



USER GUIDE

Matching, Linking, and Merging

Rel. 10.3-MP4 (March 29, 2022)

Table of Contents

Table of Contents	2
Matching, Linking, and Merging	8
Determine Match Criteria	8
Choose a Match Action	8
Identify Duplicates	8
Match and Link	8
Match and Merge	9
Perform Match Tuning	9
Match Criteria	11
Match Criteria Tab	11
Legacy Match Criteria Without Embedded Match Codes	12
Initial Setup for Matching Algorithms	14
Create Setup Group Type for Matching Algorithms	14
Link Matching Algorithms Object Types to Setup Group Types	17
Create a Matching Algorithm Setup Group	18
Configuring Matching Algorithms	21
Configuration	21
Match Codes	27
Match Code Values	28
Creating Match Code Values	29
Evaluator	29
Configuring a Match Code Generator	31
Business Function Match Code Generator	34

Address Match Code Generator	34
Email Match Code Generator	34
Natural Key Match Code Generator	35
Organization Name and Address Match Code Generator	36
Person Name and Address Match Code Generator	36
Phone Match Code Generator	38
Configuring a Match Code Filter	39
Selecting Match Codes	42
Data Profile Analysis Example	42
Observations	42
Match Code Strategy Options	45
Matcher Strategy Options	46
Match Criteria Data Elements	47
Configuration	48
Standard Data Elements	49
Attribute Value	49
Business Function Normalizer	50
STEP Function	51
JavaScript Function	52
Party Data Normalizers	52
Address Normalizer	52
Email Normalizer	54
Organization Name Normalizer	55
Person Name Normalizer	56
Phone Normalizer	58
Words Normalizer	59

Customer Data JavaScript Normalizers ..	60
Match Criteria Matchers	62
Configuration	62
Standard Matchers	64
Business Function Matcher	64
JavaScript Function Matcher	67
STEP Function Matcher	67
Customer Data Matchers	68
Address Matcher	68
Email Matcher	70
Organization Name Matcher	71
Person Name Matcher	73
Phone Matcher	76
Word Matcher	77
Extending Customer Data Matchers With JavaScript	80
Match Criteria Rules	81
Configuration	82
Legacy Match Criteria Without Embedded Match Codes	84
Window Size	84
Legacy Match Criteria Options	85
String Comparison Algorithms	85
Multi Word Damerau-Levenshtein Distance	85
Number Distance	85
JavaScript	86
Configuring a Legacy External Match Code	87
JavaScript Formula Type	89

Calculated Formula Type	91
Customizing Match Criteria with JavaScript Functions	93
normalizeValue	93
normalizeStreet	93
Core Matching Functions	95
Match Actions	97
Identify Duplicates	98
Configuration	98
Identify Duplicates in Workbench	98
Identify Duplicates in Web UI	98
Match and Link	100
Data Model	100
Match Score	101
Information Flow	102
Internal Data Source Objects	103
User Actions	104
Configuring Match and Link	105
Configure a Match and Link Solution	105
Configuring Matching Component Model	106
Configuration	106
Configuring the 'Link Golden Record' Object Type	109
Configuring the Matching - Link Golden Record Component Model	112
Configuring the Link Golden Record Match Action	114
Configuration	114
Survivorship in Match and Link	117

Trusted Source	117
Most Recent	118
Business Action Rule	118
Configuring the Link Event Processor	119
Configuration	119
Match and Link in Workbench	121
Compare Match Result	121
Adding Additional Matching Algorithm ...	123
Confirm or Reject a Duplicate	124
View Matched Objects in Tree	125
Merge Confirmed Duplicates	126
Match and Link in Web UI	129
Configuring a Deduplication Clerical Review	130
Create and Configure the Deduplication List	130
Specify Node Details Buttons	134
Create a Task List	134
Specify Mappings	136
Golden Record Linked Members Component	140
Configuration	140
Using Action Buttons	142
Merging Confirmed Matches	144
Configuration	144
Add Merge Nodes Screen	144
Map Merge Node Screen	146
Add Merge Confirmed Match From Grid Action	147

Performing a Merge	148
Potential Duplicates List	152
Configuring the Deduplication Table	152
Match and Merge	157
Data Model	157
Information Flow	157
Match and Merge Traceability	161
Configure Traceability in Web UI	161
Golden Record Source Information	161
Golden Record Source Traceability Screen	162
View Traceability in Workbench	162
Examples	162
Storing Source Records for Golden Records	164
Storage Functionality	165
Configuration	165
Maintenance	166
Configuring Match and Merge	167
Configure a Merge Solution	167
Configuring Matching Component Model	169
Configuration	169
Configuring the Merge Golden Record Object Type	172
Configuring the Matching - Merge Golden Record Component Model	175
Prerequisites	175
Configuration	177
Match Criteria	180

Match Criteria Tab	180
Legacy Match Criteria Without Embedded Match Codes	181
Match and Merge Clerical Review - Merge	183
Creating a Merge Golden Record Clerical Review Workflow	184
Configuration	184
Considerations	185
Match and Merge Clerical Review - Unmerge	186
Unmerging Golden Records	189
Configuration	190
Creating an Unmerge Golden Record Clerical Review Workflow	192
Configuration	193
Creating Merge Golden Record Match Action Handlers	197
Configuring Merge Golden Record Match Action	199
Configuration	199
Survivorship in Match and Merge	202
Configuring the Match Data Exchange Method	203
Asynchronous Merge Inbound Integration Endpoint	203
Synchronous Match and Merge Web Service Endpoint	204
Configuring the Merge Event Processor	206
Configuration	206
Configuring and Using Match and Merge in Web UI	208

Golden Record Clerical Review Task List	213
Configuration	213
Using Action Buttons	217
Golden Record Advanced Merge Dialog	220
Configuration	221
Header Configuration	223
Using the View and Buttons	229
Golden Record Source Traceability Screen	232
Value Traceability Popup	233
Configuration	234
Configuring the Unmerge Wizard	236
Configuration	236
Using the Unmerge Wizard	248
Manage Source Records and Golden Records	248
Moving a Source Record	251
Reactivating a Golden Record	252
Finalizing an Unmerge	253
List Processing Deduplicate Records	257
Golden Records Survivorship Rules	258
Survivorship in Match and Link	259
Trusted Source	259
Most Recent	260
Business Action Rule	260
Survivorship in Match and Merge	261
Survivorship in Match and Merge - Business Actions	262
Match and Merge Web Service Endpoint	263

Survivorship in Match and Merge - Unmerge	264
Updating a Golden Record Created through Unmerging	264
Using Survivorship Rules within the Unmerge Process	264
Unmerging a Golden Record from Another Golden Record	265
Configuring Survivorship Rules	266
Configuration	266
Considerations	268
Survivorship Business Action Rule	270
Business Action	270
Survivorship Data Container Rules	271
Data Container: Most Recent	271
Data Container: Trusted Source	272
Survivorship Name Rules	275
Name: Most Recent	275
Name: Multi Context Trusted Source	275
Name: Trusted Source	276
Survivorship Reference Rules	278
Reference: Most Recent	278
Reference: Multi Context Trusted Source	279
Reference: Trusted Source	281
Survivorship Value Rules	283
Value Default: Most Recent	283
Value Default: Trusted Source	283
Value: Most Recent	284
Value: Multi Context Trusted Source	286

Value: Trusted Source	287
Creating a Merge Keep First Handler	289
Match Tuning	291
Match Tuning Configuration	291
Match Results Tab Use	291
Initial Setup for Match Tuning	292
Create Setup Group Type for Match Tuning	293
Link Match Tuning Object Types to Setup Group Types	295
Create a Match Tuning Setup Group	296
Configuring Match Tuning	299
Configuration	300
Matching Algorithm - Match Result Tab	305
Pair Export	307
Pair Import Confirmed	308
Pair Export Confirmed	309
Matching Algorithm - Duplicates Tabs	315
Machine Learning Matching Agent	316
Solution Overview	317
ASPiRE License	317
Configuring MLMA	318
Configuration	318
Configuring the MLMA REST Gateways	319
Configure Stibo Systems Cloud Authentication Gateway	319
Configure Authentication Business Function	321
Configure Stibo Systems Cloud Service REST Gateway	324

Configuring the Matching Agent Object Type	327
Configuring the Clerical Review Workflow for MLMA	328
Prerequisites	328
Configuring the Workflow	328
Adding MLMA Recommendations to a Clerical Review Task	330
Maintaining the MLMA Data Model	333
Prerequisites	333
Setting up the Matching Agent Data Model	333
Matching Agents	335
Prerequisites	335
Create Matching Agents	335
Do Training	338
Manually Do Recommendations	339
Clerical Review Task List with Matching Agent Recommendations	340
Clerical Review Decisions and Match Agent Training	340
Recommendations in Clerical Review Task List	340
Find Similar	342
Find Similar - getSimilarObjects	344
Request	344
Match Algorithm Configuration	345
Find Similar getSimilarObjects Configure Match Codes and Matching Algorithm	346
Match Codes	346
Matching Algorithm	346

Find Similar getSimilarObjects Node Binds	348
Configure a Match Code for the Core Web Service	348
Find Similar getSimilarObjects on Initiate Item Screen	351
Using Find Similar on Initiate Item Screen	351
Configuring Find Similar on Initiate Item Screen	355
Prerequisites	355
Find Similar getSimilarObjects on Add Reference Action	362
Using Find Similar on Add Reference Action	362
Configuring Find Similar on Add Reference Action	365
Find Similar - entities/find-similar	370

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

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

Choose a Match Action

When setting up a matching solution, users must choose a match action (as part of a matching algorithm) to determine what the system does with duplicates. The match action defines the workflow and the data model around the objects you are matching.

