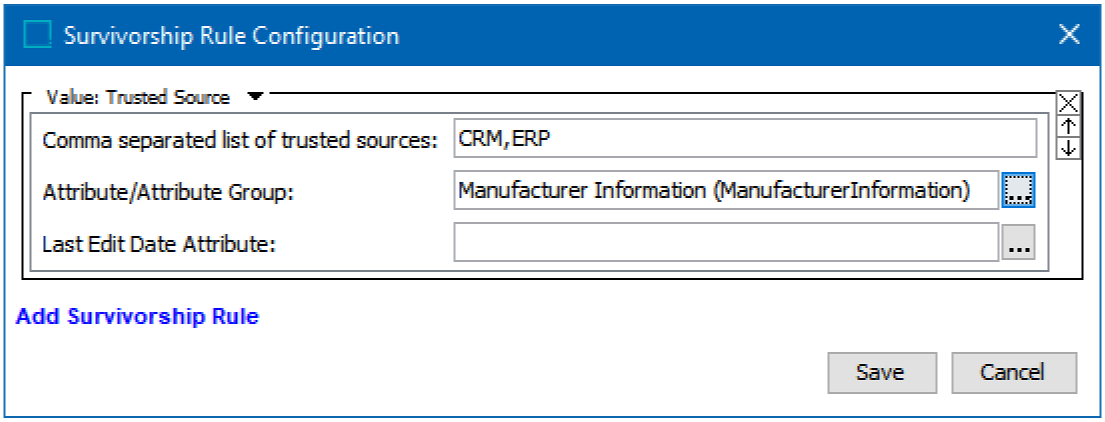
  When not checked, only local values are written to the golden record for the selected attribute / attribute group.

Value: Trusted Source

Valid for strategies: merge or link

Specifies that the value is taken from the most trusted source. The analysis is performed in the single context / workspace selected in the algorithm, and that data is promoted across all contexts / qualifiers.

In a match and merge solution, when a source system updates a record in STEP, the trusted source picks up on the golden record feedback loop and serves as a connected source. For more information on connected sources, refer to the **Survivorship in Match and Merge** topic.



Comma separated list of trusted sources: CRM,ERP

Attribute/Attribute Group: Manufacturer Information (ManufacturerInformation)

Last Edit Date Attribute:

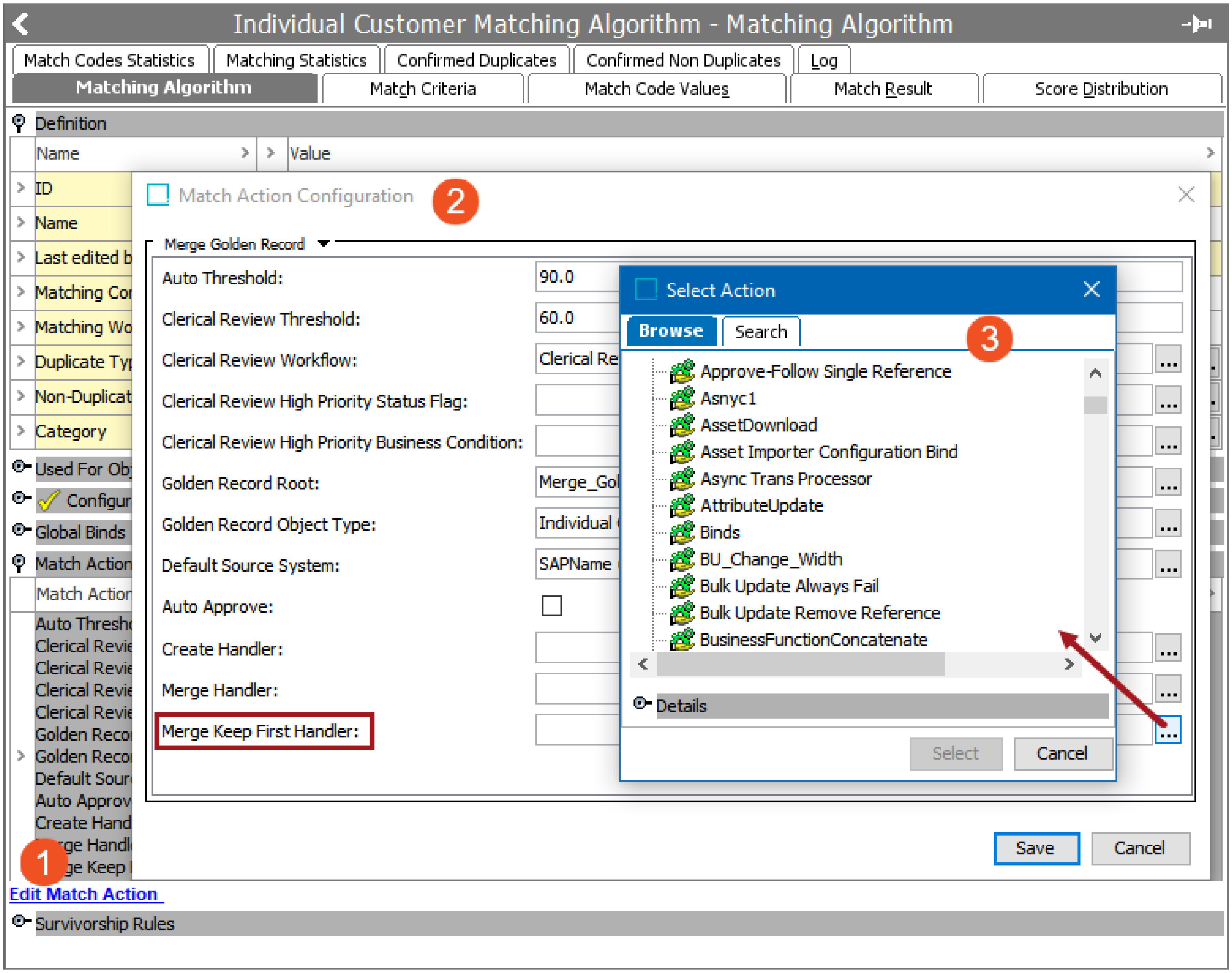
- **Comma separated list of trusted sources -** Enter a comma-separated list of the case-sensitive Source System ID for all trusted sources, starting with the most trusted source, then the next-most, and so on. Content is taken from the first trusted source with data. If content does not exist for any of the trusted sources, nothing is promoted to the golden record and the existing golden record value is cleaned. For information on the Source System ID Attribute setting, refer to the **Configuring the Matching - Merge Golden Record Component Model** topic.
- **Attribute / Attribute Group** - Click the ellipsis button (...) and select a single attribute or all attributes in a specific group for which the rule applies.
- **Last Edit Date Attribute -** When no attribute is selected, the most recent date is the STEP object revision timestamp when the given element of the survivorship rule entered STEP.

Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

- When the selected attribute is valid for this object, timestamp is taken from the object.
- When the selected attribute is not valid for the object, the value is taken from the given element of the survivorship rule, for example, a data container object or a reference object.

Creating a Merge Keep First Handler

For both 'match and merge' and match and link' solutions, merging two existing golden records – which can happen when updating information on one record results in both records being the same real-world object – one of the records must survive and the other must be deactivated. The default is to allow the record with the oldest STEP revision to persist and to deactivate the youngest record. This behavior can be overridden by adding a **Merge Keep First Handler**.



Auto Threshold: 90.0

Clerical Review Threshold: 60.0

Clerical Review Workflow: Clerical Re

Clerical Review High Priority Status Flag:

Clerical Review High Priority Business Condition:

Golden Record Root: Merge_Gol

Golden Record Object Type: Individual

Default Source System: SAPName

Auto Approve: ☐

Create Handler:

Merge Handler:

Merge Keep First Handler:

The supplied golden records are retrieved by the STEP manager with the context and workspace defined by the matching algorithm. Even if the Approved workspace is selected, the Main workspace is used since changes are not allowed in the Approved workspace.

- **Merge Keep First Handler -** Create a business condition to determine which golden record survives when two golden records are being merged. Write the business condition to evaluate 'True' to keep the first golden record and evaluate 'False' to keep the second golden record. If this handler is not used, the default behavior keeps the golden record that was created first.
    - The surviving golden record is bound to the 'Current Object' parameter. Refer to the **Current Object Bind** topic in the online help **Resource Materials** documentation.
    - The golden record to be deactivated / deleted is bound to the 'Secondary Object' parameter. Refer to the **Secondary Object Bind** topic in the online help **Resource Materials** documentation.

For information about other handlers, refer to the **Creating Merge Golden Record Match Action Handlers** topic.

Match Tuning

Tuning a matching algorithm is the process of refining the algorithm to produce the desired outcome for a variety of data scenarios. The tools available for tuning include:

- A **match tuning configuration** which allows data stewards to evaluate and adjust a matching algorithm for better accuracy when importing source records. With this tool, users can analyze data and iterate on the matching algorithm before running an import.
- The **Match Result tab** in conjunction with the **Duplicates tabs** which use standard data profiling tools to identify data entries that are appropriate for matching records.

Match Tuning Configuration

Use the following steps to configure and use your tuning solution:

1. Initial setup for match tuning, as defined in the **Initial Setup for Match Tuning** topic.
2. Create a match tuning configuration, as defined in the **Configuring Match Tuning** topic.

Match Results Tab Use

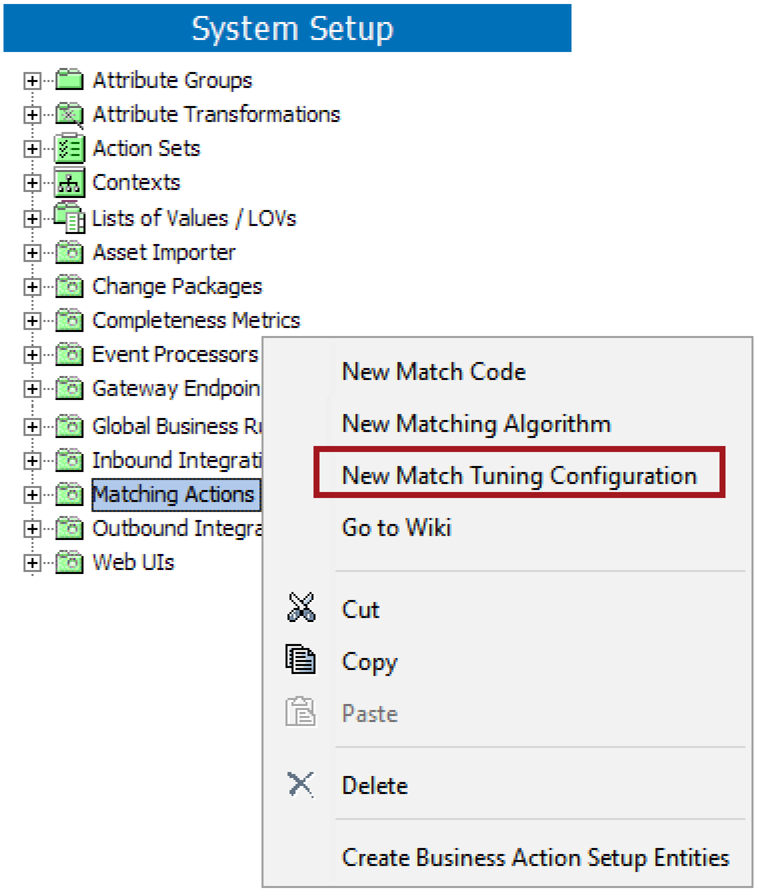
Review and improve a matching algorithm's effectiveness using the legacy functionality:

1. Match Result tab, as defined in the **Matching Algorithm - Match Result Tab** topic.
2. Confirmed Duplicates and Confirmed Non Duplicates tabs, as defined in the **Matching Algorithm - Duplicates Tabs** topic.

Initial Setup for Match Tuning

This one-time setup is required to define a match tuning group type which then holds the match tuning objects you create for use in match, link, and merge solutions.

Review your System Setup tab to determine if a match tuning node already exists. Right-click on the node and verify that the 'New Match Tuning Configuration' is enabled. The name of the node on your system is not required to match the one in the image below.



If you do not have a match tuning node, complete the following one-time setup steps.

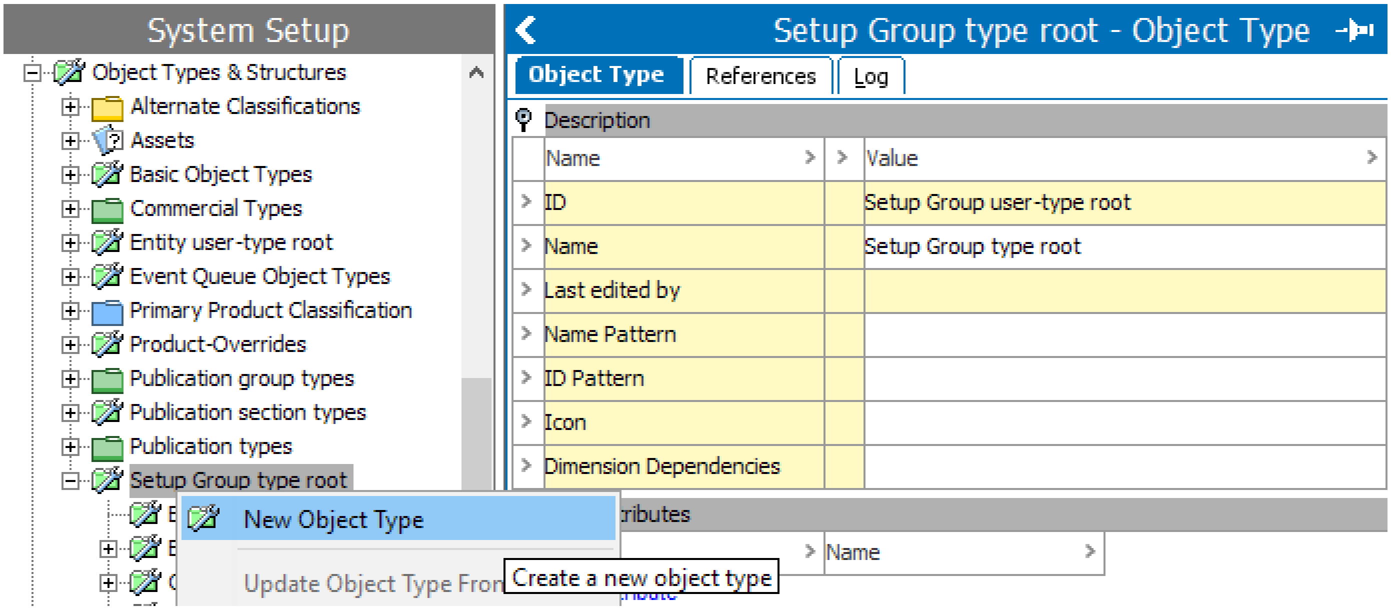
1. Create setup group type for match tuning.
2. Link match tuning object types to setup group types.
3. Create a match tuning setup group.

Once the setup has been completed, the steps in this section are only needed if you want additional levels of organization.

Create Setup Group Type for Match Tuning

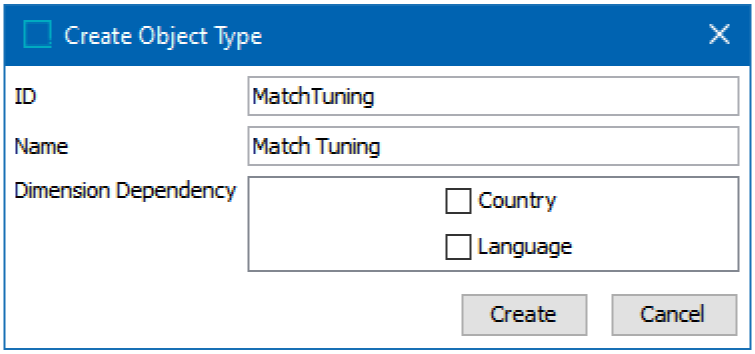
A match tuning group type defines the structure and allowed locations of a match tuning configuration.

1. Go to System Setup > Object Types & Structures > select **Setup Group type root**.
2. Right-click **Setup Group type root**, and the New Object Type option will display.



Name	Value
ID	Setup Group user-type root
Name	Setup Group type root
Last edited by	
Name Pattern	
ID Pattern	
Icon	
Dimension Dependencies	

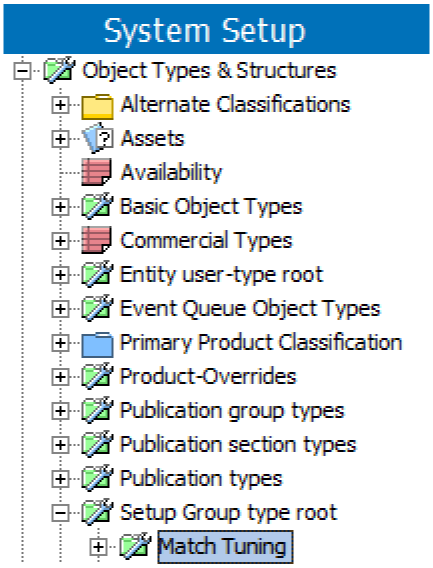
3. Click **New Object Type**, and the Create Object Type dialog will display.



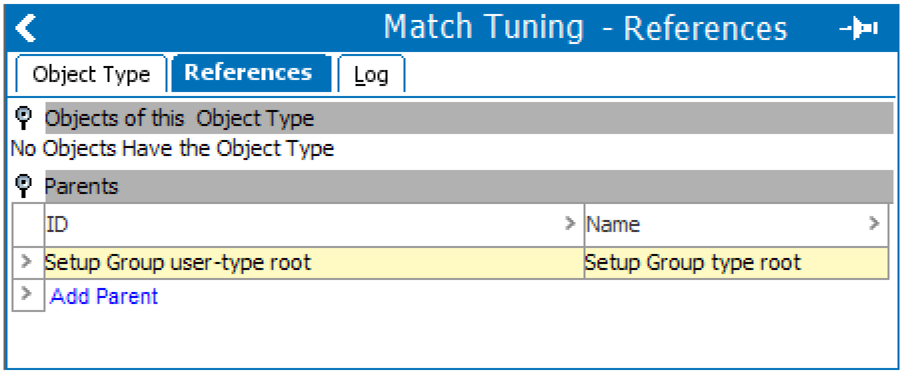
ID	MatchTuning
Name	Match Tuning
Dimension Dependency	☐ Country ☐ Language

4. Enter an **ID**.
5. Enter a **Name**.
6. Click **Create.**

   The Create Object Type dialog closes, and the newly created object type for the matching algorithm displays beneath the Setup Group type root.



7. Select the newly added Setup Group type > References tab > open the Parents flipper.



Objects of this Object Type

No Objects Have the Object Type

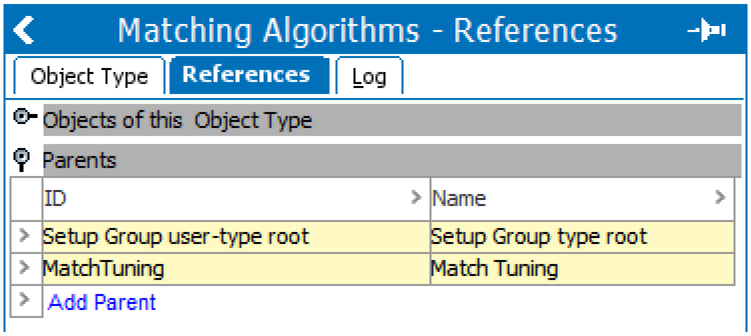
Parents

ID	Name
Setup Group user-type root	Setup Group type root
Add Parent	

Important: By default the Setup Group type root is listed as the parent. Optionally add the newly created setup group type as a parent of itself so that additional match tuning group types can be added below the main level.

8. Click **Add Parent**, and the Select New parent dialog displays.
9. Browse or search to select **the relevant setup group type**.
10. Click the **Select** button.

    The dialog closes, and the newly created setup group type (i.e., Match Tuning) is listed as a parent along with the Setup group user-type root.

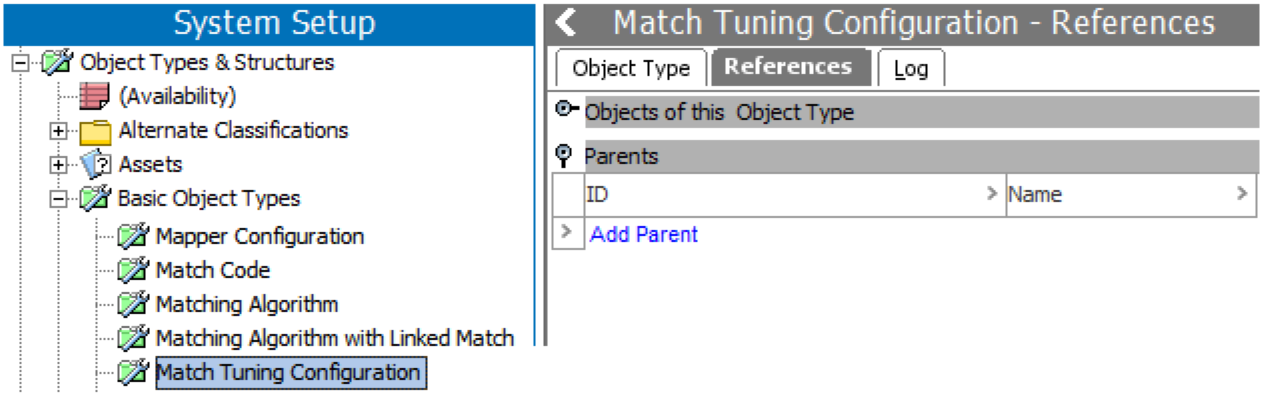


ID	Name
Setup Group user-type root	Setup Group type root
MatchTuning	Match Tuning

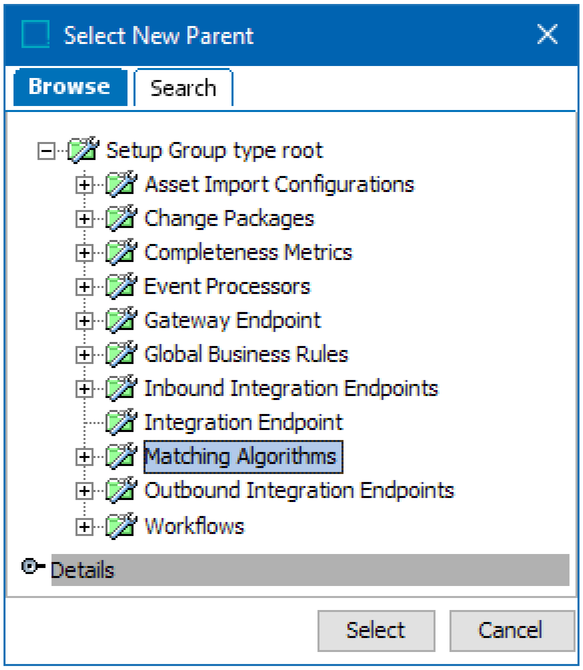
Link Match Tuning Object Types to Setup Group Types

Linking determines the object types that can be displayed at each level of a hierarchy.