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, see 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, see 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, see **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, see the **Match Tuning** topic.

Match Criteria

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.

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 entity. 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.

Match Criteria Tab

The Match Criteria tab defines how to compare two objects and evaluate to what degree they are similar.

Important: Create new match algorithms with embedded match codes as defined in the **Configuring Matching Algorithms** topic.

The Match Criteria Tab is separated into the following flippers:

- **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.
- **Matchers** do the actual comparisons of values. A matcher compares one logical aspect of the objects, assigning a equality percentage to that aspect based on the related values.
- **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.
- **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. For information on choosing match codes, see the **Selecting Match Codes** topic.
- **Match Code Filter** allows users to remove specific match code values based on a Transformation Lookup Table.

- **Evaluator** is the user's test tool and allows a user to execute the entire match criteria setup on two select records.

The screenshot displays the 'Individual Matching Tmp - Match Criteria' configuration window. It includes several sections:

- Data Elements:** Lists elements like normName, normAddress, normEmail, and normPhone with their respective normalizers.
- Matchers:** Lists matchers for name, address, email, and phone.
- Rules:** Shows a 'Rules Strategy' set to 'Max' with three rules based on address, email, name, and phone weights.
- Match Code Generators:** Lists generators for emailMatchCode, phoneMatchCode, and nameAndAddress.
- Match Code Filter:** A section for filtering match codes.
- Evaluator:** The active section showing a comparison between two nodes: 'Aaron Kirk (558762)' and 'Aaron Kirk (558774)'. It displays a table of matchers and their scores, and a list of common match codes.

Rules	Score	Matched	Match Reason
1	100.0	true	address = true (100.0), email = true (100.0), name = true (100.0), phone = true (100.0)
2	100.0	true	address = true (100.0), email = true (100.0), name = true (100.0), phone = true (100.0)
3	100.0	true	address = true (100.0), email = true (100.0), name = true (100.0), phone = true (100.0)
Final Score: 100.0			

Legacy Match Criteria Without Embedded Match Codes

For match algorithms without embedded match codes, see the **Match Codes** topic.

A matching algorithm with the Match Criteria flipper displayed indicates it is a legacy algorithm where the match codes are not embedded and must be created manually.

Use the following steps to configure:

1. Click the **Add Criterion** link to display the 'Select Match Criterion' dialog.
2. Specify a **Name**.
3. Choose a match criterion from the **Select Match Criterion** dropdown.
4. Click the **Add** button.
5. Click into the **Criterion** field and then click the ellipsis button (...) to open the editor.
6. Create the matching criterion and click **OK**.
7. Click into the **Weight** field and specify a weight for the criterion.

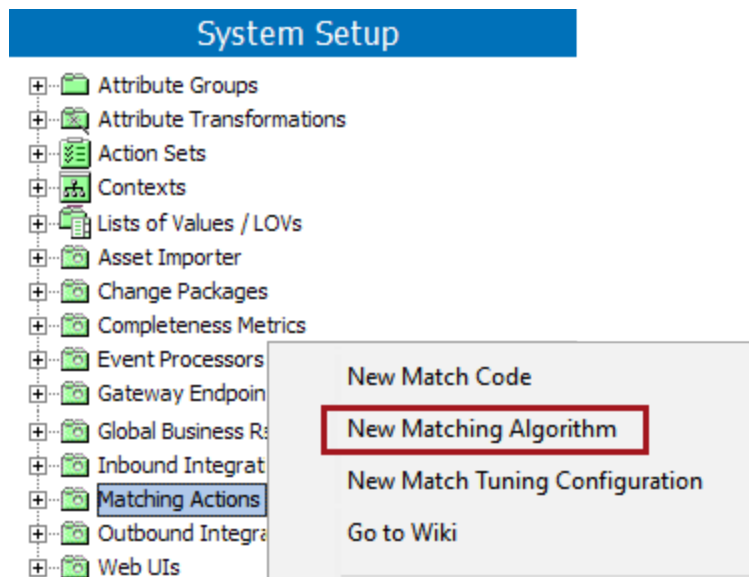
Match Criteria		
Name	Criterion	Weight
DT	Decision Table: Sub Tables 0, Expressions 17, Rules 4	10.0
Add Criterion		

Match code generators are only available when match codes are not embedded into the matching algorithm.

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.

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.

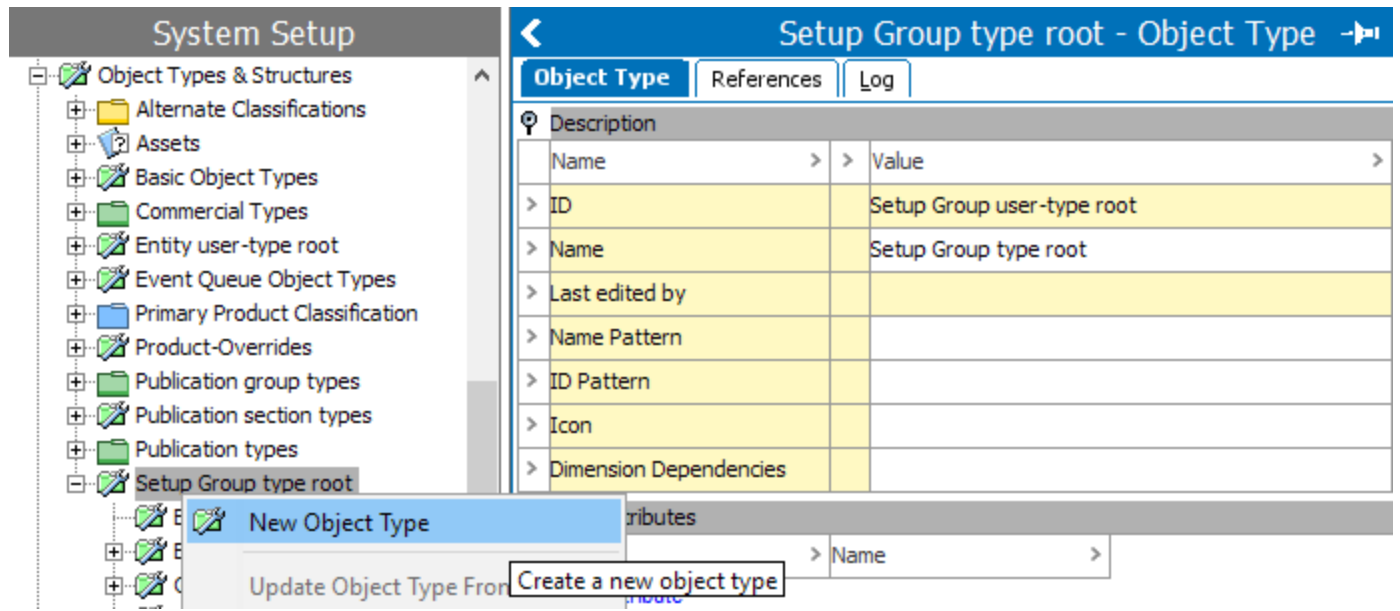
1. Create setup group type for matching algorithms.
2. Link matching algorithm object types to setup group type.
3. Create a matching algorithm setup group.