1. Go to System Setup> Object Types & Structures > **Basic Object Types**.
2. Select **your match tuning object type** to display the editor.


System Setup
Object Types & Structures
(Availability)
Alternate Classifications
Assets
Basic Object Types
Mapper Configuration
Match Code
Matching Algorithm
Matching Algorithm with Linked Match
Match Tuning Configuration
Match Tuning Configuration - References
Object Type
References
Log
Objects of this Object Type
Parents
ID
Name
Add Parent

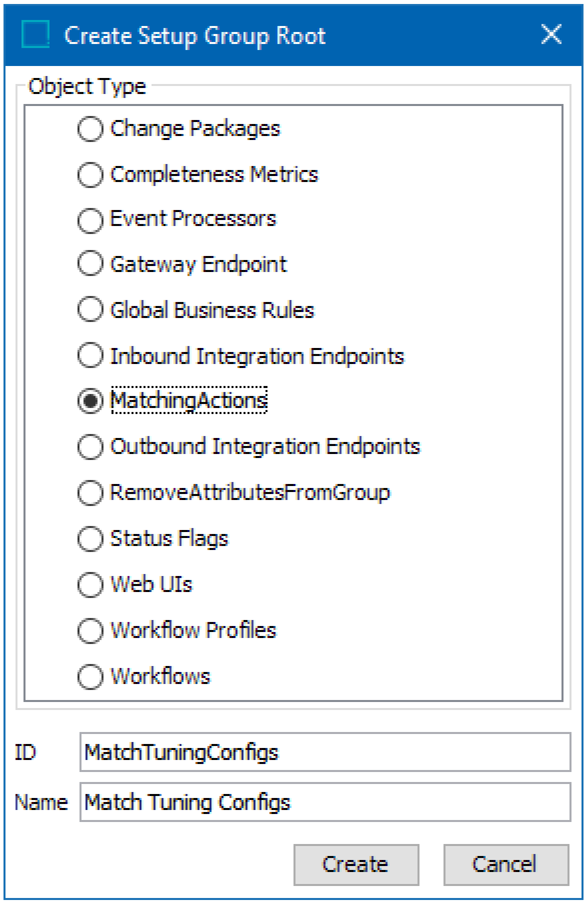
3. Click the **References** tab.
4. Open the **Parents** flipper.
5. Click the **Add Parent** link, and the Select New Parent dialog displays.
6. Browse or search to select **the relevant setup group type**.
7. Click the **Select** button.



Create a Match Tuning Setup Group

Creating a setup group allows your match tuning setup group type to appear as a node in the System Setup hierarchy.

1. Go to System Setup > select **any object in the hierarchy**.
2. On the menu bar, select **Maintain** > **Insert** > **Setup Group Root**, and the Create Setup Group Root dialog will display.



Create Setup Group Root

Object Type

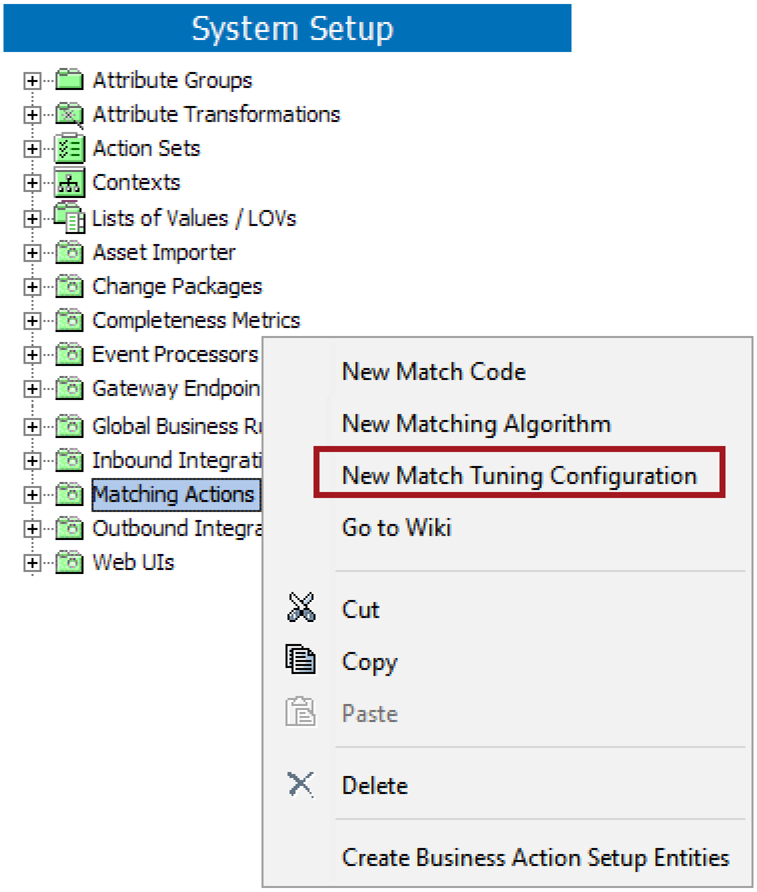
○ Change Packages
○ Completeness Metrics
○ Event Processors
○ Gateway Endpoint
○ Global Business Rules
○ Inbound Integration Endpoints
● MatchingActions
○ Outbound Integration Endpoints
○ RemoveAttributesFromGroup
○ Status Flags
○ Web UIs
○ Workflow Profiles
○ Workflows

ID: MatchTuningConfigs

Name: Match Tuning Configs

3. Select **your match tuning object type**.
4. Enter an **ID**.
5. Enter a **Name**.
6. Click **Create**.

   The setup group is created and appears as a node in the System Setup hierarchy, and allows the creation of match tuning configurations.



7. Continue with the **Configuring Match Tuning** topic.

Configuring Match Tuning

A match tuning configuration allows users to analyze data and iterate on the matching algorithm before running an import.

The Evaluate Matching Algorithm action on a Match Tuning Configuration can generate a Pair Export report and a Match Codes Export report.

A data steward should use the reports to:

- ensure the matching algorithm produces the correct results.
- ensure the matching algorithm can work efficiently with the data.

Match Tuning is an iterative process, adjusting the match codes, match criteria, and thresholds, and then evaluating the results repeatedly until the algorithm is good enough.

Match Tuning goals should include:

- No match code group size larger than 100 and generally, most objects (95 percent) should be in a match code group with a size of 10 or smaller since match codes have a huge impact on performance.
- Use Replacement Lookup Tables to avoid comparisons where possible, paying attention to values like company main phone numbers or shared group email addresses.
- In general, it is recommended to limit the number of records going into clerical review as much as possible, however, it is always a business evaluation where to set the thresholds for auto merge and auto reject. When not sufficiently limited, the result is often an extensive list of unhandled tasks in the clerical review task list.

For other tuning options, refer to the **Match Tuning** topic.

Prerequisites

1. Ensure that the initial setup is complete, as defined in the **Initial Setup for Match Tuning** topic.
2. Open the Matching component model and view the 'Match Tuning Asset Object Types' parameter to identify the asset object types used to store the profile data for match tuning. If none are selected, refer to the **Configuring Matching Component Model** topic.
3. In the Object Types & Structures node, open the Assets folder and verify the Match Tuning Asset Object Types have the 'Reference Target Lock Policy' parameter set to 'Strict.' For information on this parameter, refer to the **Reference Target Lock Policy on Object Types** topic of the **System Setup** documentation.
4. Create or identify a classification folder to hold sample import data used during match tuning.
5. Identify the matching algorithm to be fine-tuned.

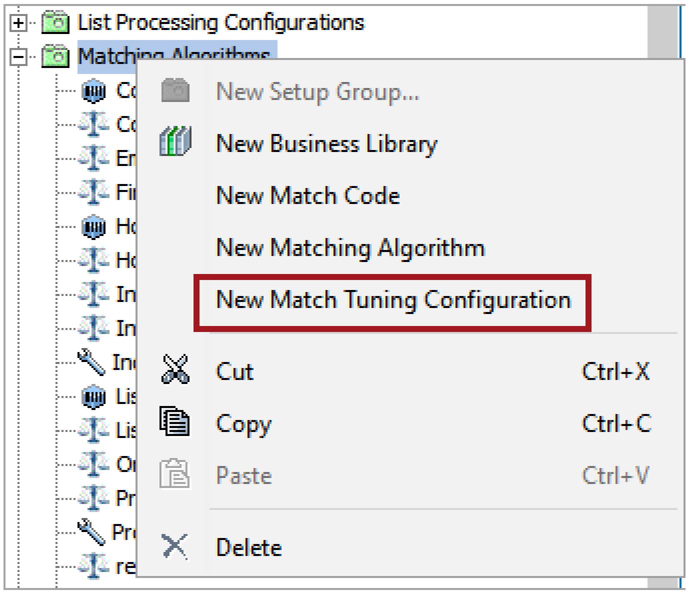
6. Configure a data profile, as defined in the **Data Profiles** topic of the **Data Profiling** documentation. Be aware that since the data being profiled originates outside of STEP, features such as bulk update, search, and saving collections are not available.

7. Consider normalizing values that are always populated before using them in matching. For use case examples, refer to the **Data Governance** topic in the **Customer MDM Solution Enablement** documentation.

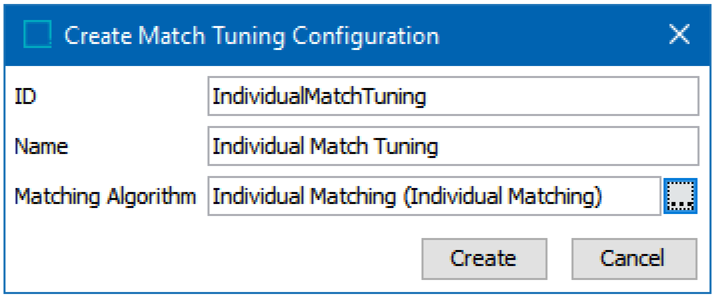
Configuration

Use the following steps to set up match tuning.

1. In the System Setup tab, right-click the match tuning configurations node and select 'New Match Tuning Configuration.' The name of the node on your system may be different than shown in the images.



2. In the 'Create Match Tuning Configuration' dialog, add an **ID**, a **Name**, and specify a matching algorithm to test, Click the **Create** button.



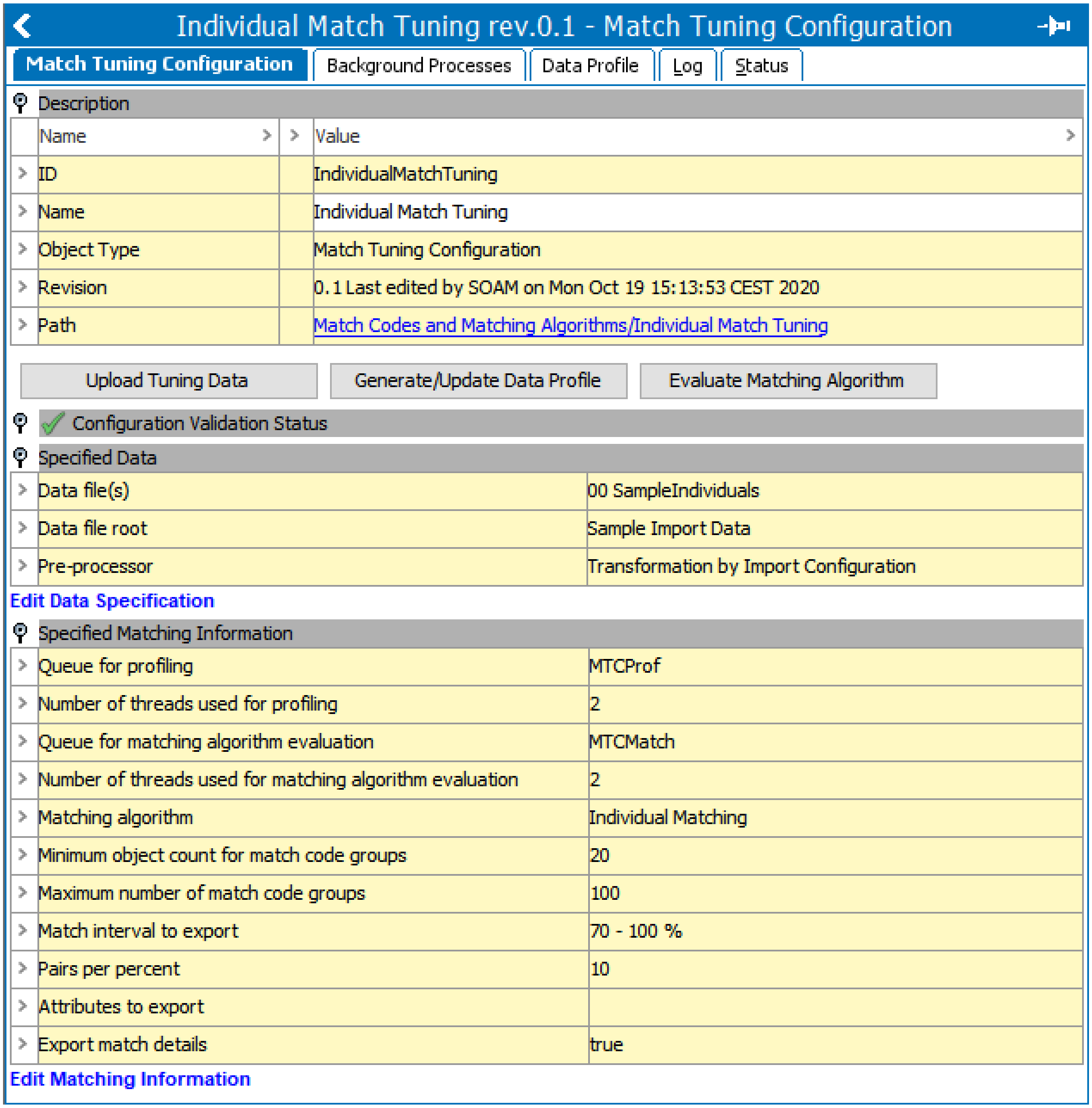
Create Match Tuning Configuration

ID: IndividualMatchTuning

Name: Individual Match Tuning

Matching Algorithm: Individual Matching (Individual Matching)

3. Click the 'Match Tuning Configuration' tab to view the overall configuration.


Individual Match Tuning rev.0.1 - Match Tuning Configuration
Match Tuning Configuration
Background Processes
Data Profile
Log
Status
Upload Tuning Data
Generate/Update Data Profile
Evaluate Matching Algorithm
Edit Data Specification
Edit Matching Information

Description

Name	Value
ID	IndividualMatchTuning
Name	Individual Match Tuning
Object Type	Match Tuning Configuration
Revision	0.1 Last edited by SOAM on Mon Oct 19 15:13:53 CEST 2020
Path	Match Codes and Matching Algorithms/Individual Match Tuning

✓ Configuration Validation Status

Specified Data

Data file(s)	00 SampleIndividuals
Data file root	Sample Import Data
Pre-processor	Transformation by Import Configuration

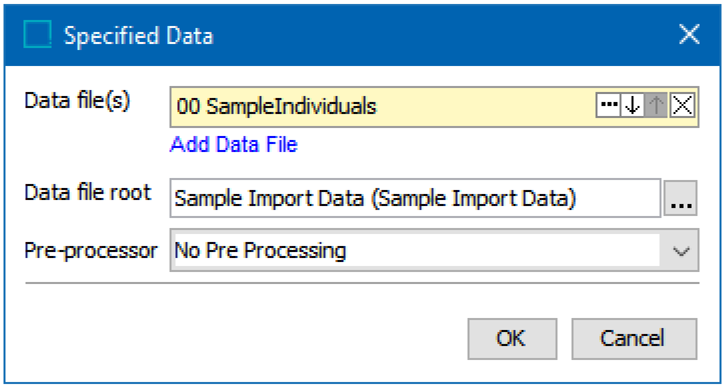
Specified Matching Information

Queue for profiling	MTCProf
Number of threads used for profiling	2
Queue for matching algorithm evaluation	MTCMatch
Number of threads used for matching algorithm evaluation	2
Matching algorithm	Individual Matching
Minimum object count for match code groups	20
Maximum number of match code groups	100
Match interval to export	70 - 100 %
Pairs per percent	10
Attributes to export	
Export match details	true

On the Configuration Validation Status flipper:

- ✓ a green check indicates that the configuration is valid.
- ✗ a red X indicates errors exist. Open the flipper to review the errors.

4. Open the 'Specified Data' flipper and click the **Edit Data Specification** link to display the 'Specified Data' dialog.



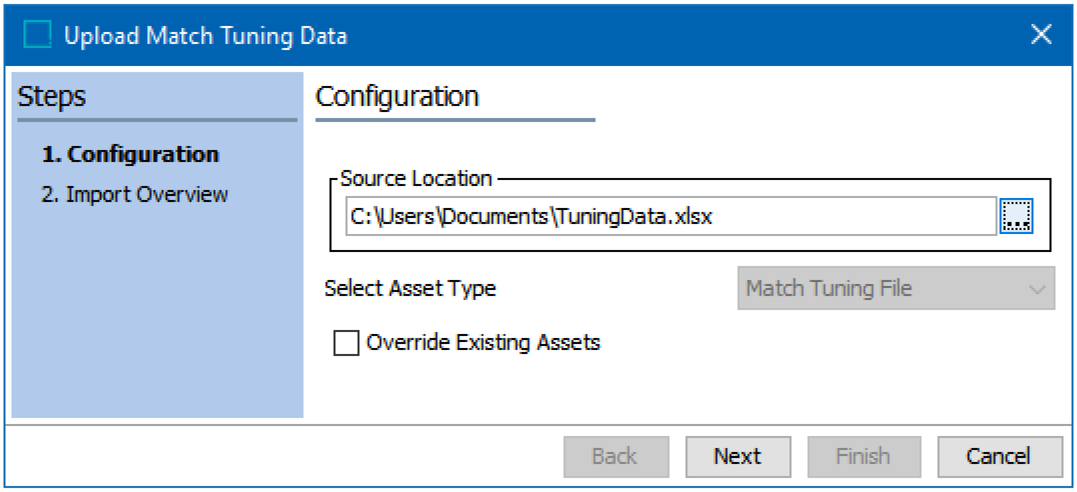
Data file(s) 00 SampleIndividuals

Data file root Sample Import Data (Sample Import Data)

Pre-processor No Pre Processing

- For the **Data file(s)** parameter, click the **Add Data File** link to select uploaded data files. The Data file root parameter (set below) defines the folder for the uploaded files.
- For the **Data file root** parameter, click the ellipsis button (**...**) and specify the location where tuning data is stored and enable the Upload Tuning Data button.
- For the **Pre-processor** parameter, if required, select a pre-processor to convert non-STEPXML data. For more information on converting the CSV / Excel files in this way, refer to the **IIEP - Configure Match and Merge Importer Processing Engine** topic of the **Data Exchange** documentation.

5. Click the **Upload Tuning Data** button to display the Upload Match Tuning wizard. Uploading data saves it as an asset in STEP and makes it available for selection in the match tuning configuration.



Source Location C:\Users\Documents\TuningData.xlsx

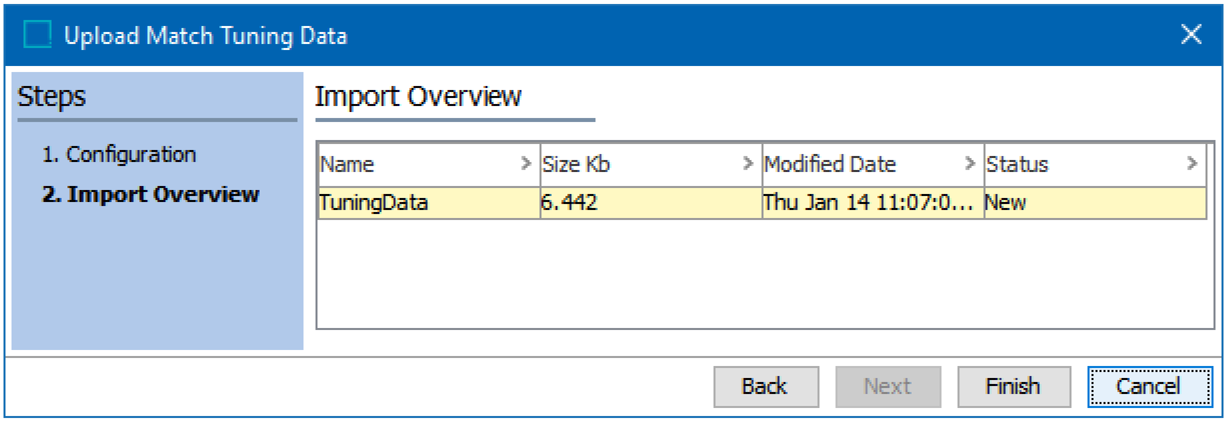
Select Asset Type Match Tuning File

□ Override Existing Assets

The sample data used for tuning is uploaded as follows:

- For the **Source Location** parameter, click the ellipsis button (**...**) and select a data file. Asset name must be less than 40 characters long.
- For the **Select Asset Type** parameter, specify the match tuning asset type.