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

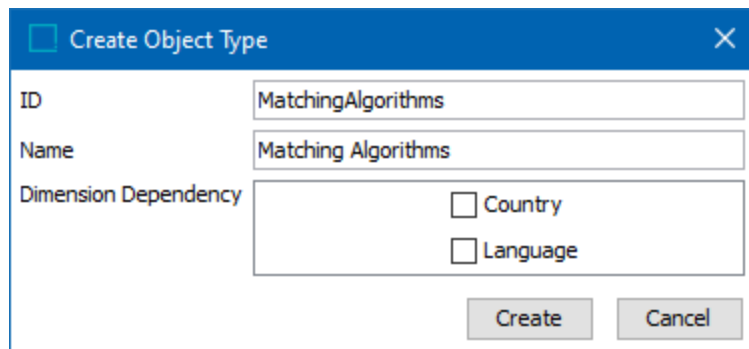
Create Setup Group Type for Matching Algorithms

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.

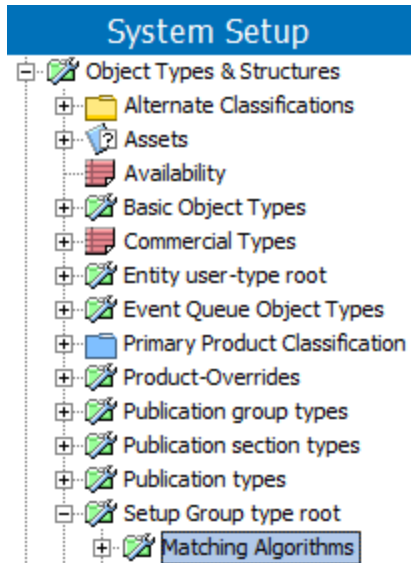


3. Click **New Object Type**, and the Create Object Type dialog will display.

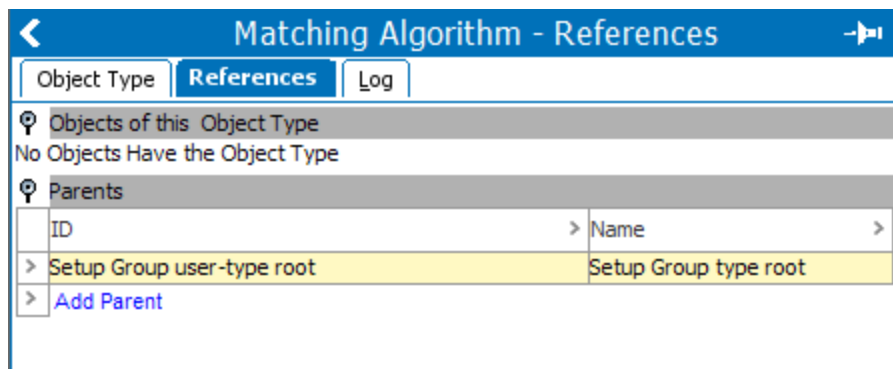


4. Enter an **ID**.
5. Enter a **Name**.
6. Click **Create**.

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



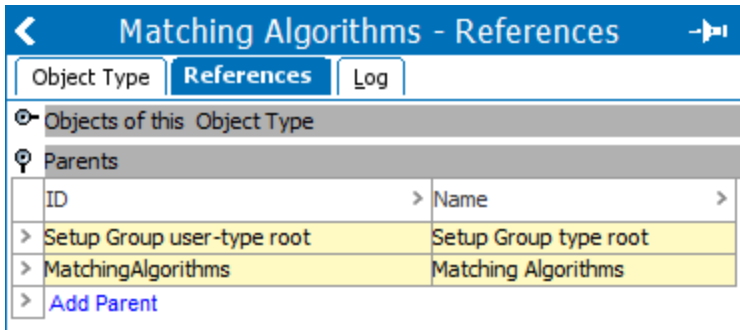
7. Select the newly added Setup Group type > References tab > open the Parents flipper.



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.

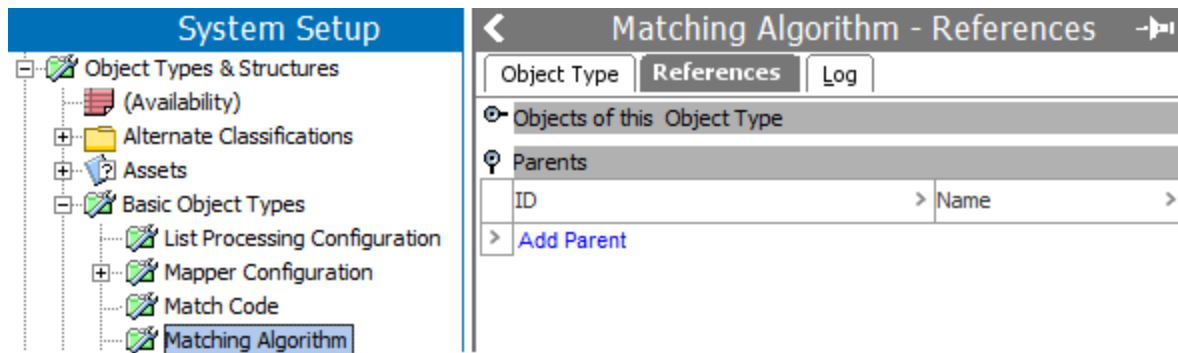
The dialog will close, and the newly created setup group type (i.e., Matching Algorithms) is listed as a parent along with the Setup group user-type root.



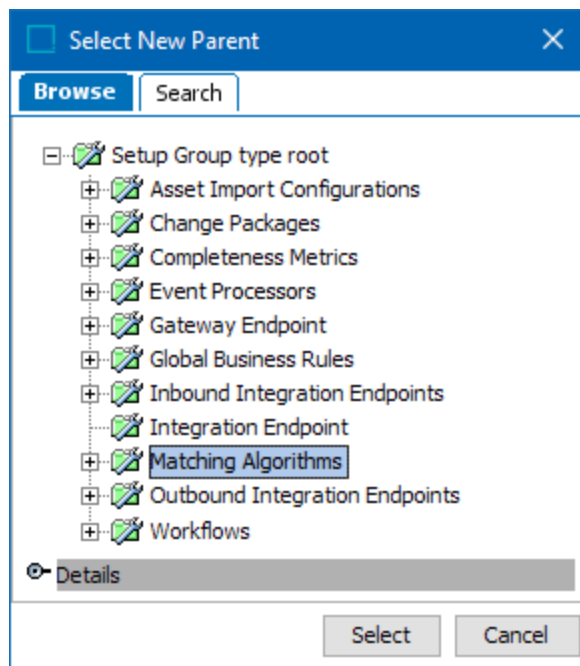
Link Matching Algorithms Object Types to Setup Group Types

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.



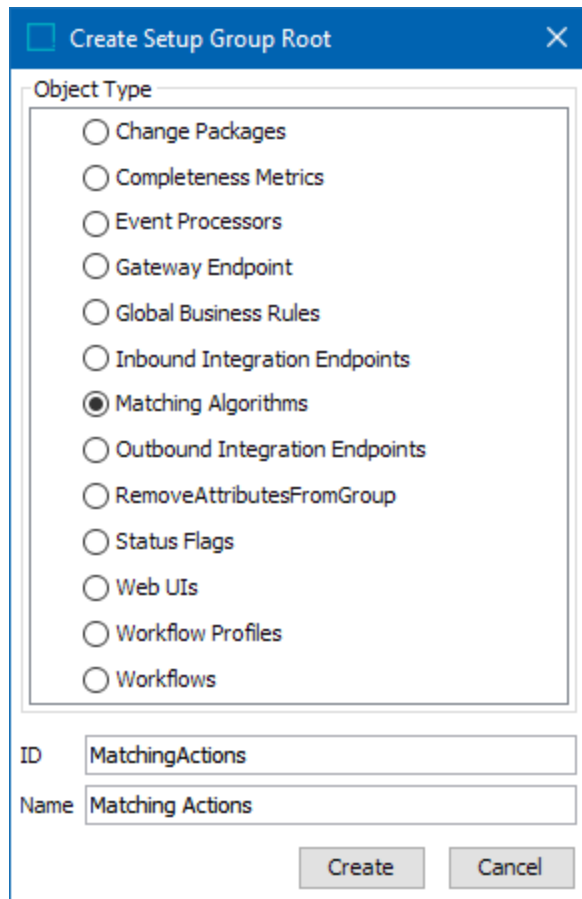
3. Click the **References** tab.
4. Open the **Parents** flipper.
5. Click the **Add Parent** link, and the Select New Parent dialog displays.
6. Browse or search to select **the relevant setup group type**.
7. Click the **Select** button.