- For the **Override Existing Assets** parameter, when checked, previously uploaded tuning data is overwritten.
- Click **Next** to review the Import Overview and click **Finish** to upload the tuning data. Any errors with the import are displayed.



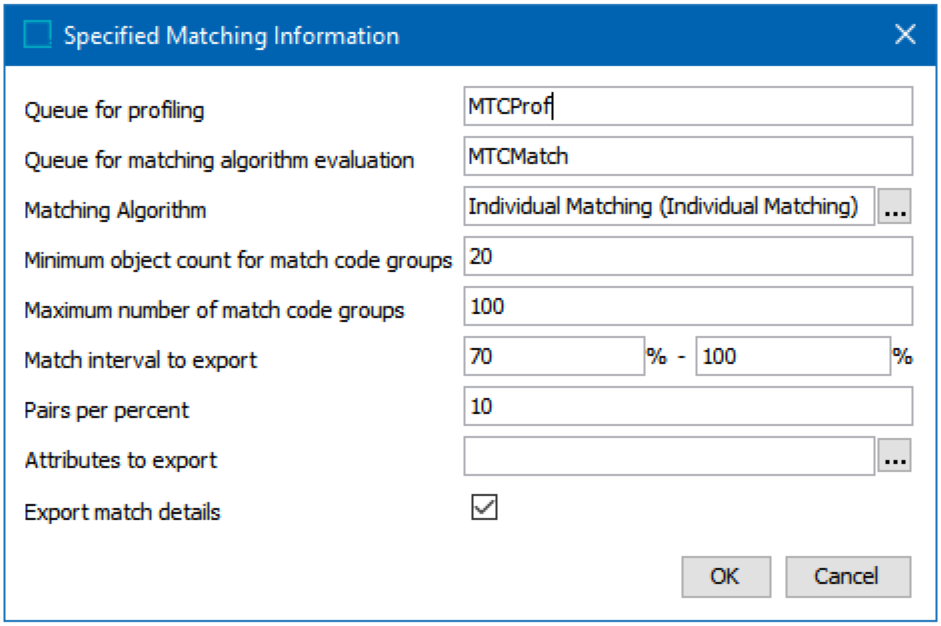
Upload Match Tuning Data

Steps
1. Configuration
2. Import Overview

Import Overview

Name	Size Kb	Modified Date	Status
TuningData	6.442	Thu Jan 14 11:07:0...	New

6. Click the **Generate / Update Data Profile** button to generate the data profile. Monitor the background process for success or errors. Resolve any errors and repeat this step.
7. Click the Data Profile tab to review the results of the profiling process.
8. Click the Match Tuning Configuration tab, open the Specified Matching Information flipper and click the **Edit Matching Information** link to modify the relevant parameters:



Specified Matching Information

Queue for profiling	MTCProf
Queue for matching algorithm evaluation	MTCMatch
Matching Algorithm	Individual Matching (Individual Matching) ...
Minimum object count for match code groups	20
Maximum number of match code groups	100
Match interval to export	70 % - 100 %
Pairs per percent	10
Attributes to export	...
Export match details	☑

- **Queue for profiling** - The background process queue created for profiling.
- **Queue for matching algorithm evaluation** - The background process queue created for matching algorithm evaluation.
- **Matching Algorithm** - Click the ellipsis button (**...**) and browse or search for the matching algorithm the match tuning configuration should test.
- **Minimum object count for match code groups** - Enter the minimum number of objects to be exported per match code group.
- **Maximum number of match code groups** - Enter the maximum amount of match code groups the tuning data can generate.
- **Match interval to export** - Specify an interval that includes pairs expected to be both matches and non-matches, as well as pairs that are not clear matches or non-matches. Only pairs with scores within this interval are exported.
- **Pairs per percent** - Enter the maximum number of pairs to be exported for each percentage point.
- **Attribute to export** - Click the ellipsis button (**...**) and select the attribute values that should be exported.
- **Export match details** - Check the box to add additional columns with part scores from decision table comparators and sub decision tables.

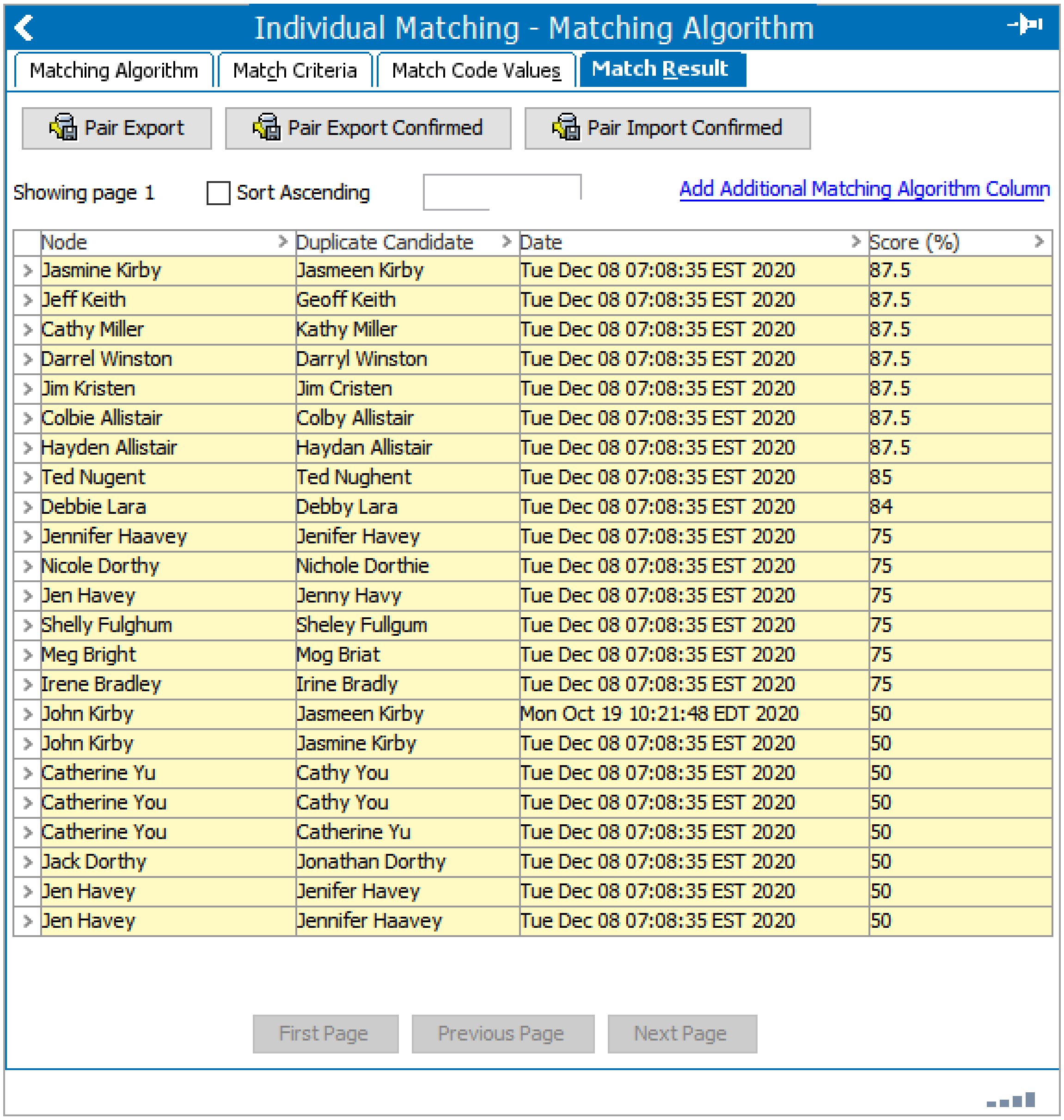
9. Click the **Evaluate Matching Algorithm** button to start a background process that creates a pair export file and match codes export file.
10. Click the Background Processes tab, click the BGP link for the completed Matching process.
11. In the BGP Result flipper, download the exported file and review the profile data.

Matching Algorithm - Match Result Tab

When a matching algorithm is applied, the identified matches are displayed on the 'Match Result' tab of the matching algorithm. This tool can be used along with the duplicates tabs, as defined in the **Matching Algorithm - Duplicates Tabs** topic.

For **Match and Merge**, it is recommended to do the first rounds of tuning using the match tuning option, as defined in the **Match Tuning** topic. Match and Merge cannot be reapplied in the same way that Match and Link can (as described below). Match tuning using Match Result does not override earlier merge decisions by the match algorithm. While you can tune a running system using the Match Result tab, you may have to manually unmerge erroneously merged records.

For **Match and Link**, you can bypass the match tuning step since the algorithm is non-invasive towards the source records and you can rerun to fully recalculate the golden records.



Node	Duplicate Candidate	Date	Score (%)
Jasmine Kirby	Jasmeen Kirby	Tue Dec 08 07:08:35 EST 2020	87.5
Jeff Keith	Geoff Keith	Tue Dec 08 07:08:35 EST 2020	87.5
Cathy Miller	Kathy Miller	Tue Dec 08 07:08:35 EST 2020	87.5
Darrel Winston	Darryl Winston	Tue Dec 08 07:08:35 EST 2020	87.5
Jim Kristen	Jim Cristen	Tue Dec 08 07:08:35 EST 2020	87.5
Colbie Allistair	Colby Allistair	Tue Dec 08 07:08:35 EST 2020	87.5
Hayden Allistair	Haydan Allistair	Tue Dec 08 07:08:35 EST 2020	87.5
Ted Nugent	Ted Nughent	Tue Dec 08 07:08:35 EST 2020	85
Debbie Lara	Debby Lara	Tue Dec 08 07:08:35 EST 2020	84
Jennifer Haavey	Jenifer Havey	Tue Dec 08 07:08:35 EST 2020	75
Nicole Dorthy	Nichole Dorthie	Tue Dec 08 07:08:35 EST 2020	75
Jen Havey	Jenny Havy	Tue Dec 08 07:08:35 EST 2020	75
Shelly Fulghum	Sheley Fullgum	Tue Dec 08 07:08:35 EST 2020	75
Meg Bright	Mog Briat	Tue Dec 08 07:08:35 EST 2020	75
Irene Bradley	Irine Bradly	Tue Dec 08 07:08:35 EST 2020	75
John Kirby	Jasmeen Kirby	Mon Oct 19 10:21:48 EDT 2020	50
John Kirby	Jasmine Kirby	Tue Dec 08 07:08:35 EST 2020	50
Catherine Yu	Cathy You	Tue Dec 08 07:08:35 EST 2020	50
Catherine You	Cathy You	Tue Dec 08 07:08:35 EST 2020	50
Catherine You	Catherine Yu	Tue Dec 08 07:08:35 EST 2020	50
Jack Dorthy	Jonathan Dorthy	Tue Dec 08 07:08:35 EST 2020	50
Jen Havey	Jenifer Havey	Tue Dec 08 07:08:35 EST 2020	50
Jen Havey	Jennifer Haavey	Tue Dec 08 07:08:35 EST 2020	50

Truth Table

Determining how well different versions of a matching algorithm work requires a 'truth table'; a set of known data that includes verified duplicates and non-duplicates. A truth table includes pairs of objects that a user has inspected and determined are duplicates or not. A truth table can be built from the Match Result tab using either the information in the tab or the ‘Pair Export’ option.

Using ‘Pair Import Confirmed’ and ‘Pair Export Confirmed’ features, a **Match and Link** or **Identify Duplicates** solution can continuously evaluate the results of the algorithm against the truth table. This import is less valuable in **Match and Merge** as it does not use Confirmed Duplicate references, instead, it merges the information directly into the golden records.

Note: For Match and Merge solutions, the Pair Import and Export tools are not applicable for early evaluations. Instead, use of the Match Tuning functionality to adjust matching algorithms. For more information, refer to the **Match Tuning** documentation.

Pair Export

The Pair Export option generates a CSV file that can be used for manual, offline confirmation and rejection of matched pairs. Use this option to export match scores.

The file has a header and the following standard columns:

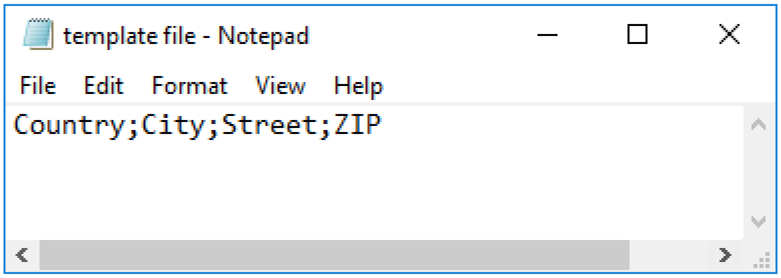
- **<Pair>** - One row per source object and the 'Pair' info is used to indicate which objects belong together. The first two rows have the value '1,' the next two rows have '2,' and so on.
- **<Match y n>** - Indicates whether pairs are matches or not. A value is only required for the first object in a pair.
- **<Equality>** - The calculated equality percentage between the two objects.
- **<ID>** - ID of the object in the current row.
- **<Name>** - Name of the object in the current row.
- **<URL>** - STEP URL of the object in the current row.

While no template is required for the initial export, to work with the data offline, include attribute values in the file via a template file.

Prerequisite

Create a basic text document template file to be selected in the dialog as follows:

- Attribute IDs separated by semicolons (;)
- Save as CSV format

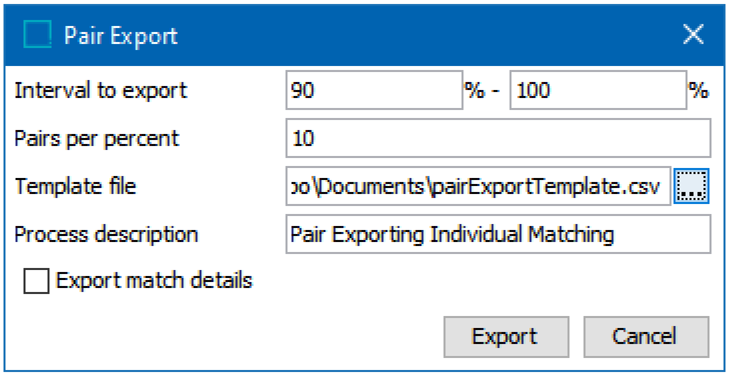


```
Country;City;Street;ZIP
```

Configuration

Use the following steps to perform Pair Export.

1. Click the **Pair Export** button.
2. In the Pair Export dialog, specify the following:



Pair Export

Interval to export: 90 % - 100 %

Pairs per percent: 10

Template file: ɔo\Documents\pairExportTemplate.csv

Process description: Pair Exporting Individual Matching

☐ Export match details

   - **Interval to export:** Specify an equality percentage interval that includes pairs expected to be both matches and non-matches, as well as pairs that are not clearly matches or non-matches. Only pairs with equality scores within this interval are exported.
   - **Pairs per percent:** Specify the maximum number of pairs to be exported for each percentage point.
   - **Template file:** Select the template file that contains the required attribute values.
   - **Process description:** Provide a description for the background process found under the Background Process tab.
   - **Export Match Details:** When checked, columns with part scores from decision table comparators and sub decision tables are included.

3. Click the **Export** button to start the background process.
4. From the BGP, open the exported file in Excel, and enter the decisions in the <Match y n> column for the first object in a pair.
5. Save your changes.
6. Use the **Pair Import Confirmed** option defined below to apply the manual matches added to the file.

Pair Import Confirmed

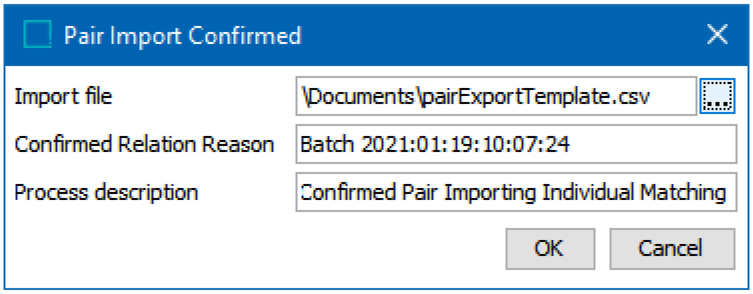
After the file exported via the pair export option has been populated with matches, it can be imported via the 'Pair Import Confirmed' option. The 'Pair Import Confirmed' process uses its the data for identification purposes but does not import anything other than the confirmation data. This avoids reverting values updated elsewhere since the pair export was performed.

Configuration

Use the following steps to perform Pair Import Confirmed.

1. Click the **Pair Import Confirmed** button.

2. In the Pair Import Confirmed dialog, specify the following:



Pair Import Confirmed

Import file: \Documents\pairExportTemplate.csv

Confirmed Relation Reason: Batch 2021:01:19:10:07:24

Process description: Confirmed Pair Importing Individual Matching

   - **Import File:** Select the CSV file to import. This file must have been produced by the Pair Export process, use a semicolon delimiter, and include the header row.

   - **Confirmed Relation Reason:** Provide a reason for confirming the objects as duplicates or non-duplicates. This reason is saved on each confirmed relation as a meta data attribute and can be viewed on the matching tab of the relevant objects.

   - **Process description:** Provide a description for the background process.

3. Click the **OK** button to start the background process.

4. Review the BGP Execution Report for a count of the matches and the Confirmed Duplicates and Confirmed Non Duplicates tabs for the modified records.

Pair Export Confirmed

The Pair Export Confirmed option allows you to compare two versions of a matching algorithm against the confirmed duplicates / non duplicates truth table constructed manually or via the steps described above. A background process generates a CSV file with the comparison results and enables the Match Distribution tool. This tool allows the user to view the differences between the match algorithms and compare their accuracy.

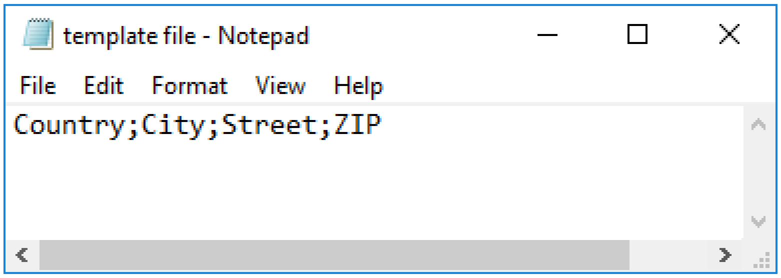
Prerequisites

1. Duplicate your matching algorithm and edit the copy as desired. You will compare the original and the copy which has been fine-tuned.

2. Create a basic text document template file to be selected in the dialog as follows:

   - Attribute IDs separated by semicolons (;)

- Save as CSV format



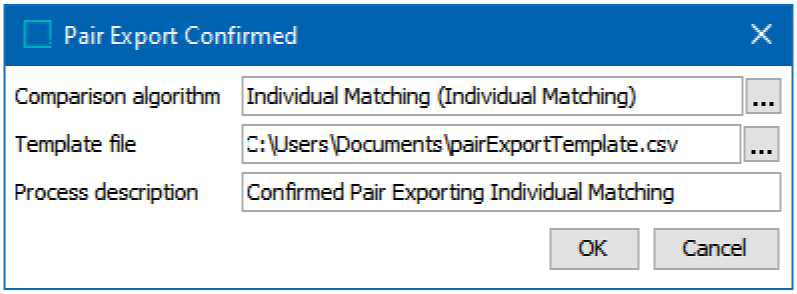
```
Country;City;Street;ZIP
```

Configuration

Use the following steps to perform Pair Export Confirmed.

1. Click the **Pair Export Confirmed** button.

2. In the 'Pair Export Confirmed' dialog, specify the following:



Comparison algorithm	Individual Matching (Individual Matching)
Template file	C:\Users\Documents\pairExportTemplate.csv
Process description	Confirmed Pair Exporting Individual Matching

   - **Comparison Algorithm:** Select the fine-tuned matching algorithm that you want to compare with the selected algorithm (the original).

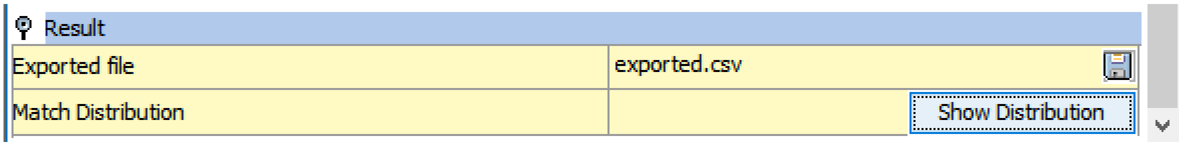
   - **Template File:** Select the CSV file to import. This file must have been produced by the Pair Export process, use a semicolon delimiter, and include the header row.

   - **Process description:** Provide a description for the background process.