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

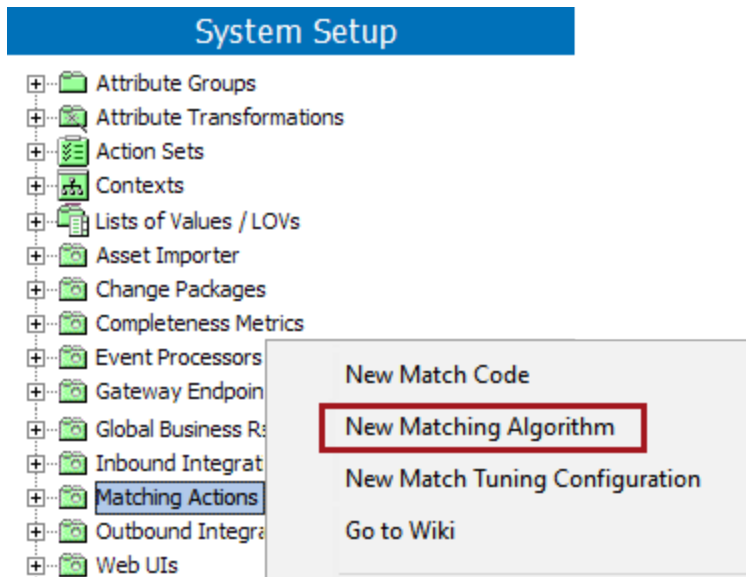
- Change Packages
- Completeness Metrics
- Event Processors
- Gateway Endpoint
- Global Business Rules
- Inbound Integration Endpoints
- Matching Algorithms
- Outbound Integration Endpoints
- RemoveAttributesFromGroup
- Status Flags
- Web UIs
- Workflow Profiles
- Workflows

ID:

Name:

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

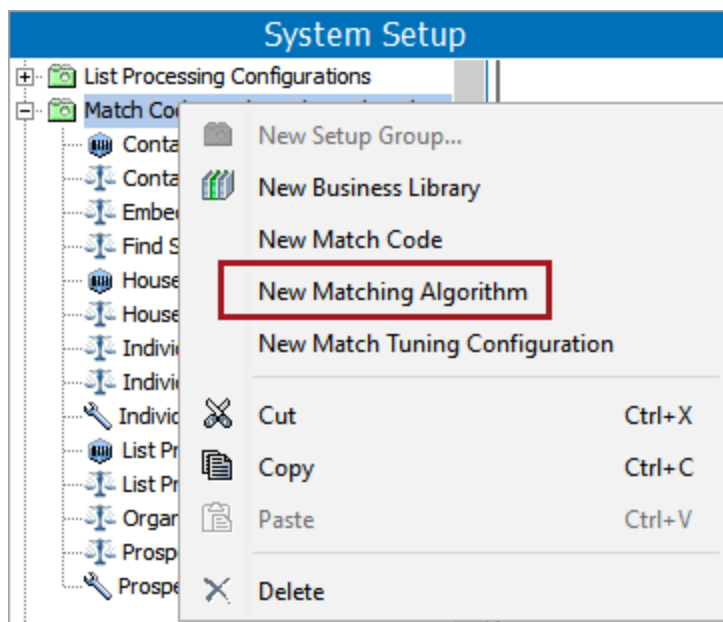
Matching algorithms are used in **Match and Merge** and **Match and Link** solutions.

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

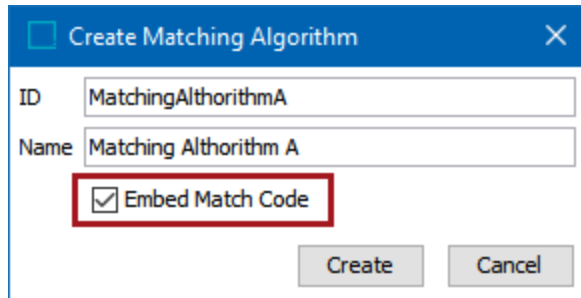
Configuration

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 algorithms.
 - 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, see the **Match Codes** topic.



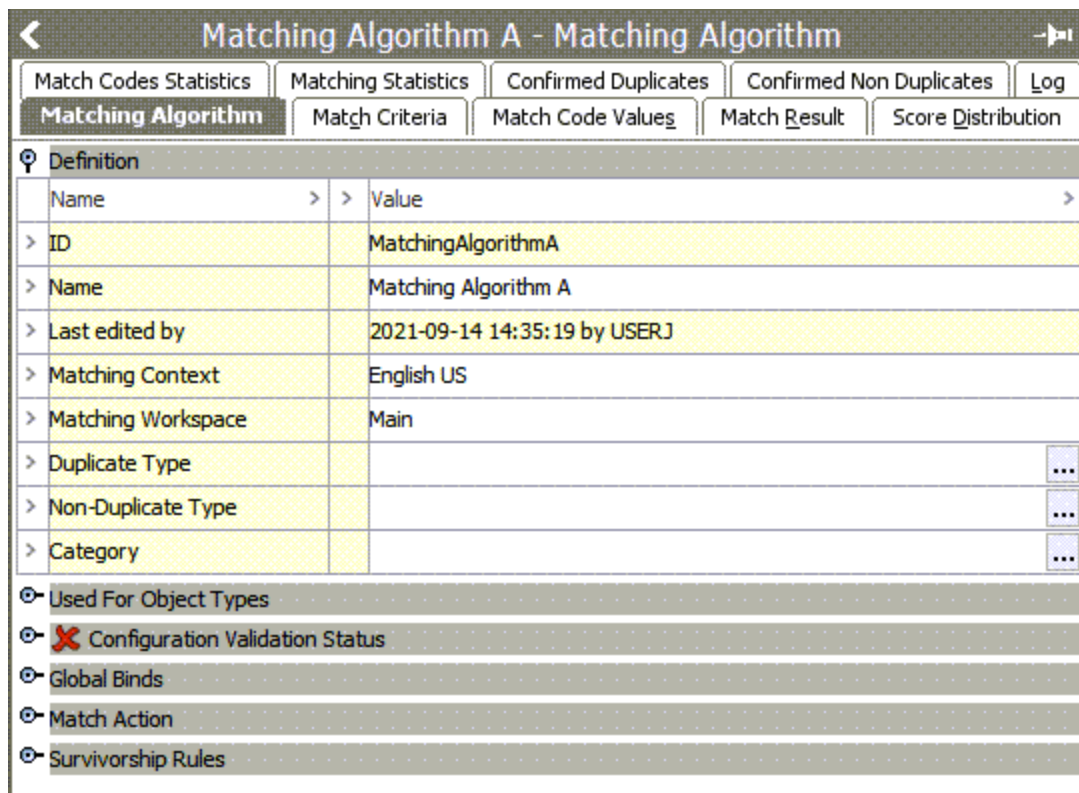
Create Matching Algorithm

ID: MatchingAlthorithmA
 Name: Matching Althorithm A
 Embed Match Code

Create Cancel

- 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. See the last step in this topic for more information.



Matching Algorithm A - Matching Algorithm

Match Codes Statistics Matching Statistics Confirmed Duplicates Confirmed Non Duplicates Log

Matching Algorithm Match Criteria Match Code Values Match Result Score Distribution

Definition

Name	Value
ID	MatchingAlthorithmA
Name	Matching Althorithm 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

3. In the Definition flipper, for the **Matching Context** parameter, specify the context to run the matching algorithm. By default, the current context is set.
4. For the **Matching Workspace** parameter, specify the workspace to run the matching algorithm. By default, the Main workspace is selected.
5. 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, see the **Configuring Matching Component Model** topic.

Note: The Confirmed Duplicate reference type must be multivalued, inheritance None, 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, see the **Configuring Matching Component Model** topic.

Note: The Confirmed Non-Duplicate reference type must be multivalued, inheritance None, 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.

- In the Global Binds flipper, potentially improve the performance by creating global binds to obtain all attribute values used in the decision table comparison.

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 before the match criteria logic is applied and can be referenced from both JavaScript and STEP functions.

Important: Global binds are not optimized for use with In-Memory.

- Click the **Edit Global Binds** link to open the 'Edit Binds' dialog shown below.
- 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 **OK** to close the dialog and return to the Matching Algorithm object.