3. Click the **OK** button to start the background process.

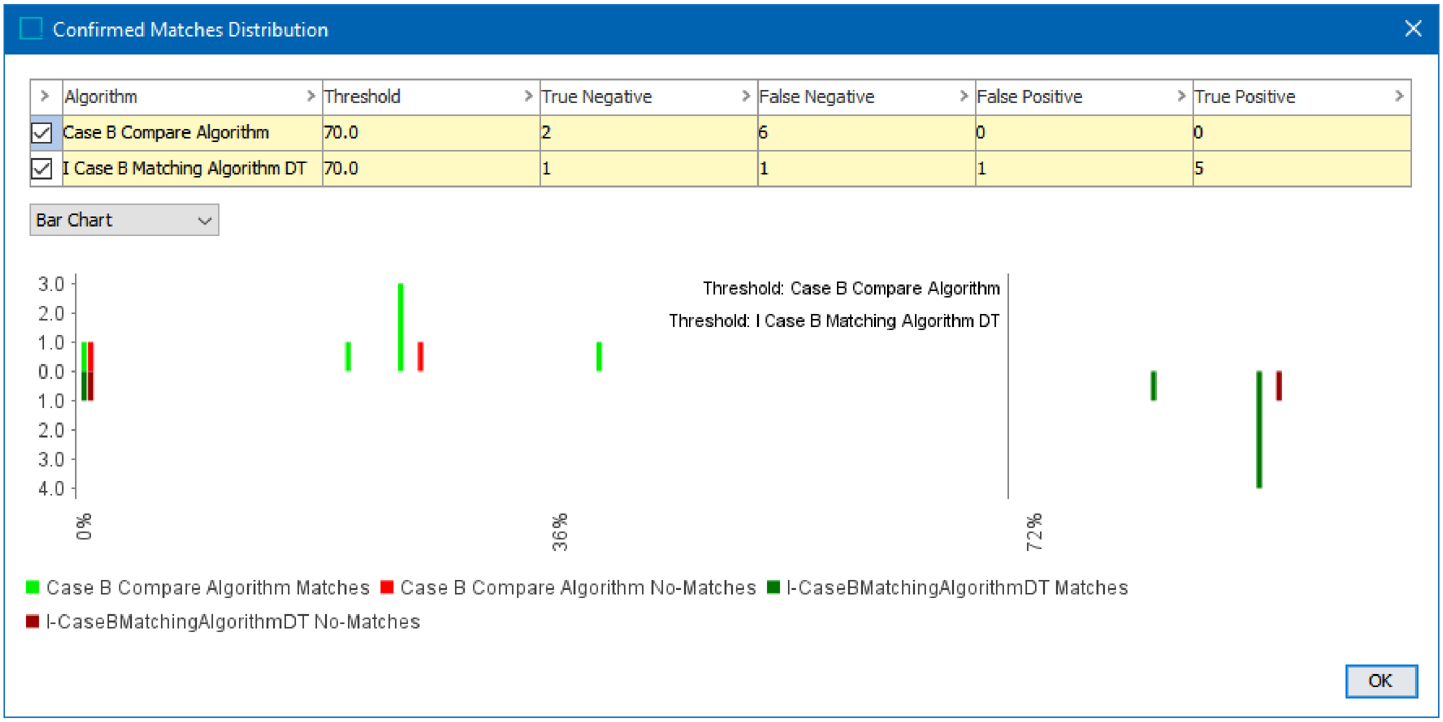
4. Click the **Go to process** button, or on the BG Processes tab, expand the 'Matching Pair Export' node and select the relevant confirmed export process.

5. On the BGP, open the Result flipper and on the Match Distribution row click the **Show Distribution** button.



Exported file	exported.csv
Match Distribution	

6. On the Confirmed Matches Distribution dialog:
   - In the table, select a row to view the **algorithm data** in a chart. Each column is defined in a section following these steps.
   - From the dropdown, select **Bar Chart** or **Accumulated Chart**. Each is defined in a section following these steps.


Confirmed Matches Distribution
Bar Chart
3.0
2.0
1.0
0.0
1.0
2.0
3.0
4.0
0%
36%
72%
Threshold: Case B Compare Algorithm
Threshold: I Case B Matching Algorithm DT
🟩 Case B Compare Algorithm Matches
🟥 Case B Compare Algorithm No-Matches
🟩 I-CaseBMatchingAlgorithmDT Matches
🟥 I-CaseBMatchingAlgorithmDT No-Matches
OK

	Algorithm	Threshold	True Negative	False Negative	False Positive	True Positive
☑	Case B Compare Algorithm	70.0	2	6	0	0
☑	I Case B Matching Algorithm DT	70.0	1	1	1	5

7. Review the data and determine possible next steps to improve the algorithm. Click the **OK** button to close the dialog.

8. Repeat this process as required.

9. When the fine-tuned version of the matching algorithm produces fewer or zero 'False Positives' and 'False Negatives', choose an option to update the algorithm in use:

   - Copy the logic to the original matching algorithm

   - Replace the original algorithm with the fine-tuned version

Algorithm Data

When reviewing the results, false negatives and false positives are the errors produced by the algorithm when compared to the manually reviewed pairs. While the goal of fine-tuning an algorithm is to achieve 0 false results, having a count of 0 does not mean that the algorithm is perfect. The reliability of the result depends on the amount of data in the testing data set and how well the test data set represents the full data.

On the Confirmed Matches Distribution dialog, the table shows the following information about each algorithm:

- **Algorithm:** The ID of the algorithm.
- **Threshold:** The threshold used to distinguish between positives and negatives.
- **True Negative:** The number of comparisons that were classified as a non-match, both manually, and by the algorithm.
- **False Negative:** Count of comparisons that were manually classified as a match, but the algorithm classified as a non-match because the scores were below the threshold.
- **False Positive:** Count of comparisons that were manually classified as a non-match, but the algorithm classified as a match because the scores were above the threshold.
- **True Positive:** Count of comparisons that were classified as a match both manually and by the algorithm.

Data Charts

For both the Bar Chart and the Accumulated Chart, the colors are identified in the chart legend shown below the chart, and generally:

- Green represents relations that have been manually confirmed as duplicates
- Red represents relations that have been manually confirmed as non-duplicates.

The threshold of the algorithm is shown as a vertical line.

Bar Chart

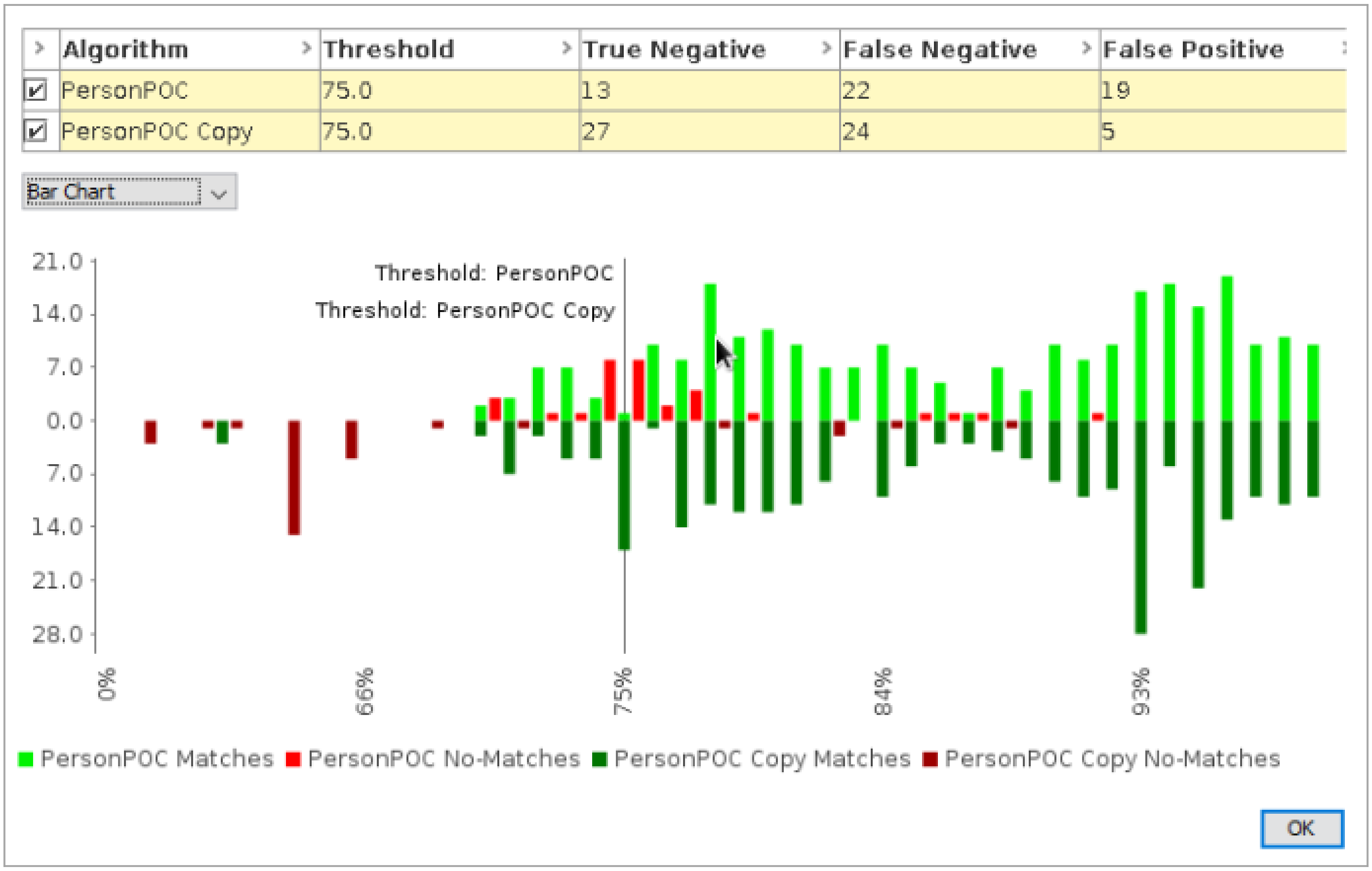
The bars in the chart show the frequency of the scores of the selected algorithm. The bar chart can either show a single algorithm or two algorithms in a special compare mode that enables a detailed comparison of the two algorithms.

Red bars are usually displayed to the left of the threshold indicator and green bars to the right.

- Green bars displayed to the left of the threshold represent false negatives.
- Red bars displayed to the right of the threshold indicator represent false positives.

For exact numbers of false positives and false negatives, review the table. Because the bars have a resolution of 1 percent point, the exact number of false positives and false negatives are not available in the graph.

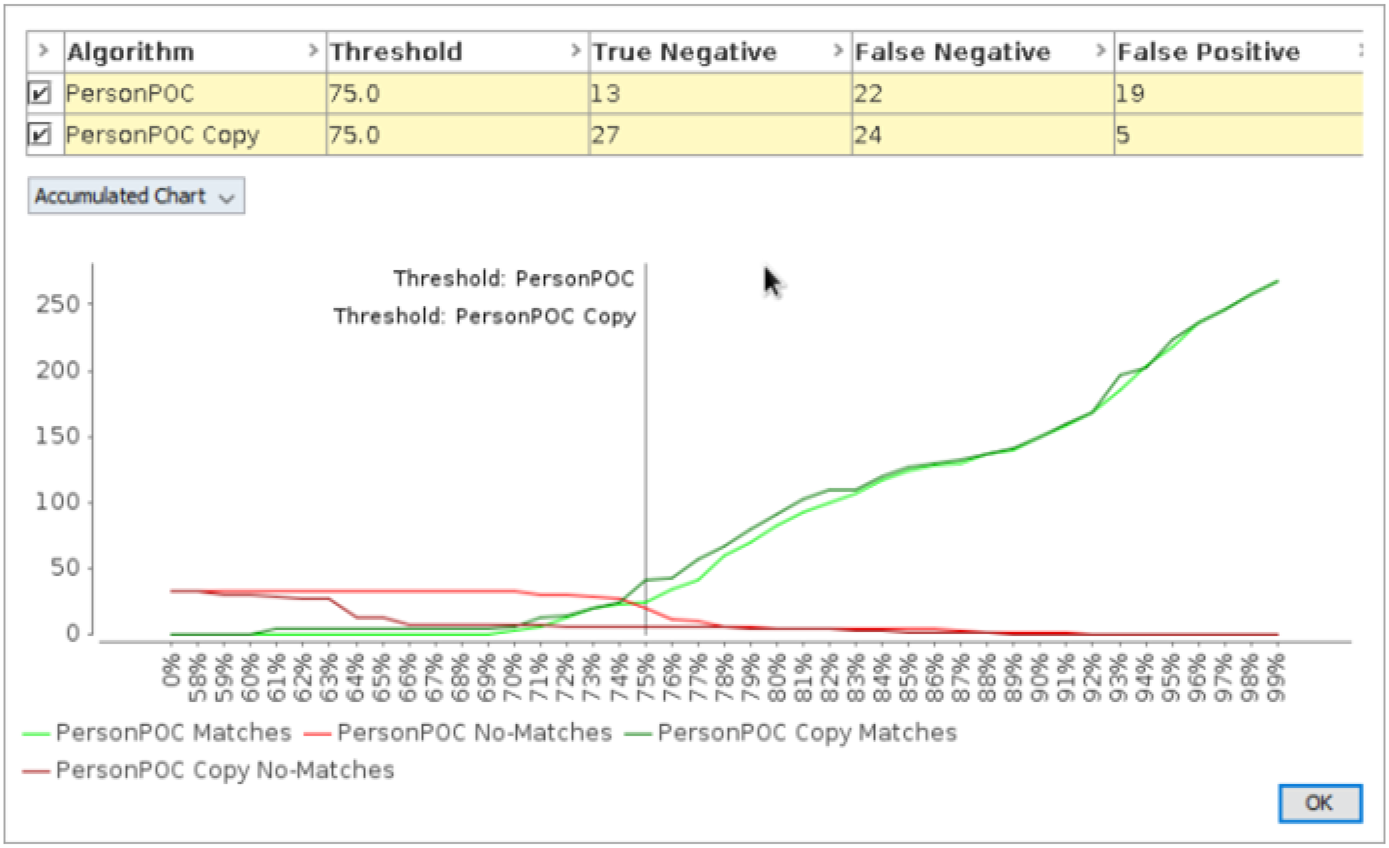
- Click a colored bar to display the Match Pair List dialog. This includes an extract of the corresponding data from the CSV file to allow inspection of the attribute values of the pairs.
- In the Match Pair List dialog, click the binocular button () to open the matching algorithm editor with the relevant pair selected in the System Setup tab. This allows investigation of the algorithm behavior for a given pair.



	Algorithm	Threshold	True Negative	False Negative	False Positive
☑	PersonPOC	75.0	13	22	19
☑	PersonPOC Copy	75.0	27	24	5

Accumulated Chart

The chart shows the accumulated score frequency for the algorithms. Manually classified matches are green and accumulate to the right of the threshold line. Manually classified no-matches are red and accumulate to the left. The accumulated chart is useful to compare the matching abilities of two algorithms because it is easy to evaluate the number of scores up to a certain point. The chart is also useful for identifying a good threshold value.



	Algorithm	Threshold	True Negative	False Negative	False Positive
☑	PersonPOC	75.0	13	22	19
☑	PersonPOC Copy	75.0	27	24	5

Matching Algorithm - Duplicates Tabs

The following tabs are legacy functionality but can help in tuning and monitoring an algorithm's results:

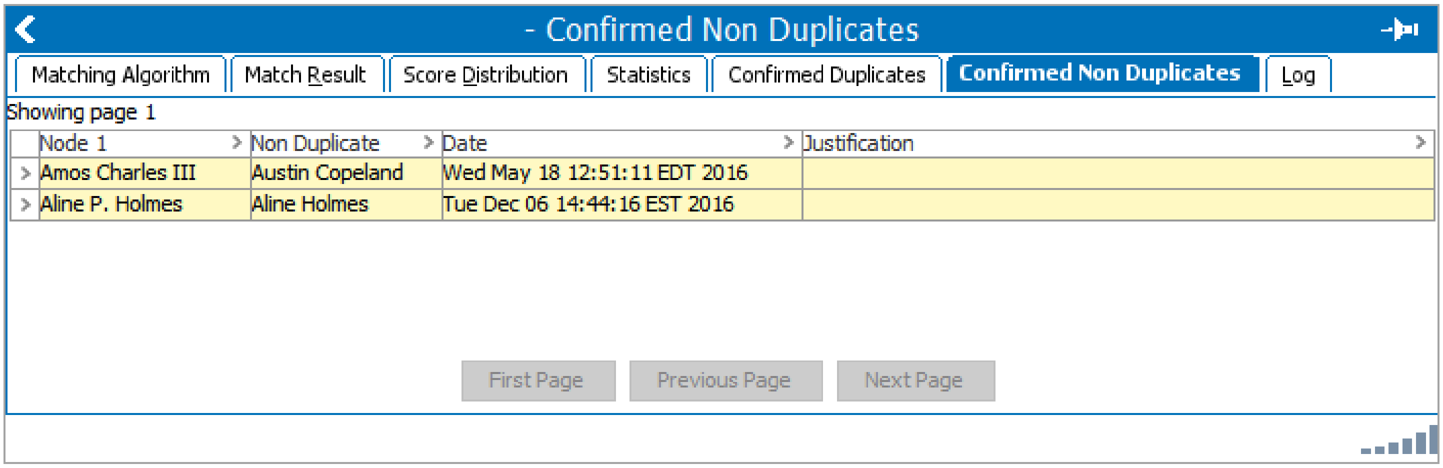
- Confirmed Duplicates Tab
- Confirmed Non Duplicates Tab

These tabs can be used along with the Match Result tab, as defined in the **Matching Algorithm - Match Result Tab** topic. For another tuning option, refer to the **Match Tuning** topic.

All confirmed duplicates and confirmed non duplicates are displayed in the appropriate tab in the workbench on the algorithm. For a Match and Merge solution, the duplicate reference is deleted by the merge operation, which means the 'Confirmed Duplicates' tab is almost always empty.

Records are identified as 'confirmed duplicates' or 'confirmed non duplicates' via a reference type selected in the Matching component model and in the **Duplicate Type** and **Non-Duplicate Type** parameters in the Matching Algorithm.

The specified reference type is added when a user manually reviews and confirms a match or non-match.

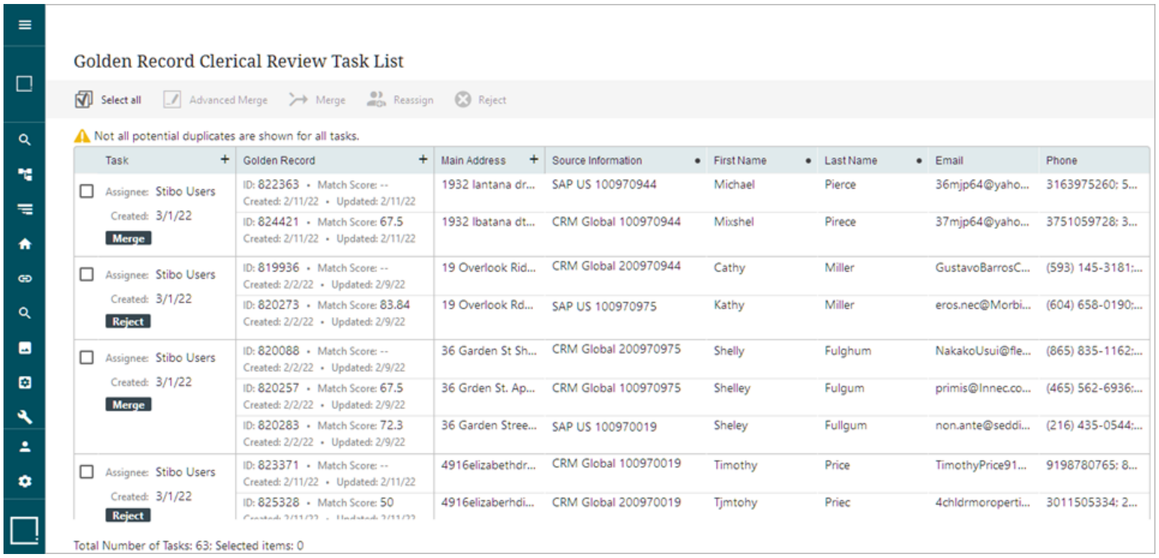


Showing page 1

Node 1	Non Duplicate	Date	Justification
Amos Charles III	Austin Copeland	Wed May 18 12:51:11 EDT 2016	
Aline P. Holmes	Aline Holmes	Tue Dec 06 14:44:16 EST 2016	

Machine Learning Match Recommendations

During clerical review for a match and merge solution, a data steward could face thousands of records that must be either merged or rejected. The Machine Learning Match Recommendations (MLMR) ease the workload by providing recommendations for merging or rejecting based on the data steward's previous decisions. This functionality works entirely on the Clerical Review Task List and does not influence the matching algorithm. When using the recommendations combined with the filtering and merge / reject bulk update capabilities, the data steward can resolve the task list more rapidly.



Golden Record Clerical Review Task List

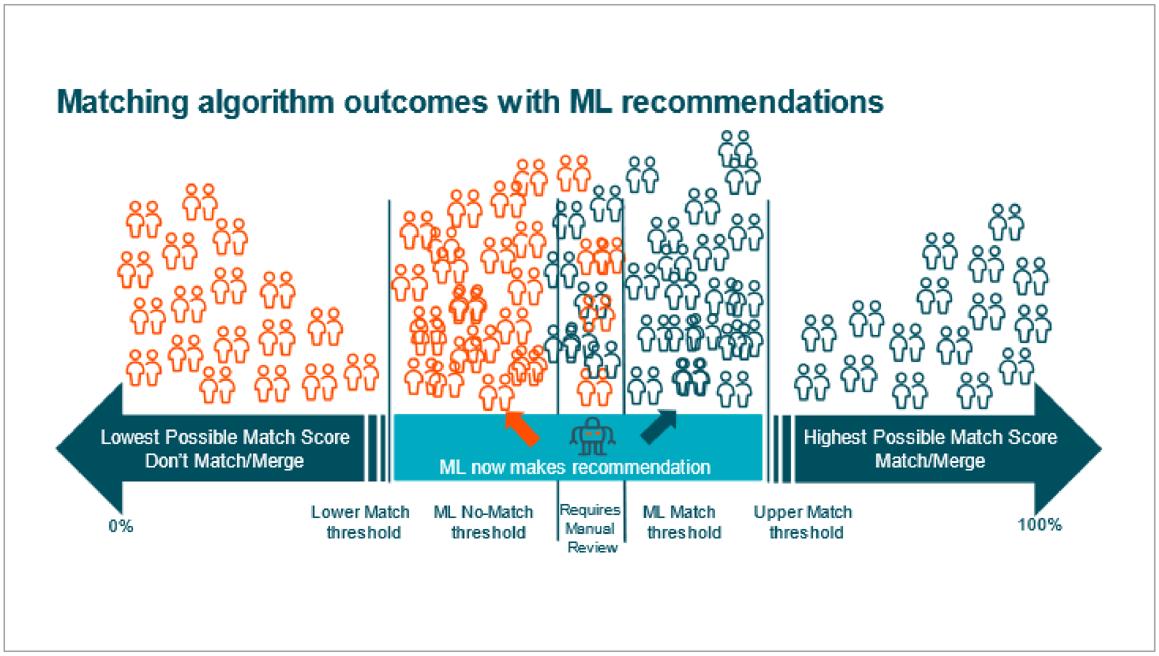
⚠ Not all potential duplicates are shown for all tasks.

	Task	Golden Record	Main Address	Source Information	First Name	Last Name	Email	Phone
□	Assignee: Stibo Users Created: 3/1/22 Merge	ID: 822363 • Match Score: -- Created: 2/11/22 • Updated: 2/11/22	1932 lantana dr...	SAP US 100970944	Michael	Pierce	36mjp64@yaho...	3163975260; 5...
		ID: 824421 • Match Score: 67.5 Created: 2/11/22 • Updated: 2/11/22	1932 lbatana dt...	CRM Global 100970944	Mixshel	Pirece	37mjp64@yaho...	3751059728; 3...
□	Assignee: Stibo Users Created: 3/1/22 Reject	ID: 819936 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	19 Overlook Rid...	CRM Global 200970944	Cathy	Miller	GustavoBarrosC...	(593) 145-3181;...
		ID: 820273 • Match Score: 83.84 Created: 2/2/22 • Updated: 2/9/22	19 Overlook Rd...	SAP US 100970975	Kathy	Miller	eros.nec@Morbi...	(604) 658-0190;...
□	Assignee: Stibo Users Created: 3/1/22 Merge	ID: 820088 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	36 Garden St Sh...	CRM Global 200970975	Shelly	Fulghum	NakakoUsui@fle...	(865) 835-1162;...
		ID: 820257 • Match Score: 67.5 Created: 2/2/22 • Updated: 2/9/22	36 Grden St. Ap...	CRM Global 100970975	Shelley	Fulgum	primis@Innec.co...	(465) 562-6936;...
		ID: 820283 • Match Score: 72.3 Created: 2/2/22 • Updated: 2/9/22	36 Garden Stree...	SAP US 100970019	Sheley	Fullgum	non.ante@seddi...	(216) 435-0544;...
□	Assignee: Stibo Users Created: 3/1/22 Reject	ID: 823371 • Match Score: -- Created: 2/11/22 • Updated: 2/11/22	4916elizabethdr...	CRM Global 100970019	Timothy	Price	TimothyPrice91...	9198780765; 8...
		ID: 825328 • Match Score: 50	4916elizaberhdi...	CRM Global 200970019	Tjmtohy	Priec	4chldrmoroperti...	3011505334; 2...

Total Number of Tasks: 63; Selected items: 0

The MLMR uses the data steward's merge / reject decisions within the Clerical Review Task List to train a machine-learning model based on those decisions that provides merge and reject recommendations as a label on each task, which the data steward can either heed or disregard.

Solution Overview



Within a matching algorithm, the user can create a matching agent. Once enabled, the matching agent collects merge and reject decisions made in the Clerical Review Task List, which are stored as a local copy. The ASPiRE environment then uses these decisions to train a machine-learning model using the matching agent data model configured on the matching algorithm. With the trained model, the ASPiRE environment produces merge and reject recommendations, which are shown in the Clerical Review Task List.

Note: The machine-learning model and recommendation algorithm runs as a multitenant microservice in ASPiRE and is maintained outside of the normal STEP release cycle. The MLMR feature is new and adjustments to the machine-learning algorithm will happen, which could lead to changes in the number of recommendations given in the Clerical Review Task List.

The matching agent will start providing the first recommendations after a minimum of 30 reject decisions and 30 merge decisions. After that, it continuously provides a new set of updated recommendations every time 10 percent more tasks are completed. Depending on the number of tasks in Clerical Review, it might take some time before the recommendations are shown in the Web UI. The training and recommendation process runs as background processes (BGP) that you can monitor in the workbench.

The number of recommendations provided depends on the decisions made by the data steward. If the decisions are very inconsistent, meaning that similar tasks are both merged and rejected, then it is likely that only few recommendations are given. On the contrary, if decisions are consistent for similar patterns in the data, then the matching agent gives more recommendations. In the beginning, when the data steward has made less than 200 – 300 decisions, the number of recommendations can vary from training to training but will stabilize over time as the data steward makes more decisions.

Note: When performing more than 20 merge or reject decisions in one operation, those decisions are not included as training data and have no influence on future recommendations.

The matching event processor updates new and changed tasks with a new merge / reject recommendation. This happens when an enabled matching agent exists that has successfully completed the training process.

The following topics outline the setup and function of the MLMR:

1. Configuring MLMR
2. Maintaining the MLMR Data Model
3. Matching Agents
4. Clerical Review Task List with Matching Agent Recommendations
5. MLMR Considerations
6. Matching Agent Support Guidelines ()

Configuring the MLMR

To use the Machine Learning Match Recommendations (MLMR), you must first configure several elements. The MLMR uses a gateway integration endpoint to communicate between a matching algorithm and the cloud on which the MLMR works.

Prerequisite

Implement a match and merge solution as defined in the **Configuring Match and Merge** topic.

> **Important:** For every matching algorithm, there can only be one matching agent data model.

Configuration

The following topics outline the configuration of the MLMR:

1. Configuring the Stibo Aspire REST Gateway
2. Configuring the Matching Agent Object Type
3. Configuring the Clerical Review Workflow for MLMR
4. Adding MLMR Recommendations to a Clerical Review Task

Configuring the Stibo Aspire REST Gateway

The STEP Workbench has a dedicated gateway to connect to the Aspire cloud environment, where the Machine Learning Match Recommendations (MLMR) sends user decisions from the Clerical Review Task List to train the data model to provide merge and reject recommendations to the data steward.

For general information on how to configure a gateway integration endpoint, refer to the **Gateway Integration Endpoints** topic in the **Data Exchange** documentation.

Prerequisites

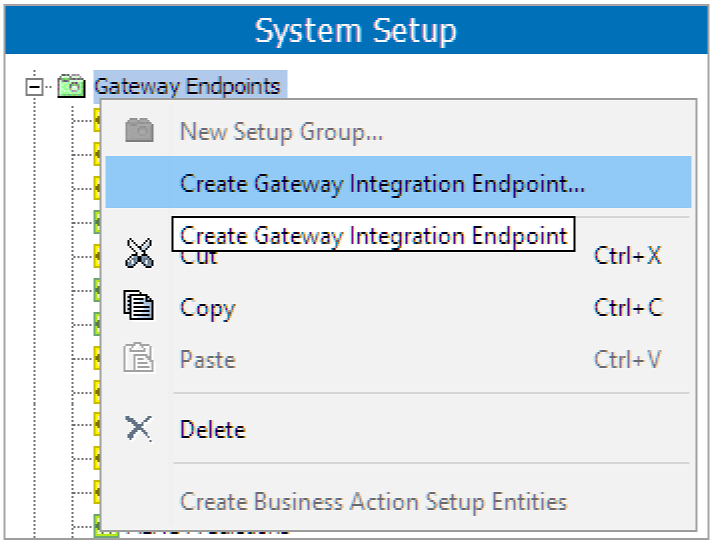
The environment your Stibo hosted STEP systems are deployed on determines the configuration of your URL and credentials for the Stibo Aspire gateway.

- **On-Premise Deployment** - Your URL and credentials must be acquired from Stibo Systems Support and entered into your sharedconfig.properties file manually. Refer to the section below for more information.
- **SaaS v1 Deployment** - Your URL and credentials must be acquired from Stibo Systems Support, which will then configure them for you. Refer to the section below for more information.
- **SaaS v2 Deployment** - Your URL and credentials for the Stibo Aspire gateway are automatically configured when creating the gateway. You do not need to contact Stibo Systems Support.

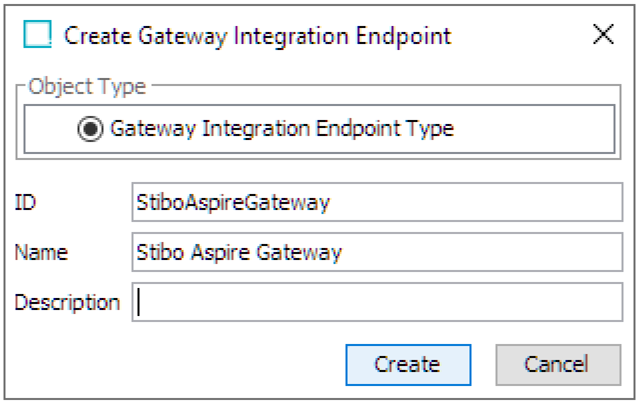
Configuring the Gateway

Follow these steps to configure the gateway:

1. Under System Setup in the workbench, navigate to the Gateway Endpoints node. Right-click and select 'Create Gateway Integration Endpoint.'


System Setup
Gateway Endpoints
New Setup Group...
Create Gateway Integration Endpoint...
Create Gateway Integration Endpoint
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete
Create Business Action Setup Entities

2. Enter an ID and Title. Click **Create**.



Create Gateway Integration Endpoint

Object Type

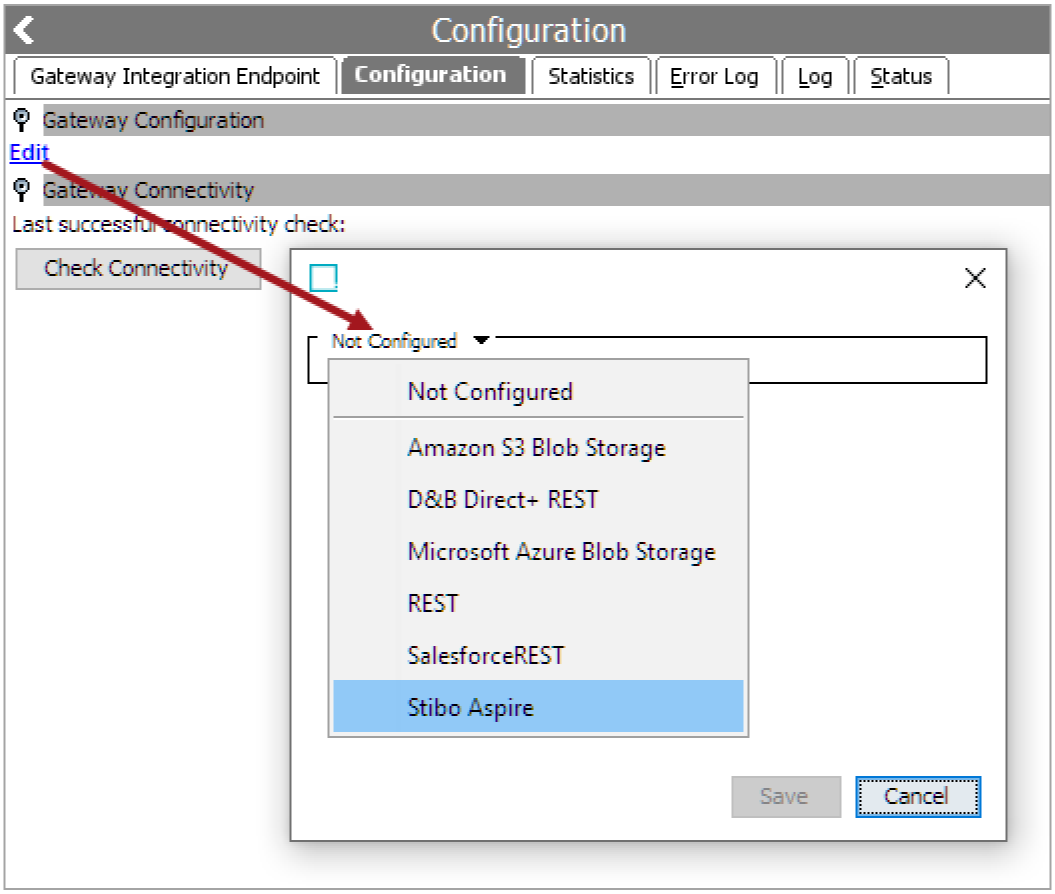
● Gateway Integration Endpoint Type

ID: StiboAspireGateway

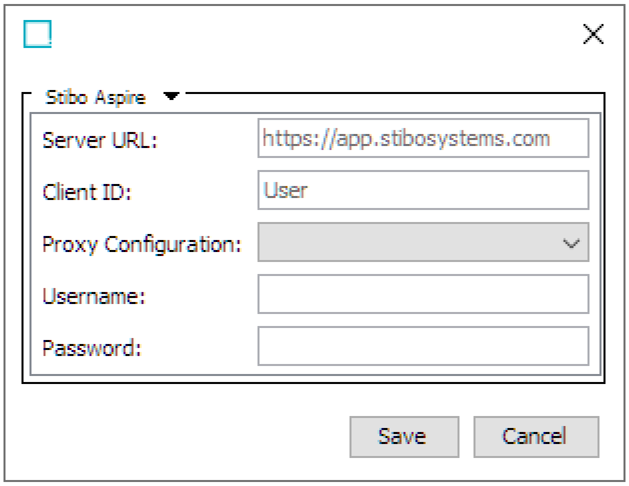
Name: Stibo Aspire Gateway

Description:

3. Under the Configuration tab, navigate below the 'Gateway Configuration' flipper and click **Edit**. From the dropdown, select 'Stibo Aspire.'



4. The Server URL and Client ID parameters should be automatically filled. If they are, click **Save**. If they are not, follow the instructions below:



Stibo Aspire

Server URL: https://app.stibosystems.com

Client ID: User

Proxy Configuration:

Username:

Password:

If your Server URL and Client ID are not pre-configured, you can obtain them by creating a ticket in the customer portal. Include the following information:

- Summary: MLMR - Requesting Aspire credentials for <customer name>
- Description: Provide Aspire credentials for the following system(s): <system1>, <system2>
- Issue Category: ML Matching Agent

If you have a SaaS v1 deployment, once Stibo Systems Support receives your issue, they will configure your sharedconfig.properties file for you.

If you have an on-premise deployment, Stibo Systems Support will provide the following properties which you must manually enter into your sharedconfig.properties file:

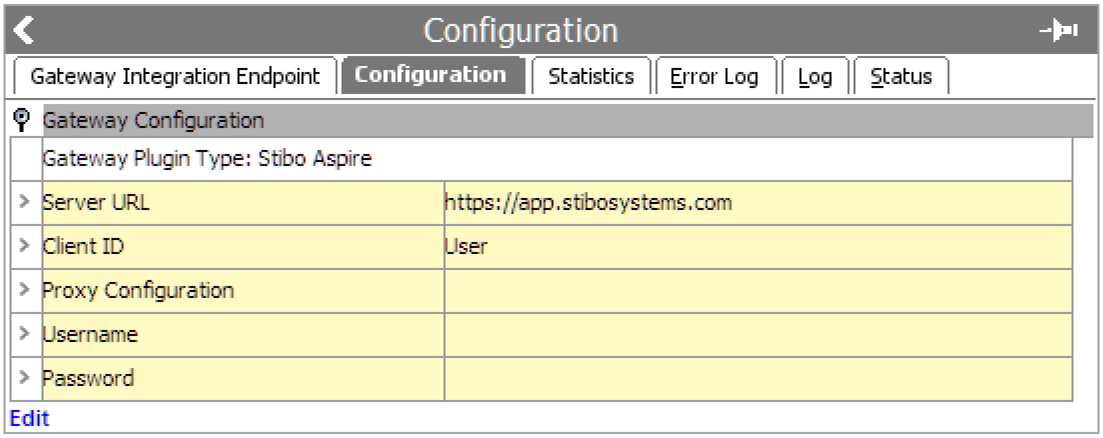
```
StiboAspireRESTGateway.ServerURL=serverURL
StiboAspireRESTGateway.Credentials=clientID,secret
```

Ensure you enter the correct Server URL and the Client ID and password provided by Stibo Systems.

Important: Restart the workbench before proceeding.

Optional: Select Proxy Configuration and type the username and password for the proxy server.

5. The gateway is now configured.

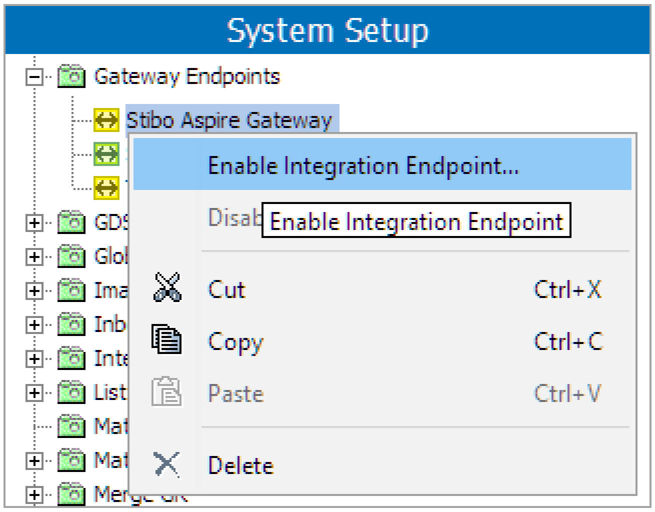


Gateway Configuration

Gateway Plugin Type: Stibo Aspire

Server URL	https://app.stibosystems.com
Client ID	User
Proxy Configuration	
Username	
Password	

Navigate back to the node under System Setup, right-click, and select 'Enable Integration Endpoint.'

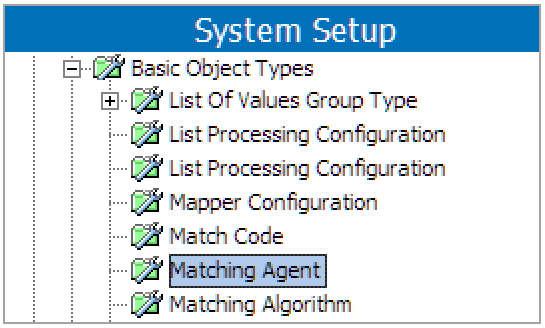


Configuring the Matching Agent Object Type

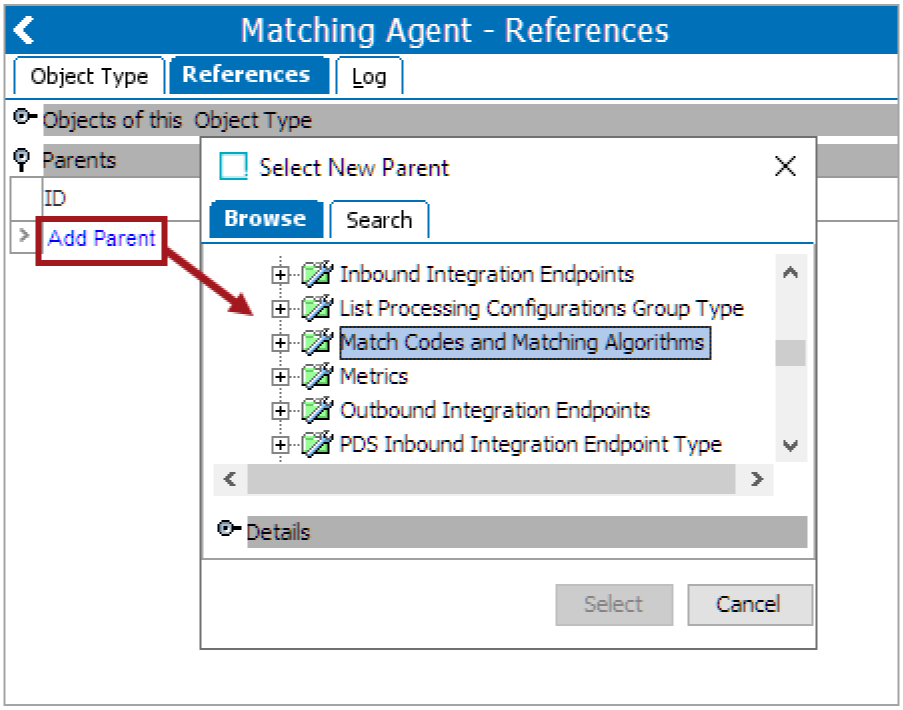
The matching agent stores the data steward's Clerical Review decisions and uses them to train a machine-learning model using a training background process (BGP). Based on the trained model, the matching agent provides merge and reject recommendations for all tasks in the Clerical Review Task List using a recommendation BGP.

The matching agent must be valid under one or more parent setup group(s).

1. In System Setup, navigate to **Object Types and Structures**, then **Basic Object Types**. Select **Matching Agent**.



2. Under the **References** tab, add the relevant setup group(s) as a parent.


Matching Agent - References
Object Type
References
Log
Objects of this Object Type
Parents
ID
Add Parent
Select New Parent
Browse
Search
Inbound Integration Endpoints
List Processing Configurations Group Type
Match Codes and Matching Algorithms
Metrics
Outbound Integration Endpoints
PDS Inbound Integration Endpoint Type
Details
Select
Cancel

Configuring the Clerical Review Workflow for MLMR

The Machine Learning Match Recommendations (MLMR) must store the merge / reject recommendations in a workflow variable with an attribute associated.

Prerequisites

Before the MLMR workflow variable can be configured, the golden record clerical review workflow must be configured. For more information, refer to the **Creating a Merge Golden Record Clerical Review Workflow** topic.

You must create an attribute to store the MatchingAgentRecommendation value on.