☐ Edit Binds
✕

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) ... ✕
+ Add Bind		

OK Cancel

8. Open the **Evaluator** flipper, select two objects to test the selected criteria on a data set.

☐ Evaluator
Maxine Hadley (558987) Evaluate

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@CONSECTING.CA	EMAIL #AENEAN.EGESTAS.HENDRERIT@CONSECTING.CA
	EMAIL #MBRUNALT@GUSTR.COM	EMAIL #MERCERBRUNALT@GUSTR.COM
phoneMatchCode	PHONE #12008221111	PHONE #12008223322
	PHONE #12146827444	PHONE #12146827443
nameAndAddress	INDIVIDUAL #H+MKS+410141315	INDIVIDUAL #H+MKS+N+846515637
	INDIVIDUAL #H+MKS+KFNKTN	INDIVIDUAL #H+MKS+N+PSN
	INDIVIDUAL #M+HTL+410141315	INDIVIDUAL #M+HTL+846515637
	INDIVIDUAL #M+HTL+KFNKTN	INDIVIDUAL #M+HTL+N+PSN

9. Set up the match action as needed. For more information, see the **Match Actions** topic.

© Stibo Systems - Confidential - Rel. 10.3-MP4 (March 29, 2022)

24

- Set up the survivorship rules as needed. For more information, see the **Configuring Survivorship Rules** topic.

- 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

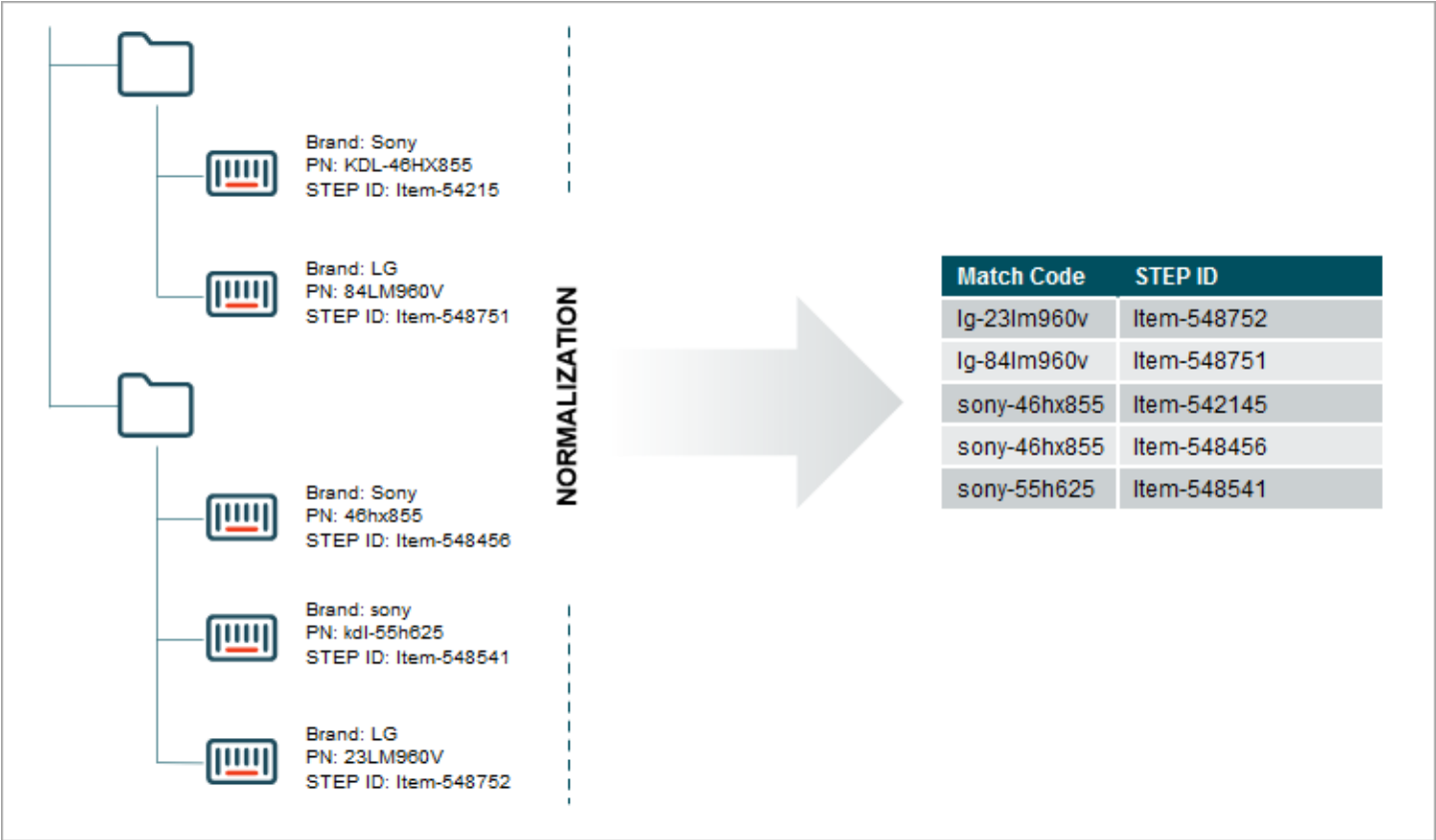
- A green checkmark indicates the matching algorithm has a valid configuration.

🔍	✔ Configuration Validation Status
---	-----------------------------------

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 is compared with the match score. 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.



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.

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.

Matching Algorithm		Match Criteria		Match Code Values		Match Result		Score Di	
☰ 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 a lot of 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, see the **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:

- Configuring a Match Code Generator on the Match Criteria tab
- Configuring a 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.

Evaluator

Select Nodes: Maxie Hadley (558990) ... Maxine Hadley (558987) ... Evaluate

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			

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

Common Match Codes: No

Configuring a Match Code Generator

Match codes are created by match codes generators. There are a number of built-in 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.

To create and configure a match code generator:

1. Open the Matching Algorithm, click the Match Criteria tab, and click the **Edit Match Criteria** link.

The screenshot displays the 'Individual Customer Matching Algorithm - Match Criteria' configuration window. The window is divided into several sections:

- Match Codes Statistics:** Includes tabs for Matching Statistics, Confirmed Duplicates, Confirmed Non Duplicates, and Log.
- Match Criteria:** The active tab, showing various configuration options.
- Data Elements:** A table listing data elements and their associated normalizers.

ID	Data Elements	Comment
> normEmail	Email Normalizer(DC:Emails)	
> normName	Organization Name Normalizer(On Obj...	
> normAddress	Address Normalizer (On Object, DC:Ad...	
> dunsNorm	Business Function Normalizer: List<Str...	
- Matchers:** A table listing matchers and their associated data elements.

ID	Matcher	Comment
> EmailMatcher	Email Matcher (normEmail)	
> NameMatcher	Organization Name Matcher (normName)	
> AddressMatcher	Address Matcher (normAddress)	
- Rules:** A table listing rules and their associated matchers.

#	AddressMatc...	EmailMatcher...	NameMatche...	Result	Comment
> 1				(EmailMatcher*85) / 100	
> 2				(NameMatcher*50.0 + Add...	
- Match Code Generators:** A table listing match code generators and their associated matchers.