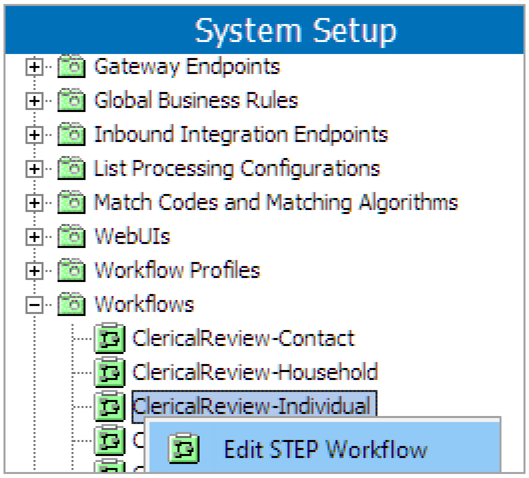
- ID: MatchingAgentRecommendation (user defined)
- Name: Matching Agent Recommendation (user defined)
- Validation Base Type: Text
- Multi Valued: No
- Attribute Type: Description
- Maximum Length: 100
- Validity: None
- Externally Maintained: Yes

Note: If you have already created the variable but not the attribute (e.g., if you upgraded from a previous STEP version to version 11.1 or higher), then once you add the attribute, you must initiate the 'Do Recommendations' action from System Setup to update all recommendations. For more information, refer to the **Matching Agents** topic.

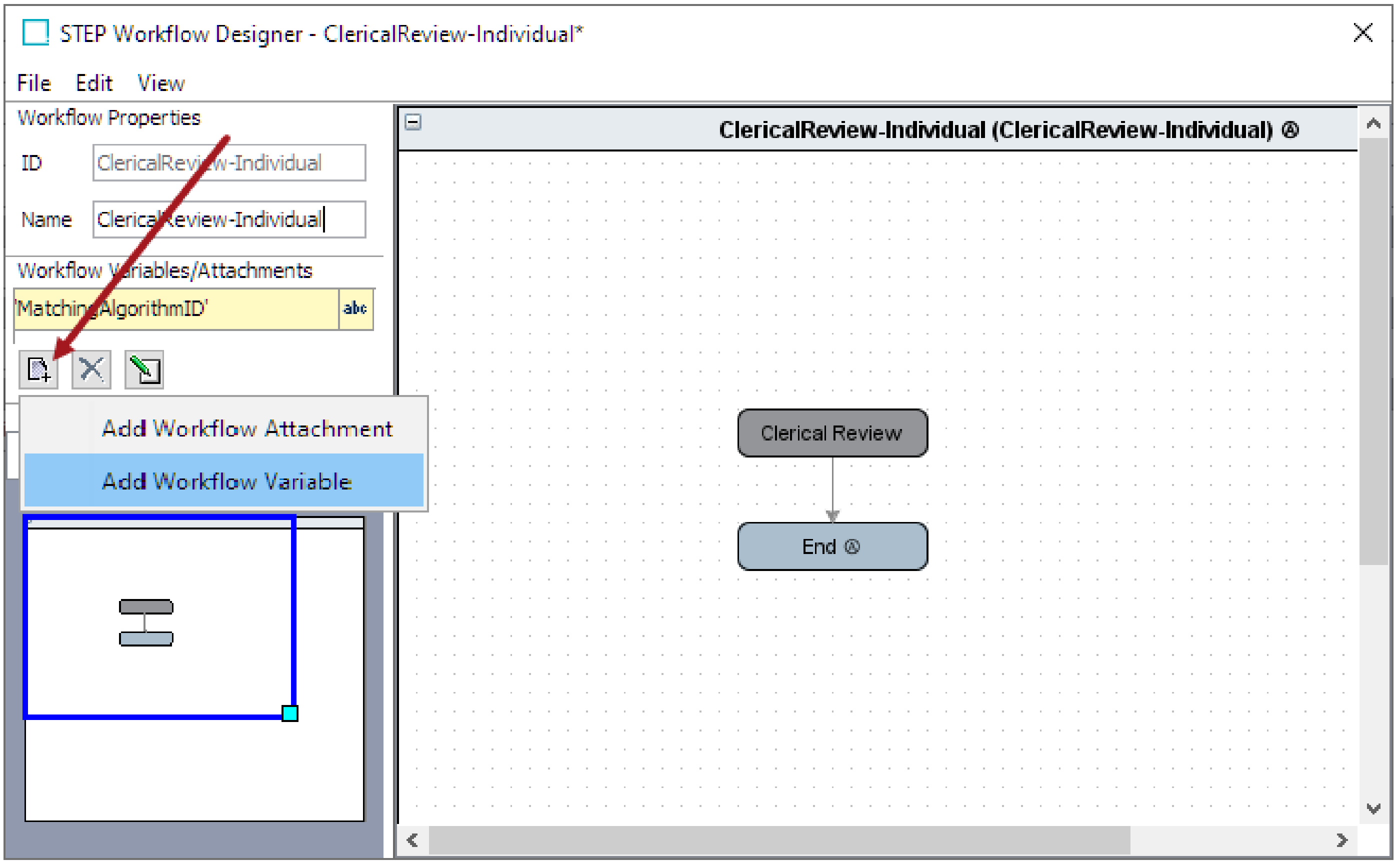
Configuring the Workflow

The following steps show how to modify the clerical review workflow for the MLMR:

1. On the System Setup tab in workbench, navigate to Workflows, then right-click on the desired clerical review workflow and select 'Edit STEP Workflow' to display the STEP Workflow Designer.



2. Click the add icon and select 'Add Workflow Variable.'



STEP Workflow Designer - ClericalReview-Individual*

Workflow Properties

ID: ClericalReview-Individual

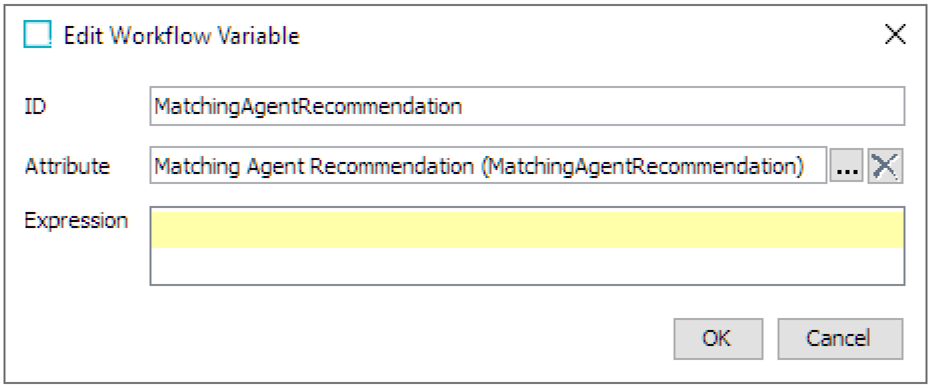
Name: ClericalReview-Individual

Workflow Variables/Attachments

'MatchingAlgorithmID'

3. In the Add Workflow Variable dialog, type in 'MatchingAgentRecommendation' for the ID.

   For the Attribute parameter, select the MatchingAgentRecommendation attribute created in the **Prerequisites** section.



ID: MatchingAgentRecommendation

Attribute: Matching Agent Recommendation (MatchingAgentRecommendation)

Expression:

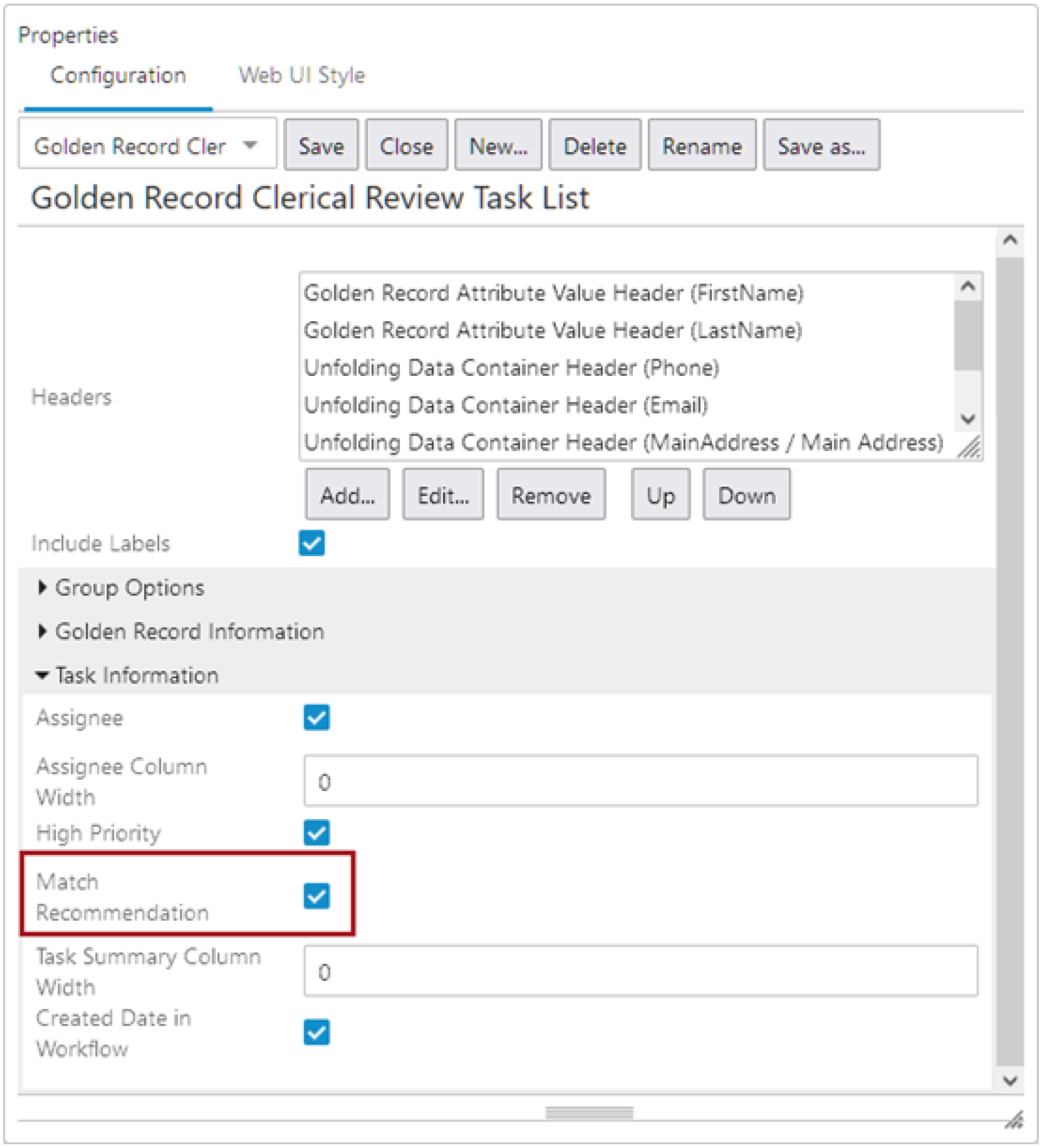
Click **OK**.

4. On the File menu, click **Save and exit** to close the STEP Workflow Designer dialog.

Adding Match Recommendations to a Clerical Review Task

Use these steps to configure an existing Clerical Review Task List for the Machine Learning Match Recommendations. To create a Clerical Review Task List, refer to the **Golden Record Clerical Review Task List** topic.

1. On your Golden Record Clerical Review Task List screen, open Design Mode.
2. On the properties dialog, open the 'Task Information' flipper and check **Match Recommendation**.



Golden Record Clerical Review Task List

Headers:
- Golden Record Attribute Value Header (FirstName)
- Golden Record Attribute Value Header (LastName)
- Unfolding Data Container Header (Phone)
- Unfolding Data Container Header (Email)
- Unfolding Data Container Header (MainAddress / Main Address)

Include Labels ☑

Assignee	☑
Assignee Column Width	0
High Priority	☑
Match Recommendation	☑
Task Summary Column Width	0
Created Date in Workflow	☑

Match recommendations now appear under the Task Information column.

Maintaining the MLMR Data Model

To function, the Machine Learning Match Recommendations (MLMR) must point to a matching algorithm. Within the matching algorithm, you must set up the data model, which the MLMR will use when making merge or reject recommendations.

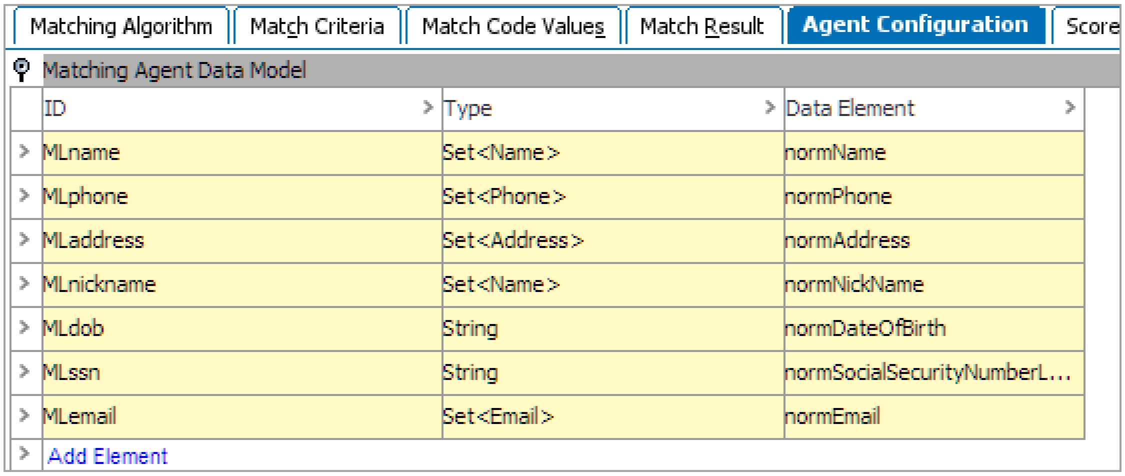
Prerequisites

Configure a matching algorithm and the Machine Learning Match Recommendations. For more information, refer to the **Configuring Matching Algorithms** topic and the **Configuring the Matching Agent Object Type** topic.

Setting up the Matching Agent Data Model

1. In the **System Setup** tab in workbench, select the desired matching algorithm, and go to the **Agent Configuration** tab.

2. Select the data elements that the matching agent must use to train the machine learning engine. This matching agent data model will be unique to the selected matching algorithm.



Matching Agent Data Model

ID	Type	Data Element
MLname	Set<Name>	normName
MLphone	Set<Phone>	normPhone
MLaddress	Set<Address>	normAddress
MLnickname	Set<Name>	normNickName
MLdob	String	normDateOfBirth
MLssn	String	normSocialSecurityNumberL...
MLemail	Set<Email>	normEmail
Add Element		

Important: For every matching algorithm, there can only be one matching agent data model.

The data elements that can be used are the ones configured on the Match Criteria tab. This way, existing data elements can be reused, or new specific ones can be added.

The matching agent supports data elements returning these types:

- String
- Set of strings

Additionally, the matching agent supports all party data matching normalizers:

- Address normalizer
- Email normalizer
- Organization name normalizer
- Person name normalizer
- Phone normalizer

Important: The attributes used in the data model must **not** be set to 'Externally Maintained.' This is required to store a correct copy of the clerical review decisions used for the machine learning training.

Data elements returning the same types as the normalizers listed above are supported. Data elements using data from a target reference are not supported, because this level of data is not stored at the point in time where the data steward makes merge / reject decisions in the clerical review task list.

For more information on normalizers and the Match Criteria tab, refer to the **Match Criteria Data Elements** topic.

Note: The organization name and person name type are not supported as sets. This means that only one element will be used in the matching agent data model. The matching agent data model will however show the type as Set<Name> and Set<Organizationname>

The user can change the data model at any time. When this is needed, the matching agent will be retrained. All previous merge / reject decisions are persisted on the matching agent, and the changed data model uses these decisions fully. For more information on how the matching agent works, refer to the **Matching Agents** topic.

Matching Agents for the MLMR

The matching agent stores the data steward's Clerical Review decisions and uses them to train a machine-learning model using a training background process (BGP). Based on the trained model, the matching agent provides merge and reject recommendations for all tasks in the Clerical Review Task List using a recommendation BGP.

These processes are described below.

Prerequisites

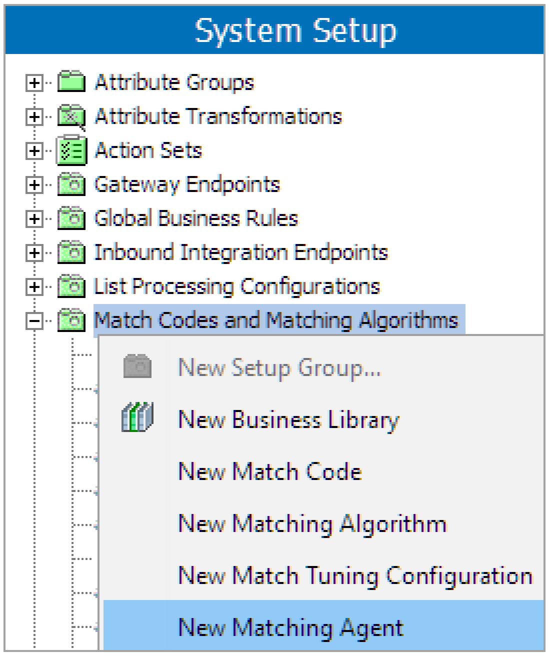
1. Identify or configure a matching algorithm on which the Machine Learning Match Recommendations (MLMR) can function. For more information, refer to the **Configuring Matching Algorithms** topic.

2. Configure the REST gateways for the matching agent. For more information, refer to the **Configuring the Stibo Aspire REST Gateway** topic.

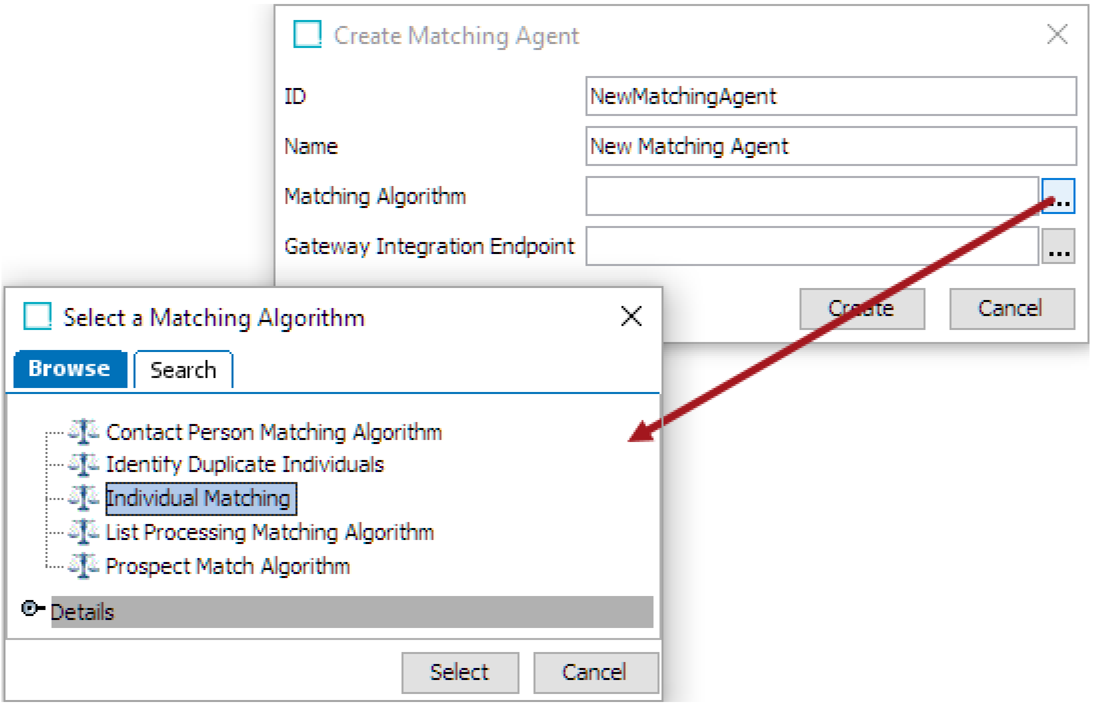
3. Configure the matching agent object type. For more information, refer to the **Configuring the Matching Agent Object Type** topic.

Create Matching Agents

1. In System Setup, navigate to the setup group used for matching algorithms, right-click the parent node, and select 'New Matching Agent.'



2. In the Create Matching Agent dialog, set the name and ID.
3. Click the ellipsis button (...) next to the **Matching Algorithm** parameter to select a matching algorithm.



Create Matching Agent

ID: NewMatchingAgent

Name: New Matching Agent

Matching Algorithm:

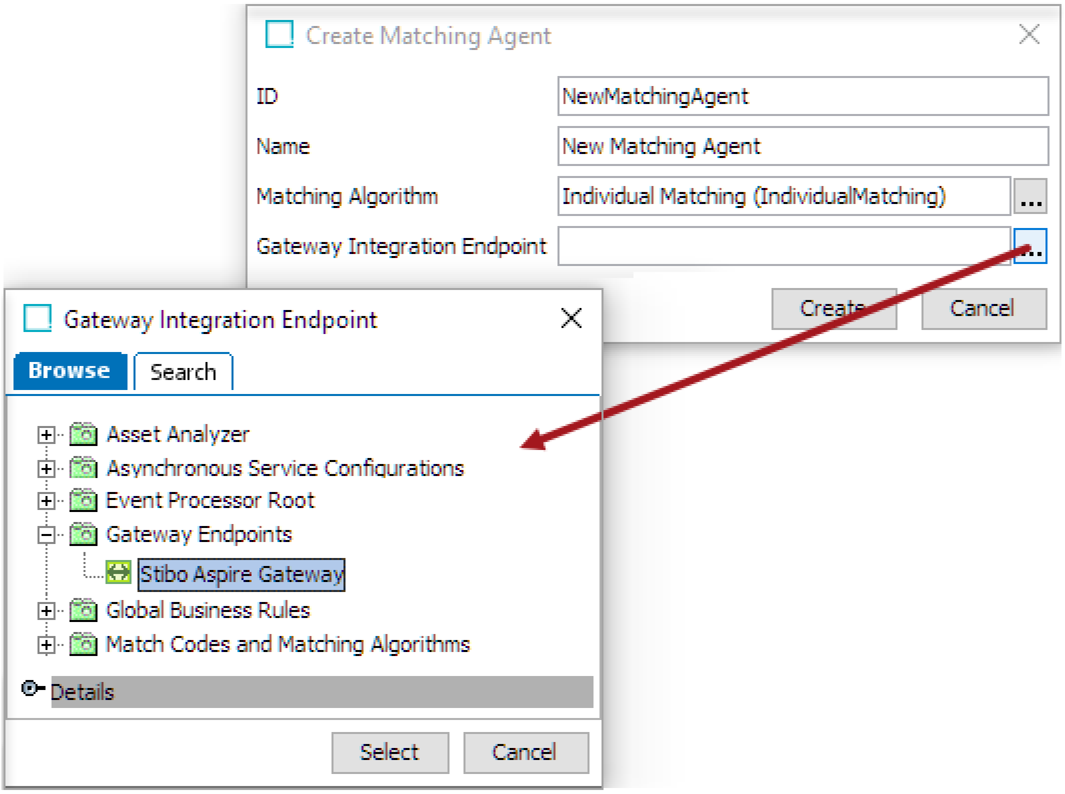
Gateway Integration Endpoint:

Select a Matching Algorithm

- Contact Person Matching Algorithm
- Identify Duplicate Individuals
- Individual Matching
- List Processing Matching Algorithm
- Prospect Match Algorithm

Note: Only matching algorithms using embedded match codes can be used.

4. For the Gateway Integration Endpoint parameter, click the ellipsis button (...) and select the **Stibo Aspire Gateway** endpoint.



Create Matching Agent

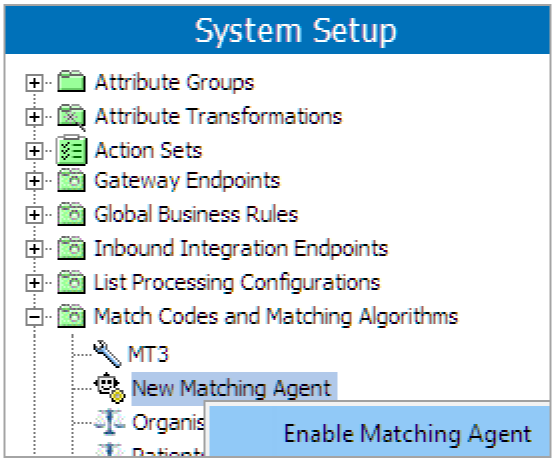
ID: NewMatchingAgent

Name: New Matching Agent

Matching Algorithm: Individual Matching (IndividualMatching)

Gateway Integration Endpoint:

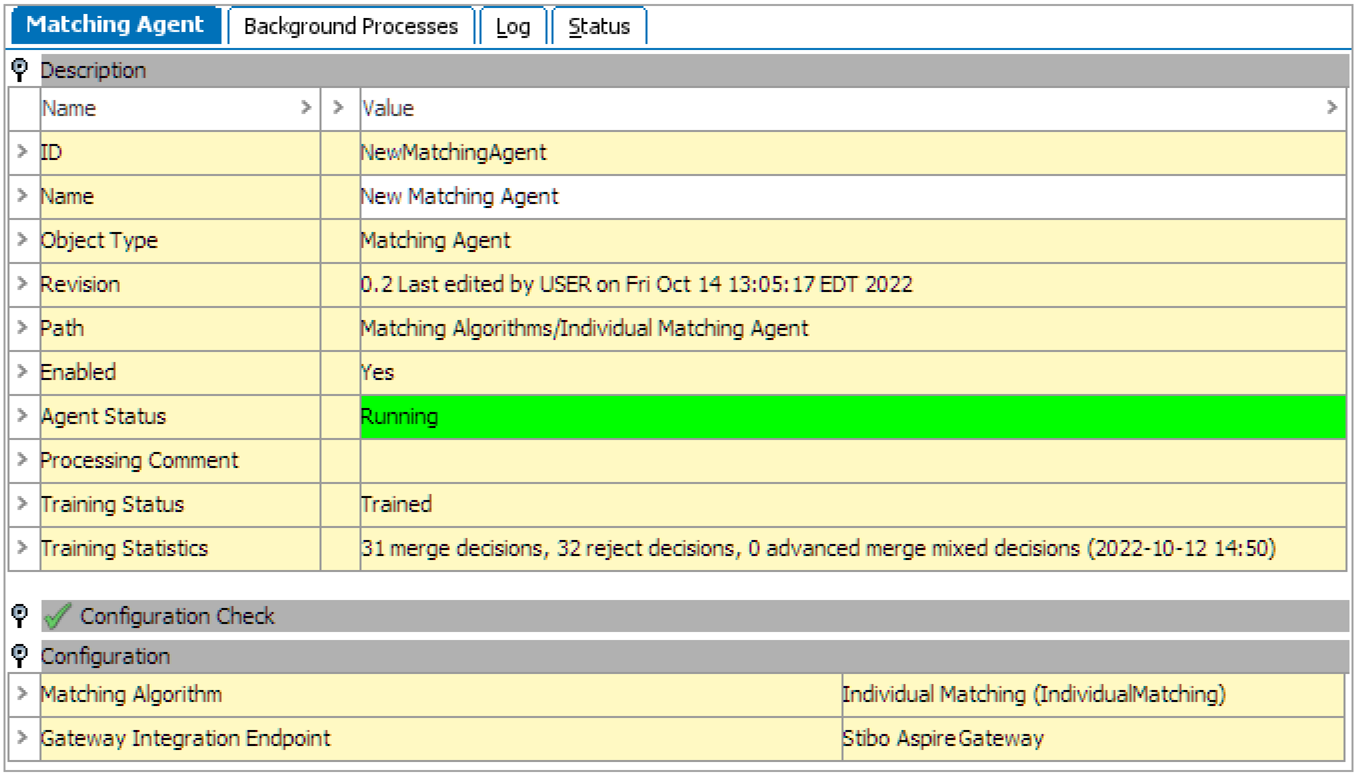
5. Right-click the matching agent and select **Enable Matching Agent**.



Note: Only one matching agent per matching algorithm can be enabled at a time. To use a different matching agent on the matching algorithm, first disable the active one, then enable the new one.

Using and Monitoring the Matching Agent

Once you have configured the matching agent, various statuses and statistics are available.



Description

Name	Value
ID	NewMatchingAgent
Name	New Matching Agent
Object Type	Matching Agent
Revision	0.2 Last edited by USER on Fri Oct 14 13:05:17 EDT 2022
Path	Matching Algorithms/Individual Matching Agent
Enabled	Yes
Agent Status	Running
Processing Comment	
Training Status	Trained
Training Statistics	31 merge decisions, 32 reject decisions, 0 advanced merge mixed decisions (2022-10-12 14:50)

✓ Configuration Check

Configuration

Matching Algorithm	Individual Matching (IndividualMatching)
Gateway Integration Endpoint	Stibo AspireGateway

- **Enabled** - Displays whether the matching agent is enabled or not. The matching agent collects clerical review decisions as long as it is enabled, regardless of what the Agent Status is.
- **Agent Status** - Displays whether the matching agent is 'Running,' 'Stopped,' 'Failed,' or 'Failed (retrying).' This status reflects the result of the training and recommendation BGPs.
  - **Running** – The matching agent is running.
  - **Stopped** – The matching agent has not yet been enabled or has been disabled by a user.
  - **Failed** - The matching agent has stopped because of a failure.
  - **Failed (retrying)** - When the matching agent runs the recommendation BGP and a FailAndRetryException error (or a connectivity error) is thrown, the matching agent enters a 'Failed (retrying)' state. This means that if the reason for the error should not cause the process to stop and prompt review by the user, like an issue with connectivity, the system will attempt to self-recover and restart the process when the issue is resolved. The logic behind the 'Failed (retrying)' state restarts the entire training or recommendation process when the matching agent is moved into that state. When the matching agent enters the 'Failed (retrying)' state, the system will attempt to retry the process every minute for 2 hours. Then the system will retry every 10th minute until it succeeds, is manually ended, or a month passes, at which point the process fails.
- **Processing Comment** - If the training or recommendation BGPs fail due to a connectivity error, the matching agent goes into the 'Failed (retrying)' state, and the Processing Comment displays a message with the first and last failures, the number of retries performed, and the next scheduled retry.

- **Training Status -** A matching agent is initially 'Untrained,' and has not yet provided recommendations. After its first successful training, the Training Status displays 'Trained.'
- **Training Statistics** - Displays statistics of the latest performed training including the total merge / reject and advanced merge decisions, as well as the time of training.

You can change the matching algorithm when the matching agent is disabled. If existing decisions are already stored on the matching agent, ensure that you use the same Golden Record object type. You can also change the gateway integration endpoint when the matching agent is disabled; however, under normal circumstances this is not necessary.

The matching agent performs certain BGPs to perform training and provide recommendations in the Clerical Review Task List.

Training process

The training BGP is responsible for using the data steward's Clerical Review decisions to train a machine-learning model. The BGP initially starts when the data steward makes a minimum of 30 merge and 30 reject decisions. Once complete, an increase of 10 percent more decisions will trigger a new training BGP. Once the training BGP finishes, the recommendation process automatically starts.

Recommendation process

The recommendation BGP processes the remaining Clerical Review tasks and provides merge or reject recommendations for them based on the trained machine-learning model.

Once the training and recommendation BGPs are complete, you can view recommendations in the Clerical Review Task List of your Web UI. For more information, refer to the **Adding Match Recommendations to a Clerical Review Task** topic.

On the recommendation BGP, you can download a details.zip attachment file. The file contains record pair details of the Clerical Review tasks with certain machine-learning related values. This is for Stibo Systems to analyze the merge / reject recommendations on the individual Clerical Review tasks, should the need arise.

Decision cleanup process

Clerical review merge / reject actions store a copy of merged and rejected golden records on the matching agent associated with the matching algorithm that owns the clerical review task. This information is used for training the matching agent.

When golden records are purged from STEP, any copies stored on matching agents will also be purged. This process of cleaning up the decision data is scheduled to run as a BGP every seventh day.

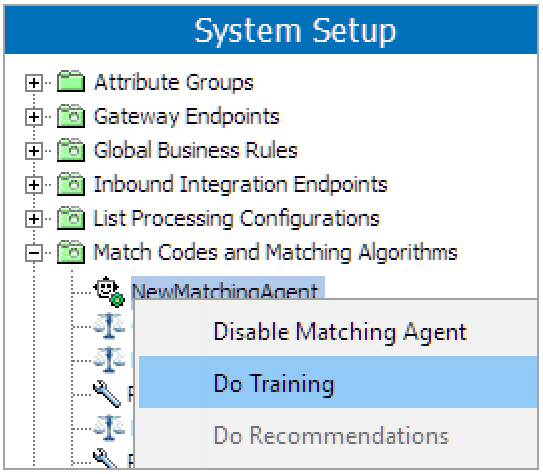
Important: Any future training BGPs will not contain purged decisions. As a result, the recommendations generated after the training process will be different because of the lack of this training data, and all previous merge / reject recommendations based on these purged decisions will be lost.

Manually Do Training

You can manually initiate the training process at any time; however, this should only be done in special scenarios. Stibo Systems does **not** recommend manual training.

Important: Manual training before the minimum required decisions have been made can result in less accurate recommendations.

To manually perform the training and get new up-to-date merge / reject recommendations, right-click the matching agent and select 'Do Training.'



After the training background process has finished, the recommendation background process automatically starts. Then the data steward begins receiving recommendations.

Once the matching agent is configured and training is complete, users can view recommendations on the Clerical Review Task List of their Web UI. For more information, refer to the **Adding Match Recommendations to a Clerical Review Task** topic.

Manually Do Recommendations

The matching agent has a right-click action 'Do Recommendations.' This manually starts the background process of getting merge / reject recommendations for all tasks in the Clerical Review Task List, based on the existing training. The Matching Event Processor ensures that all tasks are updated with an up-to-date recommendation. This action should only be used in special cases, such as if the recommendation process failed to provide recommendations on all tasks.

Note: After selecting the 'Do Training' action, the background process of getting merge / reject recommendations for all tasks in the Clerical Review Task List automatically starts, and the 'Do Recommendations' does not need to be manually selected.

Clerical Review Task List with Matching Agent Recommendations

The following assumes you have configured the Machine Learning Match Recommendations (MLMR) for clerical review. For more information, refer to the **Configuring MLMR** topic.

Clerical Review Decisions and Matching Agent Training

When a matching agent is configured and enabled, it actively tracks the data steward's merge / reject decisions. Since these decisions affect the quality of future recommendations, it is important that the user carefully considers the process around it. Examples of clerical review processes that will cause inaccurate and few recommendations:

- Merging and rejecting erroneous data on the golden records should be resolved on the golden records before resolving the task. This does not include normal outdated data and spelling differences.
- Rejecting tasks, for different reasons, through the golden records that are the same.

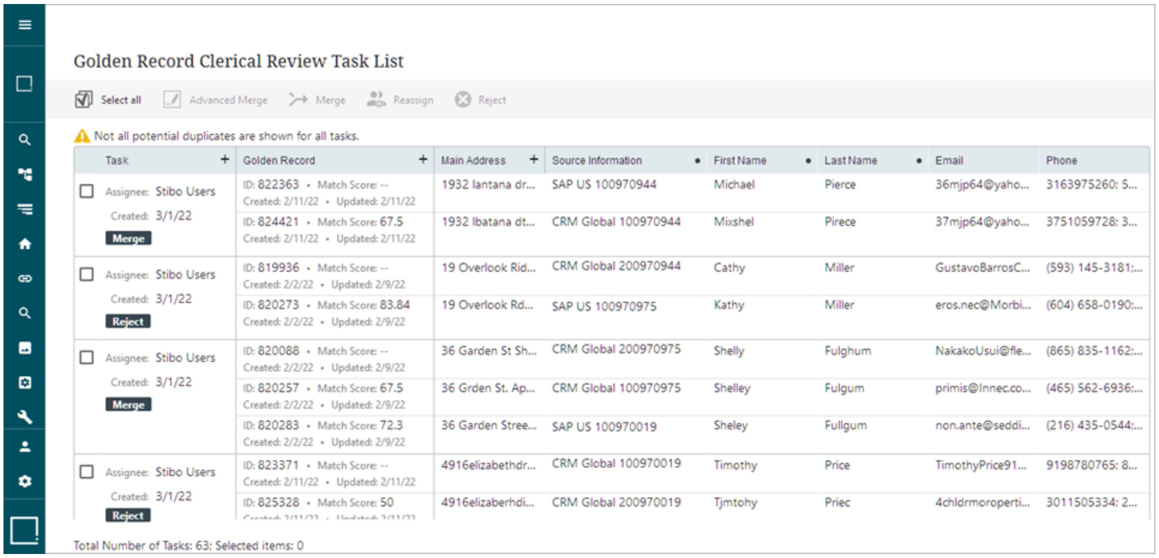
The minimum number of decisions the data steward must make is 30 merge and 30 reject decisions. Once the data steward completes this number of decisions on pair-only tasks, the recommendations will appear in the clerical review task list.

Recommendations in Clerical Review Task List

The matching agent continuously learns from the decisions the data steward makes over time. Whenever a certain percentage of additional decisions have been made, the matching agent retrains itself and updates all recommendations.

The recommendations are available as either merge or reject. If the matching agent does not recommend either, the cell will be left blank. The merge / reject recommendations are determined based on scores produced by the machine learning model and certain thresholds. For further questions about recommendations and their quality, contact Stibo Systems support.

Below is an example of how a golden record clerical review task list might appear with recommendations:



Golden Record Clerical Review Task List

⚠ Not all potential duplicates are shown for all tasks.

Task +	Golden Record +	Main Address +	Source Information ●	First Name ●	Last Name ●	Email	Phone
□ Assignee: Stibo Users Created: 3/1/22 Merge	ID: 822363 • Match Score: -- Created: 2/11/22 • Updated: 2/11/22	1932 lantana dr...	SAP US 100970944	Michael	Pierce	36mjp64@yaho...	3163975260; 5...
	ID: 824421 • Match Score: 67.5 Created: 2/11/22 • Updated: 2/11/22	1932 lbatana dt...	CRM Global 100970944	Mixshel	Pirece	37mjp64@yaho...	3751059728; 3...
□ Assignee: Stibo Users Created: 3/1/22 Reject	ID: 819936 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	19 Overlook Rid...	CRM Global 200970944	Cathy	Miller	GustavoBarrosC...	(593) 145-3181;...
	ID: 820273 • Match Score: 83.84 Created: 2/2/22 • Updated: 2/9/22	19 Overlook Rd...	SAP US 100970975	Kathy	Miller	eros.nec@Morbi...	(604) 658-0190;...
□ Assignee: Stibo Users Created: 3/1/22 Merge	ID: 820088 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	36 Garden St Sh...	CRM Global 200970975	Shelly	Fulghum	NakakoUsui@fle...	(865) 835-1162;...
	ID: 820257 • Match Score: 67.5 Created: 2/2/22 • Updated: 2/9/22	36 Grden St. Ap...	CRM Global 100970975	Shelley	Fulgum	primis@Innec.co...	(465) 562-6936;...
	ID: 820283 • Match Score: 72.3 Created: 2/2/22 • Updated: 2/9/22	36 Garden Stree...	SAP US 100970019	Sheley	Fullgum	non.ante@seddi...	(216) 435-0544;...
□ Assignee: Stibo Users Created: 3/1/22 Reject	ID: 823371 • Match Score: -- Created: 2/11/22 • Updated: 2/11/22	4916elizabethdr...	CRM Global 100970019	Timothy	Price	TimothyPrice91...	9198780765; 8...
	ID: 825328 • Match Score: 50 Created: 2/11/22 • Updated: 2/11/22	4916elizaberhdi...	CRM Global 200970019	Tjmtohy	Priec	4chldrmoroperti...	3011505334; 2...

Total Number of Tasks: 63; Selected items: 0

Note: Users can filter based on match recommendation. For more information on filtering, refer to the **Filtering Task List** section of the **Golden Record Clerical Review Task List** topic.

Furthermore, when you have selected 10 or more tasks (or when using the **Select All** button), merge / reject actions will run as bulk operations in a background process.

MLMR Considerations

When working with the Machine Learning Match Recommendations (MLMR), Stibo Systems recommends that the matching agent is configured to similar data points as the matching algorithm it is tied to. Like matching algorithm results, the matching agent recommendations are most accurate if high data quality is insured.

The number of recommendations provided depends on the decisions made by the data steward. If the decisions are very inconsistent, meaning that similar tasks are both merged and rejected, then it is likely that only a few recommendations will be given. On the contrary, if decisions are consistent for similar patterns in the data, then more recommendations are given. In the beginning, when the data steward has made less than 200 – 300 decisions, the number of recommendations will vary from training to training, but over time they will stabilize as the data steward makes more decisions.

All recommendations are evaluated by a novelty filter which will avoid giving recommendations on tasks that have characteristics which are unknown to the matching agent. This is intended to prevent recommendations from being generated for scenarios that have never been encountered and addressed by a data steward. It will be most active in the beginning, when the data steward has made a limited number of decisions. Additionally, it is active when users are working in silos and clerical review decisions are made in the data silos, e.g. a certain region where data has certain completeness. For example, when all existing decisions have been done for tasks lacking data for a social security number, the novelty filter will avoid giving any recommendation on tasks that actually have a social security number. Over time, when enough decisions have been made, most recommendations will bypass the novelty filter and thereby a larger set of recommendations will be provided.

Matching agents continue to learn based on data steward decisions. However, this presents the risk that too many inaccurate / inconsistent data steward decisions are considered by the matching agent. If the number of recommendations becomes low because of previous inconsistent decisions, you may need to start over and create a new matching agent.

Support Guidelines for the MLMR

The quality of recommendations provided by Machine Learning Match Recommendations (MLMR) in the Clerical Review Task List are dependent on the 'merge' and 'reject' decisions made by the user charged with training the matching agent. To improve the quality of the recommendations, Stibo Systems provides customers with a dedicated team prepared to engage in a collaborative process with customers to help improve the customer's understanding of the recommendations, and to improve the quality of those recommendations.

If the matching agent recommendations you receive results in questions for you or your team, in the Support Portal, create a ticket with the Issue Type 'Customer Request.' Find below a list of the fields required when creating a support issue for the MLMR and descriptions of how to provide the requested content.

> **Note:** Before you create a support issue, verify the relevant data to match on is mapped to the matching agent.

Summary: Add a short description of the issue you are experiencing in this field. Preface your summary content with ‘MLMR' so it is clear to the support team that the issue relates to the matching agent recommendations. The format will look like this: 'MLMR - <description of the issue>’.

Description: In this field, copy the data points listed below and paste it into the 'Description' field in the issue. Then add the requested information for each data point:

Description of problem:

Total number of recommendations:

Number of wrong recommendations identified:

Description of wrong recommendations: Describe what is wrong with the recommendations from the matching agent.

System Name / URL:

Training Process BGP ID:

Training Process BGP Started timestamp:

Training Process Execution Report:

<paste text>

Recommendation Process BGP ID:

Recommendation Process BGP Started timestamp:

Recommendation Process Execution Report:

<paste text>

Issue Category: ML Matching Agent

Business Domain: CMDM

Find Similar Web Services

Find Similar web services allow searching for potential duplicate records prior to creating new records, resulting in fewer duplicate objects in both source systems and STEP. Consider your requirements and then review the following table to determine the best way to implement Find Similar.