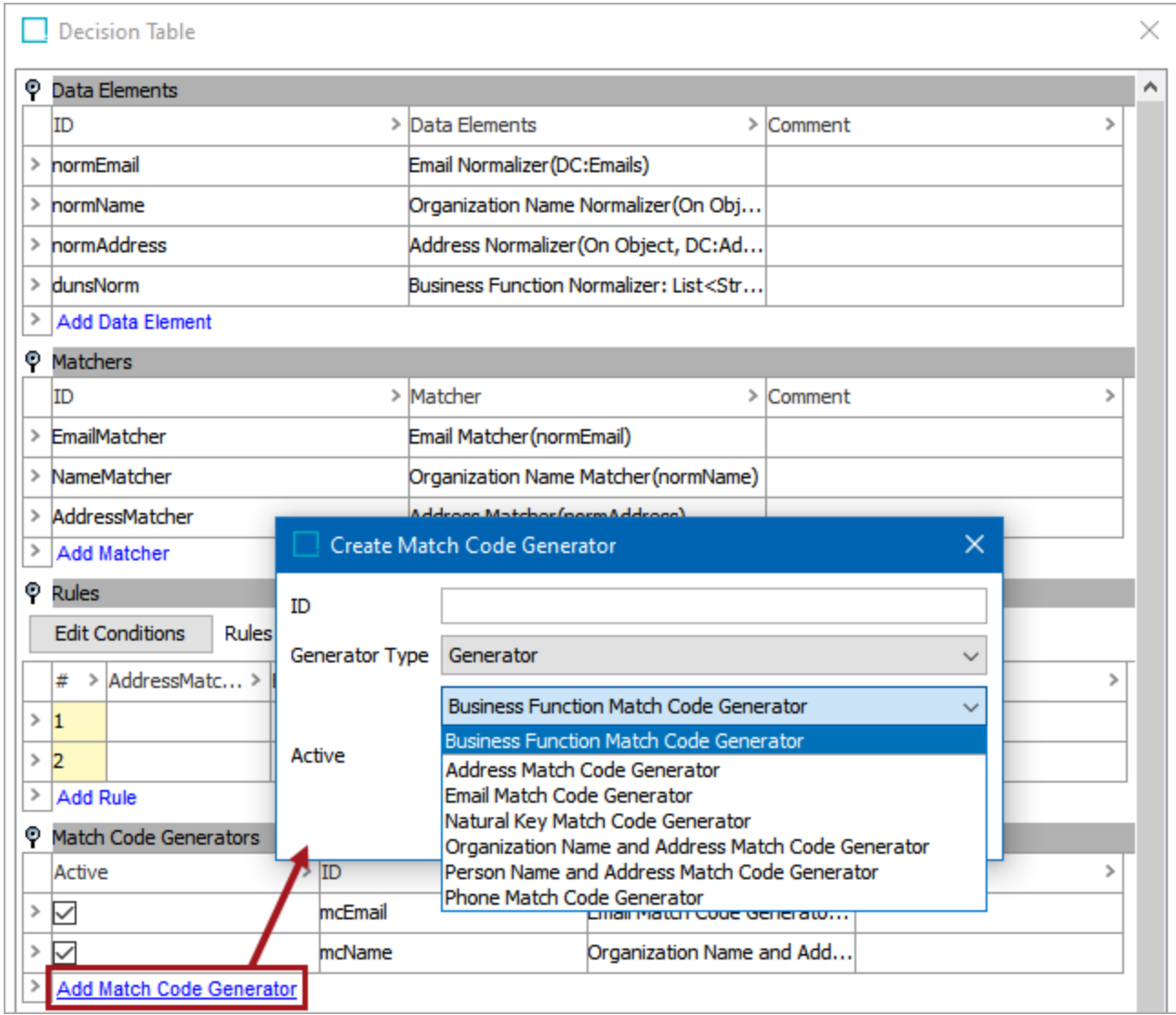
ID	Active	ID	Match Code Generator	Comment
> Prohib	<input checked="" type="checkbox"/>	mcEmail	Email Match Code Generato...	
> Evalu	<input checked="" type="checkbox"/>	mcName	Organization Name and Add...	
- Match Code Filter:** A table listing match code filters and their associated matchers.

ID	Match Code Filter	Comment
> ProhibitedStates	Table Match Code Filter: null, null	
- Evaluator:** A section for selecting nodes, currently showing 'ACE HARDWARE CORP-01139B (D&B222904)'.

Select Nodes: ACE HARDWARE CORP-01139B (D&B222904)

A red arrow points to the **Edit Match Criteria** link in the bottom left corner of the window.

2. In the Match Code Generator flipper, click the **Add Match Code Generator** link.



3. For the **ID** parameter, add an ID.
4. For the **Generator Type** parameter, leave the default and only option selected and from the second dropdown choose the type of generator. Each is defined in the following sections.
5. For the **Active** parameter, when checked, the generator is used by the algorithm.
6. Click the **Add Match Code Generator** button to display the new generator on the Decision Table dialog.
7. In the Match Code Generators flipper, click the Match Code Generator added, click the ellipsis button (...) to display a configuration dialog and provide the required settings. and click the **OK** button. Configuration for each generator is defined in the following sections.
8. Prepare to test the match code generator by creating some codes. The following example shows several matched records on the previously detailed match code generator. In this example, all of these records share

the same email. For more information, see the **Configuring Matching Algorithms** topic.

9. Test the match code generators with a variety of records and edit accordingly to achieve the proper results.

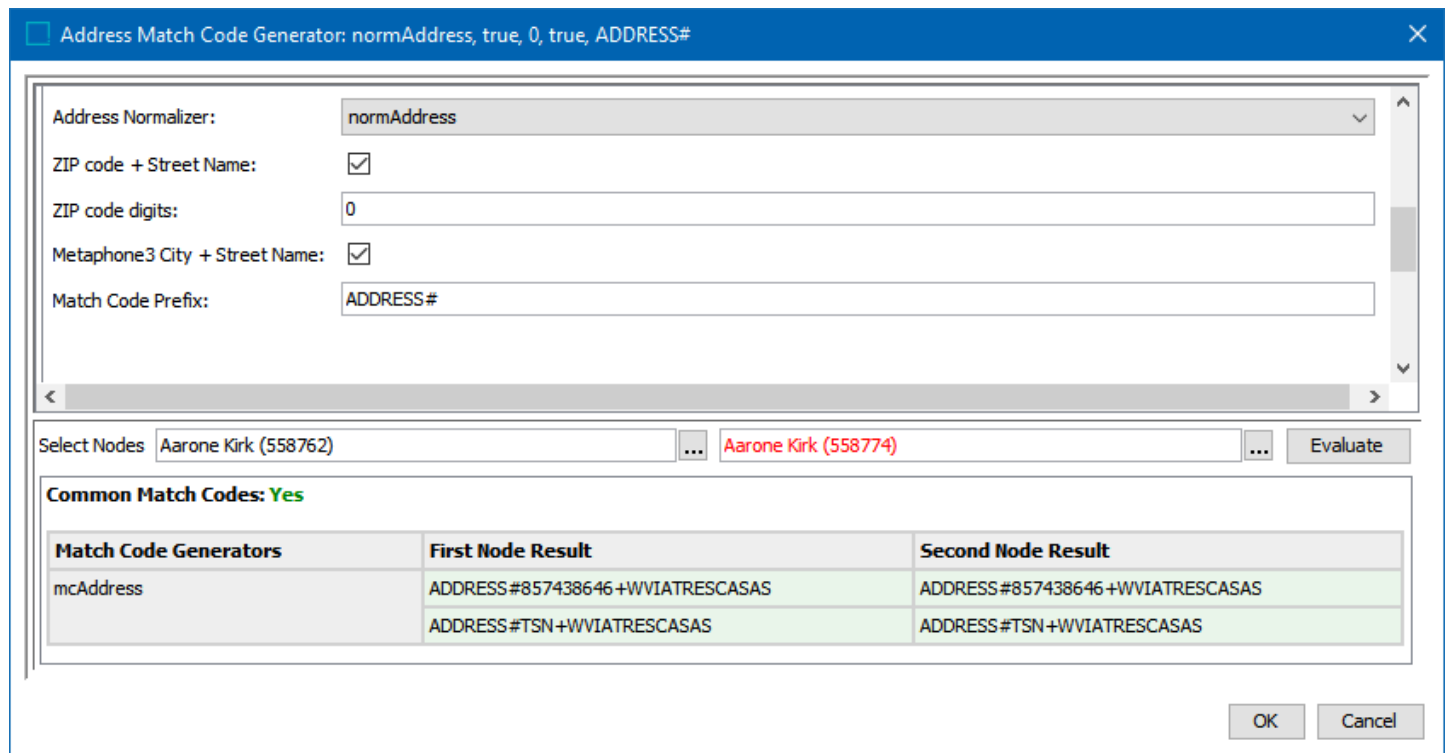
All of the match code generators described below require that users map one or more normalizers to generate codes. For more information, see the **Match Criteria Data Elements** topic.

Business Function Match Code Generator

A Business Function match code generator uses a business function to produce match codes.

Address Match Code Generator

The Address match code generator must be mapped to an address normalizer.



Address Normalizer: normAddress

ZIP code + Street Name:

ZIP code digits: 0

Metaphone3 City + Street Name:

Match Code Prefix: ADDRESS#

Select Nodes: Aarone Kirk (558762) ... Aarone Kirk (558774) ... Evaluate

Common Match Codes: Yes

Match Code Generators	First Node Result	Second Node Result
mcAddress	ADDRESS#857438646+WVIATRESCASAS	ADDRESS#857438646+WVIATRESCASAS
	ADDRESS#TSN+WVIATRESCASAS	ADDRESS#TSN+WVIATRESCASAS

OK Cancel

Email Match Code Generator

The Email match code generator must be mapped to an email normalizer.

Email Match Code Generator: normEmail, EMAIL#

Email Normalizer: normEmail

Match Code Prefix: EMAIL#

Select Nodes: Aarone Kirk (558762) ... Aarone Kirk (558774) ... Evaluate

Common Match Codes: Yes

Match Code Generators	First Node Result	Second Node Result
emailMatchCode	EMAIL #FRINGILLA.EST@EU.ORG	EMAIL #FRINGILLA.EST@EU.ORG
	EMAIL #ROSETTAFALLACI@ARMYSPY.COM	EMAIL #ROSETTAFALLACI@ARMYSPY.COM

OK Cancel

Natural Key Match Code Generator

The Natural Key code generator must be mapped to a word normalizer.

Natural Key Match Code Generator: normName, , false, KEY#

Word Normalizer: normName

Match Code Split Regex:

Apply Metaphone3:

Match Code Prefix: KEY#

Select Nodes: Monster Buy HQ (551702) ... Monster Buy HQ (551720) ... Evaluate

Common Match Codes: Yes

Match Code Generators	First Node Result	Second Node Result
NameMC	KEY #MONSTER BUY HQ	KEY #MONSTER BUY HQ

OK Cancel

Organization Name and Address Match Code Generator

It is often useful to match on organization names, but on their own, they are often not unique. For example, a supermarket chains can use the same organization name for each of their sites, so to achieve a unique 'name' to identify a site, add another piece of data. The combination of organization name and address usually constitutes a good match code.

The organization name and address match code generator must be mapped to an organization name normalizer and an address normalizer.

Organization names are often abbreviated and appended with terms like 'Inc.' that can be omitted in some systems. For that reason, the organization name should often be followed by a transformation lookup table with aliases.

Organization Name and Address Match Code Generator: legalNameNormalizer, null, normAddress, ORGANIZATION#, true, 0, true, false, 0, false
✕

Organization Name Normalizer:	<input type="text" value="legalNameNormalizer"/>
Organization Name Aliases:	<input type="text" value=""/> ...
Address Normalizer:	<input type="text" value="normAddress"/>
Match Code Prefix:	<input type="text" value="ORGANIZATION#"/>
Metaphone3 Organization Name Token + ZIP code:	<input checked="" type="checkbox"/>
ZIP code digits:	<input type="text" value="0"/>
Metaphone3 Organization Name Token + Metaphone3 City:	<input checked="" type="checkbox"/>
Metaphone3 Organization Name Token + ZIP code + Street name:	<input type="checkbox"/>
ZIP code digits:	<input type="text" value="0"/>
Metaphone3 Organization Name Token + Metaphone3 City + Street name:	<input type="checkbox"/>

Select Nodes ... Monster Buy HQ (551720) ... Evaluate

Common Match Codes: No

Match Code Generators	First Node Result	Second Node Result
mcNameAndAddress	ORGANIZATION#MNSTRPK+8220	ORGANIZATION#MNSTRPK+8270
	ORGANIZATION#MNSTRPK+PRPRINT	ORGANIZATION#MNSTRPK+HJPJRK

OK Cancel

Person Name and Address Match Code Generator

It is often useful to match on person names, but alone they are often not unique. The combination of person name and address usually constitutes a good match code.

The person name and address match code generator must be mapped to a person name normalizer and an address normalizer.

Person Name and Address Match Code Generator: normName, null, normAddress, 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: Aarone Kirk (558762) ... Aarone Kirk (558774) ... Evaluate

Common Match Codes: Yes

Match Code Generators	First Node Result	Second Node Result
nameAndAddress	INDIVIDUAL #A +KRK +857438646	INDIVIDUAL #A +KRK +857438646
	INDIVIDUAL #A +KRK +TSN	INDIVIDUAL #A +KRK +TSN
	INDIVIDUAL #K +ARN +857438646	INDIVIDUAL #K +ARN +857438646
	INDIVIDUAL #K +ARN +TSN	INDIVIDUAL #K +ARN +TSN

OK Cancel

Person names are sometimes abbreviated or exchanged for call names. For that reason, the person names should often be followed by a transformation lookup table with aliases.

Lookup Table	
<input type="checkbox"/>	Replace with default value when no matches are found (Value Substitution only):
<input type="checkbox"/>	Replace with a source value when no matches are found and default value is empty (Value Substitution only)
<input type="checkbox"/>	Ignore Case
From	To
> aaron	ron
> abbie	abbey
> abby	abbey
> abe	ab
> abel	abe
> abig	abbey
> abigail	abbey
> abr	ab
> abra	abraham
> abraham	ab

Phone Match Code Generator

The phone match code generator must be mapped to a phone normalizer.

Phone Match Code Generator: normPhone, PHONE# ✕

Phone Normalizer:

Match Code Prefix:

Select Nodes

Common Match Codes: Yes

Match Code Generators	First Node Result	Second Node Result
phoneMatchCode	PHONE#12263280522	PHONE#12263280522
	PHONE#13104011771	PHONE#13104011771

Configuring a Match Code Filter

Sometimes data exceptions 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.

To create a match code filter:

1. Create a new transformation lookup table and enter all the match codes to exclude in the 'from' column. Leave the 'to' column empty.

The screenshot shows the configuration of a 'Match Code Anon Values rev.1.0 - Transformation Lookup Table'. The left sidebar shows a tree view with 'Match Code Anon Values' selected under 'Matching Lookup Tables'. The main panel displays the following details:

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	URL
Keywords	abc...
OriginalRecord	

Below the details is the 'Lookup Table' configuration section:

- 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	
>	

At the bottom, there is an 'Add Row' button, a '5 Rows' indicator, and 'Import From Clipboard' and 'Apply' buttons.

- In System Setup, open the matching algorithm for the match code filter.
- On the Match Criteria tab click the **Edit Match Criteria** link to open the Decision Table dialog. In the Match Code Filter flipper, click **Add Match Code Filter** link.

Match Code Generators			
Active	ID	Match Code Generator	Comment
<input checked="" type="checkbox"/>	emailMatchCode	Email Match Code Generator: normEmail, EMAIL #	
<input checked="" type="checkbox"/>	phoneMatchCode	Phone Match Code Generator: normPhone, PHONE#	
<input checked="" type="checkbox"/>	nameAndAddress	Person Name and Address Match Code Generator: no...	
Add Match Code Generator			
Match Code Filter			
ID	Match Code Filter	Comment	
Add Match Code Filter			

- On the Create a Match Code Filter dialog, add an **ID**, leave the dropdowns menus, and click the **Add Match Code Filter** link.

❑ Create Match Code Filter
✕

ID

Match Code Filter Type Match Code Filter ▾

Table Match Code Filter ▾

Add Match Code Filter
Cancel

- On the Match Code Filter column, on the table match code filter row click the ellipsis button (...)

Match Code Filter		
ID	Match Code Filter	Comment
Email	Table Match Code Filter: null, null	...
Add Match Code Filter		

- On the Table Match Code Filter dialog, 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. For example, 'EMAIL#' shown below. Leave this parameter can and should be left blank if all the match codes in the transformation lookup table already have this prefix.

- For the **Select Nodes** parameter, choose two records to test if the match codes can be found on the selected nodes. Click the **Evaluate** button allows users. In this example, two organization nodes are selected, the user clicks the Evaluate button, and then the filtered results are shown.

Table Match Code Filter: null, null

Filter Table: Match Code Anon Values (MatchCodeAnonValues) ...

Match Code Prefix: EMAIL#

Select Nodes: Monster Buy HQ (551702) ... Monster Buy HQ (551702) ... Evaluate

Normalizers	First Node Result	Second Node Result
Email Filter	EMAIL#Co, EMAIL#Inc, EMAIL#Ltd, EMAIL#Limited	EMAIL#Co, EMAIL#Inc, EMAIL#Ltd, EMAIL#Limited

OK Cancel

- Click the **OK** button.
- On the Decision Table dialog, in the Evaluator flipper, select two nodes and click the **Evaluate** button to test specific nodes.
- When the filtering tests are satisfactory, click the **Save** button.

Selecting 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, see the **Data Profiling** documentation.

Review the example below to see how the data profile 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 Group

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]

Overview **Frequent Values** Rare Values Frequent Patterns Rare Patterns

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.

Overview Frequent Values Rare Values Frequent Patterns Rare Patterns

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	A99999999				
> 3	AAA9999				
> 2	99999				
> 2	999999				
> 2	9A9999				
> 2	A9999				
> 2	AAA-9999				
> 1	99A9999				
> 1	A9-9999				
> 1	A999 9999				
> 1	A9999999				

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.