Complete documentation for web service functionality is at [system]/sdk or access the **STEP API Documentation** button on the Start Page

	getSimilarObjects	entities/find-similar
Web Service	Runs the match algorithm mentioned in the input record and outputs the found record or potential duplicates using the supplied XML output template.	Runs the match algorithm defined in the setup node and outputs the found record or potential duplicates using the XML output template defined in the endpoint setup node.
Setup	Core service, available at the same path on all systems.	Available if defined by Setup node in the System Setup.
API Style	SOAP On STEP API Documentation, access the 'Soap API documentation' link, and click the 'Core WebServices available in the STEP system' link to find details on: • getSimilarObjectsRequestType • getSimilarObjectsResponseType	REST On STEP API Documentation, access the link under the 'REST API V2' heading. Under the Entities section, click the POST button to find details and the 'Try it out' option for: • /entities/find-similar
Interfaces with	Web UI via: • Initiate Item Screen • Add Reverence Action	External services
Input	List of text strings which can be bound (via Node Binds) into the match algorithm, and then used as IDs for reference targets, attribute values, etc.	Entities - including data containers and references
Object Types	All super types (refer to the **Object Super Types** topic in the **Getting Started**	Entity

	getSimilarObjects	entities/find-similar
allowed	documentation)	
Matching Algorithm	Must include Node Binds for the input values. The same matching algorithm can be used for: - 'Duplicate Handler' on an 'Initiate Item' workflow screen in Web UI. - 'Find Similar' search on an 'Add Reference' action in Web UI. - 'GetSimilarObjects' SOAP web service request.	Standard matching algorithm
For more information	Refer to the **getSimilarObjects in SOAP API** topic	Refer to the **Find Similar in RESTv2 API** topic

getSimilarObjects in SOAP API

Before creating new objects in STEP, matching algorithms can be used to search for similar existing objects, ensuring duplicates are not created.

Matching logic can be applied to three different 'search before create' methods:

- The 'Duplicate Handler' on an 'Initiate Item' workflow screen in Web UI. For more information, refer to the **Initiate Item Screen** topic in **Web User Interfaces** documentation.
- The 'Find Similar' search on an 'Add Reference' action in Web UI. For more information, refer to the **Add Reference Action** topic in the **Web User Interfaces** documentation.
- 'getSimilarObjects' SOAP web service request. For more information, refer to the **Find Similar (getSimilarObjects)** topic in the **Customer MDM Solution Enablement** documentation.

For use case examples, refer to the **Find Similar Web Service** topic in **Customer MDM Solution Enablement** documentation.

The key to Find Similar getSimilarObjects functionality is the matching setup that the customer creates and uses for duplicate handling. Every time a user enters data into the search fields and clicks OK, the Find Similar search checks the match code values involved, executes the relevant matching algorithm, and provides a set of results, if any are found. If a user is not getting the expected results, one area to assess is the algorithm configured in the 'Duplicate Handler' parameter in the 'Add Reference Action' properties. Two bind types work with the Find Similar functionality:

1. First Match Object
2. Second Match Object

A relevant match code and matching algorithm needs to be set up before attempting to use the Find Similar Search tab. For more information about setting up and using matching algorithms, refer to the **Configuring Matching Algorithms** topic of this documentation.

The matching logic is applied by comparing potential new objects with that of existing objects. More specifically, match codes are generated for the incoming objects, compared to existing objects with similar match codes, and if matches are found, a list is returned of all matched objects with match scores (also called the 'rank score' in Web UI) that met or exceeded the configured threshold in the request. Using this list, the user can decide whether to create a new object or use an existing one.

Request

The 'getSimilarObjects' request defines the criteria for the match and the information to be returned. The following should be supplied in the call.

- **Access Context -** This parameter contains the username and password for the user accessing the system. It may optionally contain the context and workspace as well.
- **Values -** The values supplied are used by the matching engine for comparison. The property URL points to the URL of the attribute ID in the system that the value should be associated with for comparison.

- **Object Type URL** - This parameter is the URL of the object type in the system which will be used as a base for comparison.
- **Matching Algorithm URL** - The URL of the matching algorithm in the system to perform the comparison.
- **Export Configuration XML** - This optional section defines the information in XML format of the potential duplicates to be returned in the response. If excluded from the request, the STEP ID, STEP URL, Title, Super Type, Object Type URL, and Score will be returned. The records will be returned in order of highest score to lowest score.
- **Search Threshold** - The score threshold of potential duplicates to be returned. If the search threshold is 70, only records that match the supplied values with a score of 70 or above will be returned in the response. The Clerical Review Threshold and the Auto Threshold defined in the matching algorithm are ignored. Refer to the **Configuring Matching Algorithms** topic for details.
- **Max Count** - The maximum number of potential duplicates to return. If the matching algorithm identifies 100 records that score above the Search Threshold and the Max Count is set to 10, only the top 10 scoring records will be returned in the response.

Match Algorithm Configuration

The 'getSimilarObjects' request relies on the match algorithm to search. For a successful match, the match codes must exist and be up to date on the records and match criteria must be set up in the system. For information on a match code formula that can access the Find Similar values, refer to the **getSimilarObjects Node Binds** topic.

getSimilarObjects Configure Match Codes and Matching Algorithm

When configured to work with getSimilarObjects SOAP web service, special considerations should be made for the Find Similar solution's match codes and matching algorithm.

Important: A getSimilarObjects request (SOAP) can only return ninety-nine results at a time.

Match Codes

Attribute value binds in the match code definition should be created specifically for getSimilarObjects SOAP cases. For example:

```
if ( node == null ) {
//generate matchcodes for getSimilarObject
}
else {
//generate matchcodes for existing objects
}
```

Note: When used for getSimilarObjects, it is safe to establish large match code groups without impacting performance.

For more information, refer to the **Match Codes** topic.

Matching Algorithm

Matching algorithm global binds should be configured to map the attributes used in the SOAP request. 'Mcevaluate' / 'evaluate' should be used in the match criterion’s STEP function / JavaScript function to retrieve these values when the current object returns null.

Note: It is recommended practice to use the decision table match criteria for this purpose.

When decision tables are used as the match criteria, any configured party data normalizers require that all configured attributes also exist as global binds. This applies to both explicitly configured attributes and for those configured via component models.

So, for example, if the Address Normalizer is used for getSimilarObjects, the country, region, city, postcode, and street attributes must have corresponding global binds.

Important: For customer data normalizers, the name of the global bind must match the ID of the corresponding attribute.

Customer data normalizers are compatible with getSimilarObjects so long as global binds are set on the matching algorithm. For more information on global binds and match criteria, refer to the **Configuring Matching Algorithms** topic.

getSimilarObjects Node Binds

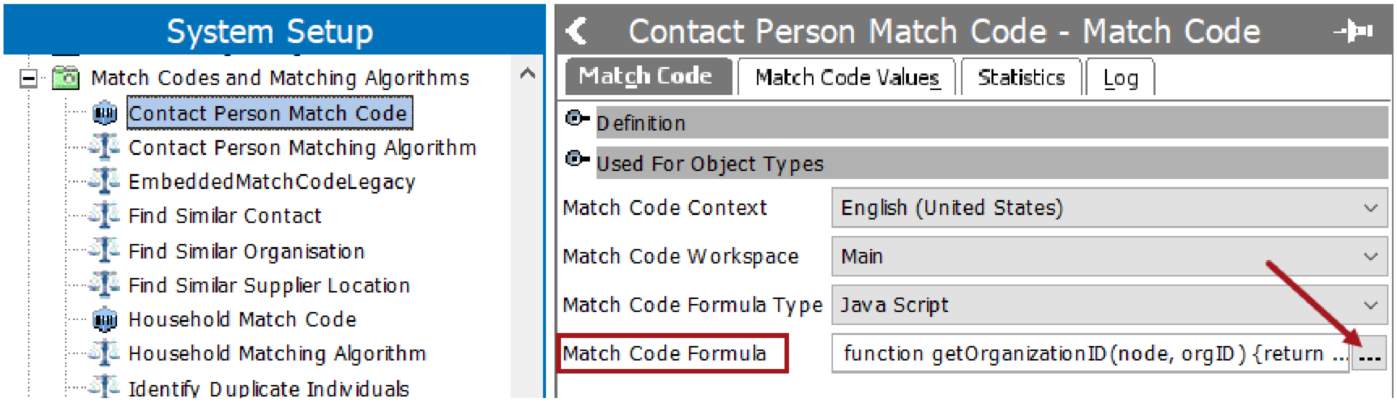
A node is a permanent STEP object. When a Find Similar getSimilarObjects call is made however, a node is not created in STEP. This non-permanent state means that the match code cannot obtain the 'Current Object,' and the matching algorithm cannot obtain the `first()` node via the 'Match Expression Context.' To return results, binds allow the matching engine to compare the values in the call with the values on the existing system nodes.

Binds associate incoming values to attributes in the system allowing the matching engine to make the appropriate comparisons. All values used in the match code should be defined under the binds flipper. Match codes should make use of the `if(node){}else{}` function for values not on the current object such as the reference being used in the code below.

Note: Using binds functionality for matching is not optimized for the In-Memory Database Component.

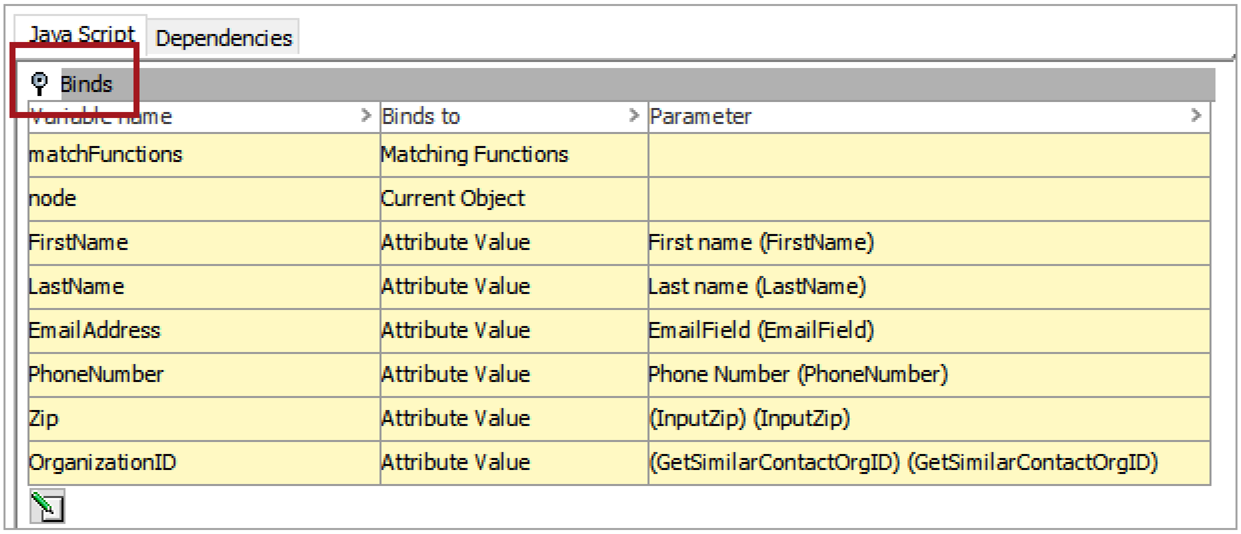
Configure a Match Code for the Core Web Service

1. In the Match Code object, on the Match Code tab, for the **Match Code Formula** parameter click the ellipsis button (...):


System Setup
Match Codes and Matching Algorithms
Contact Person Match Code
Contact Person Matching Algorithm
EmbeddedMatchCodeLegacy
Find Similar Contact
Find Similar Organisation
Find Similar Supplier Location
Household Match Code
Household Matching Algorithm
Identify Duplicate Individuals
Contact Person Match Code - Match Code
Match Code
Match Code Values
Statistics
Log
Definition
Used For Object Types
...

Match Code Context: English (United States)
Match Code Workspace: Main
Match Code Formula Type: Java Script
Match Code Formula: function getOrganizationID(node, orgID){return ...

- Open the Binds flipper and define all required binds. For information on JavaScript binds, refer to the **JavaScript Binds** topic in the online help **Resource Materials** documentation.

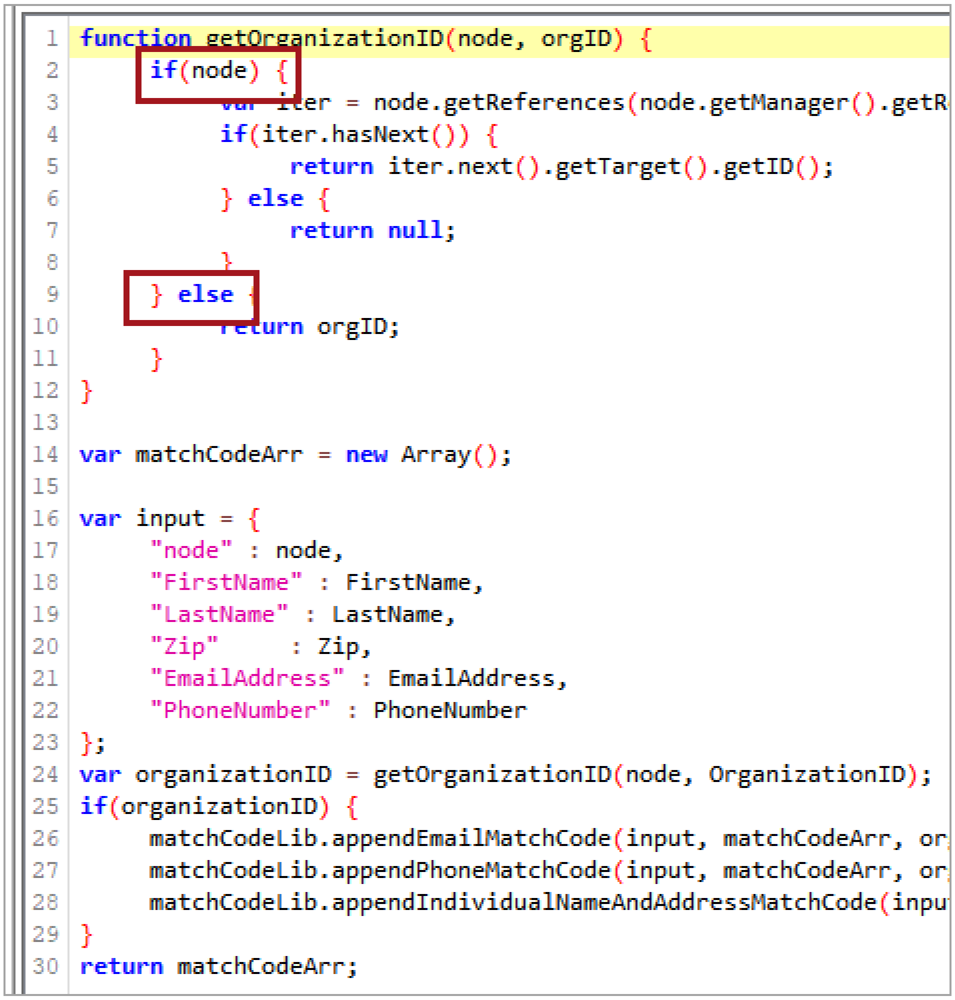


Java Script | Dependencies

Binds

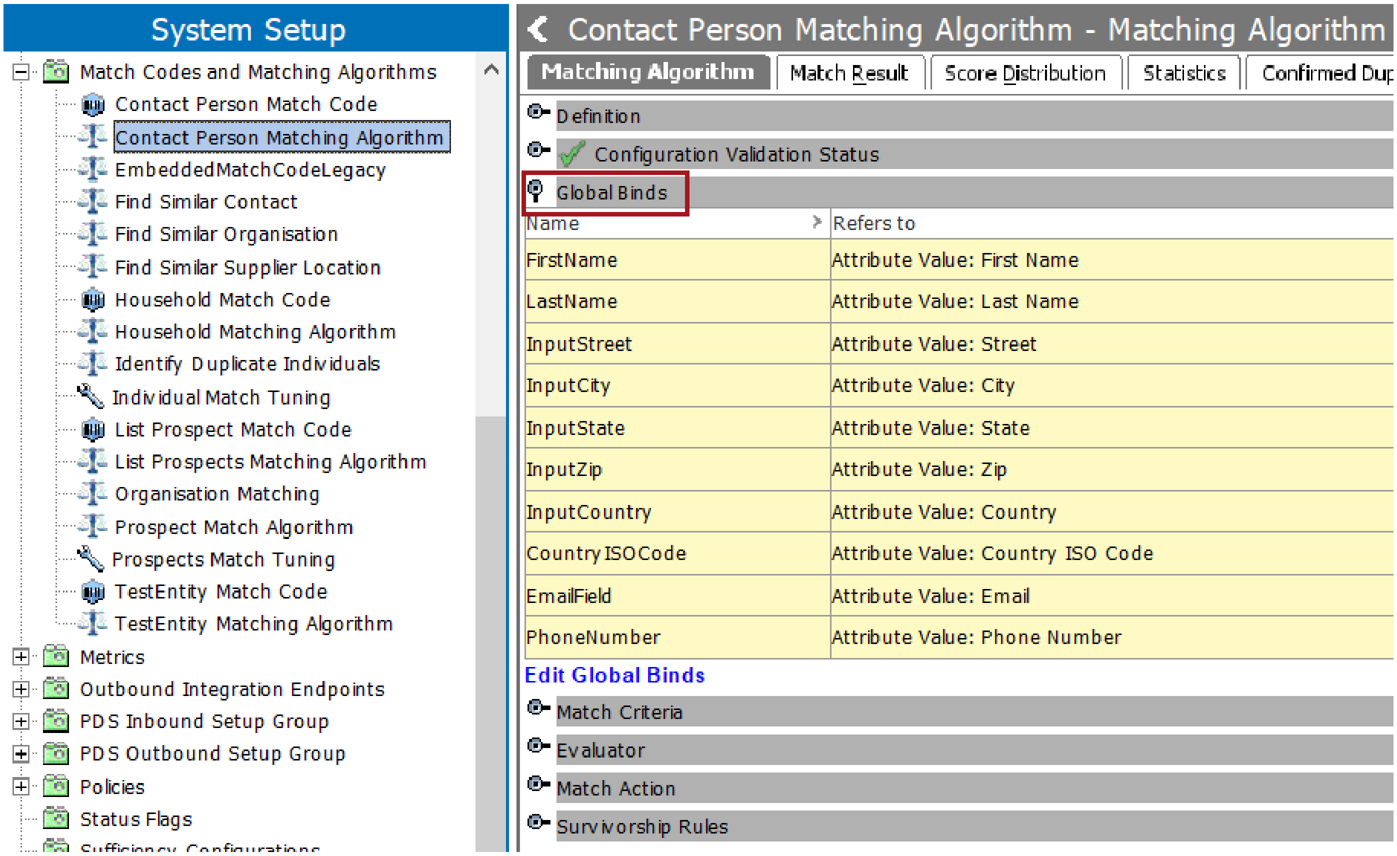
Variable name	Binds to	Parameter
matchFunctions	Matching Functions	
node	Current Object	
FirstName	Attribute Value	First name (FirstName)
LastName	Attribute Value	Last name (LastName)
EmailAddress	Attribute Value	EmailField (EmailField)
PhoneNumber	Attribute Value	Phone Number (PhoneNumber)
Zip	Attribute Value	(InputZip) (InputZip)
OrganizationID	Attribute Value	(GetSimilarContactOrgID) (GetSimilarContactOrgID)

- In the code section add the required JavaScript, using the `if(node){}else{}` function demonstrated below.



```
function getOrganizationID(node, orgID) {
     if(node) {
          var iter = node.getReferences(node.getManager().getR
          if(iter.hasNext()) {
               return iter.next().getTarget().getID();
          } else {
               return null;
          }
     } else {
          return orgID;
     }
}

var matchCodeArr = new Array();

var input = {
     "node" : node,
     "FirstName" : FirstName,
     "LastName" : LastName,
     "Zip"      : Zip,
     "EmailAddress" : EmailAddress,
     "PhoneNumber" : PhoneNumber
};
var organizationID = getOrganizationID(node, OrganizationID);
if(organizationID) {
     matchCodeLib.appendEmailMatchCode(input, matchCodeArr, or
     matchCodeLib.appendPhoneMatchCode(input, matchCodeArr, or
     matchCodeLib.appendIndividualNameAndAddressMatchCode(inpu
}
return matchCodeArr;
```

2. In the Matching Algorithm object, open the 'Global Binds' flipper and define all values used in the matching algorithm including explicitly configured attributes and those configured on the component model.



Name	Refers to
FirstName	Attribute Value: First Name
LastName	Attribute Value: Last Name
InputStreet	Attribute Value: Street
InputCity	Attribute Value: City
InputState	Attribute Value: State
InputZip	Attribute Value: Zip
InputCountry	Attribute Value: Country
CountryISOCode	Attribute Value: Country ISO Code
EmailField	Attribute Value: Email
PhoneNumber	Attribute Value: Phone Number

- When customer data normalizers are used in the matching algorithm, the name of the global bind must match the ID of the corresponding attribute.
- When the 'Address Normalizer' is used in the matching algorithm, the following attributes defined in the Address component model must be bound to the matching algorithm:
  - Input Street
  - Input City
  - Input State
  - Input Zip
  - Input Country
  - Country ISO Code

Find Similar in RESTv2 API

When integrating with external systems, the 'entities/find-similar' REST API V2 web service can prevent users from creating duplicate objects in source systems.

This functionality requires:

- A web service endpoint, as defined in the **Web Service Endpoint - Find Similar** topic in the **Data Exchange** documentation.
- A standard matching algorithm with match codes and match criteria, as defined in **Configuring Matching Algorithms** topic.

Complete documentation for web services functionality is at [system]/sdk or by accessing the **STEP API Documentation** button on the Start Page. On the STEP API Documentation page, click the link under the **REST API V2** heading.

For use case examples, refer to the **Find Similar Web Service** topic in **Customer MDM Solution Enablement** documentation.