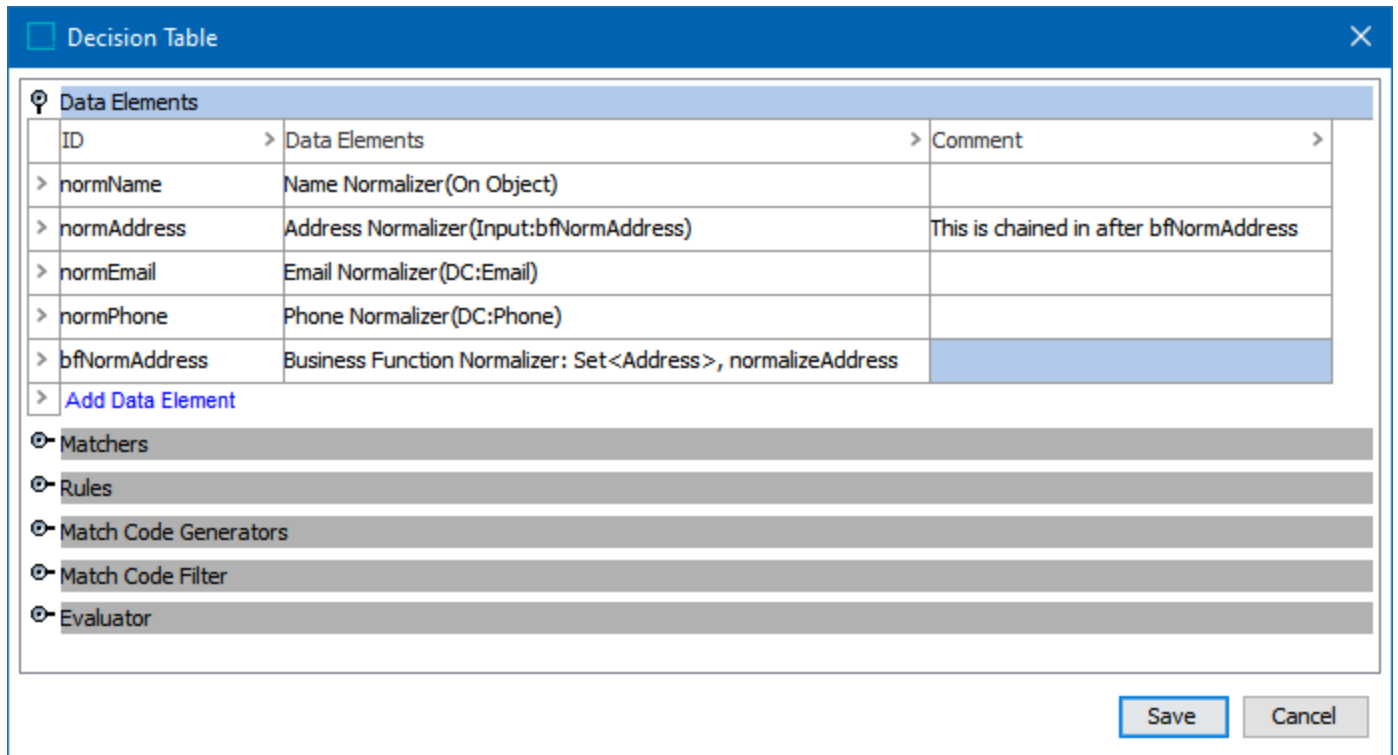
Matcher Strategy Options

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 Data Elements

The Data Elements section of a decision table defines input data for the match criteria. The data element is responsible for retrieving data and making it easy to compare. This often involves the reduction of data to a kind of canonical form, like making letters lower case in a text, removing spaces from phone numbers, or expanding abbreviations.



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)	
> bffNormAddress	Business Function Normalizer: Set<Address>, normalizeAddress	
> Add Data Element		

Matchers
Rules
Match Code Generators
Match Code Filter
Evaluator

Save Cancel

Most data elements take data from source objects, then normalize it in some form, before providing the data to matchers and match code generators.

The following data element types are available:

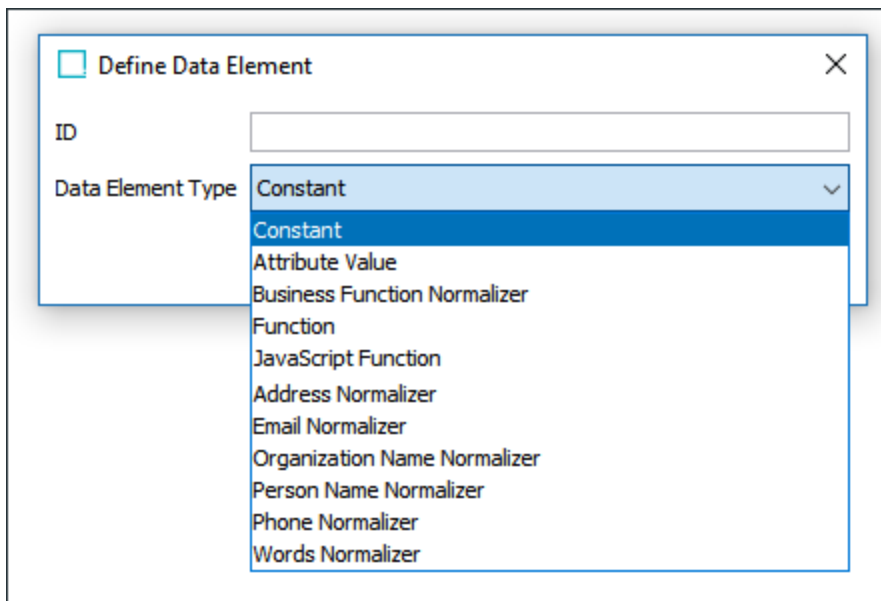
Standard Data Elements

- Constant
- Attribute Value
- Business Function Normalizer
- Function
- JavaScript Function

Party Data Normalizers

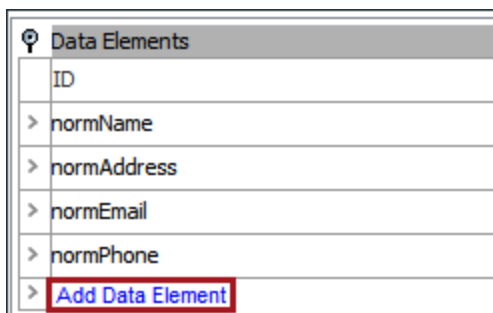
- Address Normalizer
- Email Normalizer
- Organization Name Normalizer
- Person Name Normalizer
- Phone Normalizer
- Words Normalizer

Data elements can be chained so that the output of one data element can be used as input to another data element.

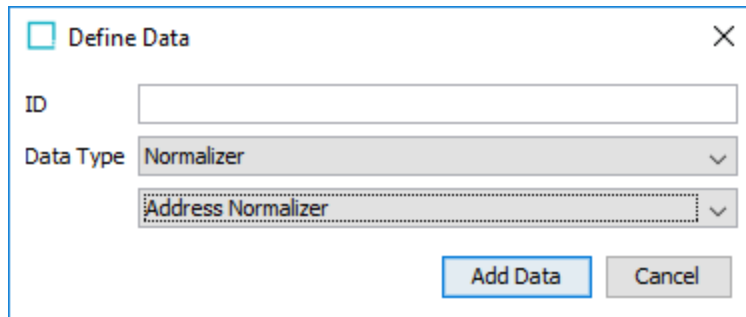


Configuration

Use the following steps to configure data elements for a matching algorithm.



1. To add data to the table, click the 'Add Data' element link.
2. In the 'Define Data Element' popup dialog, enter an ID for the data element and use the Data Type dropdown (s) to define the data type, then click the Add Data button.



Important: No two data elements, matchers, or match code generators should have the same ID.

3. The new data element will be present in the table but is still not configured. Click the ellipsis button (...) to populate the data element. See the section corresponding to your chosen data element under **Standard Data Elements** and **Party Data Normalizers** below for more information.

Standard Data Elements

Standard data elements include:

- Constants
- Attribute Value
- Business Function Normalizer
- STEP Functions
- JavaScript Functions

Attribute Value

The Attribute Value allows users to specify a single attribute and output its value. To specify an attribute, click the ellipsis button (...) and browse or search for the desire attribute. This data element does not normalize its output.

☐ Attribute Value: Phone Number
✕

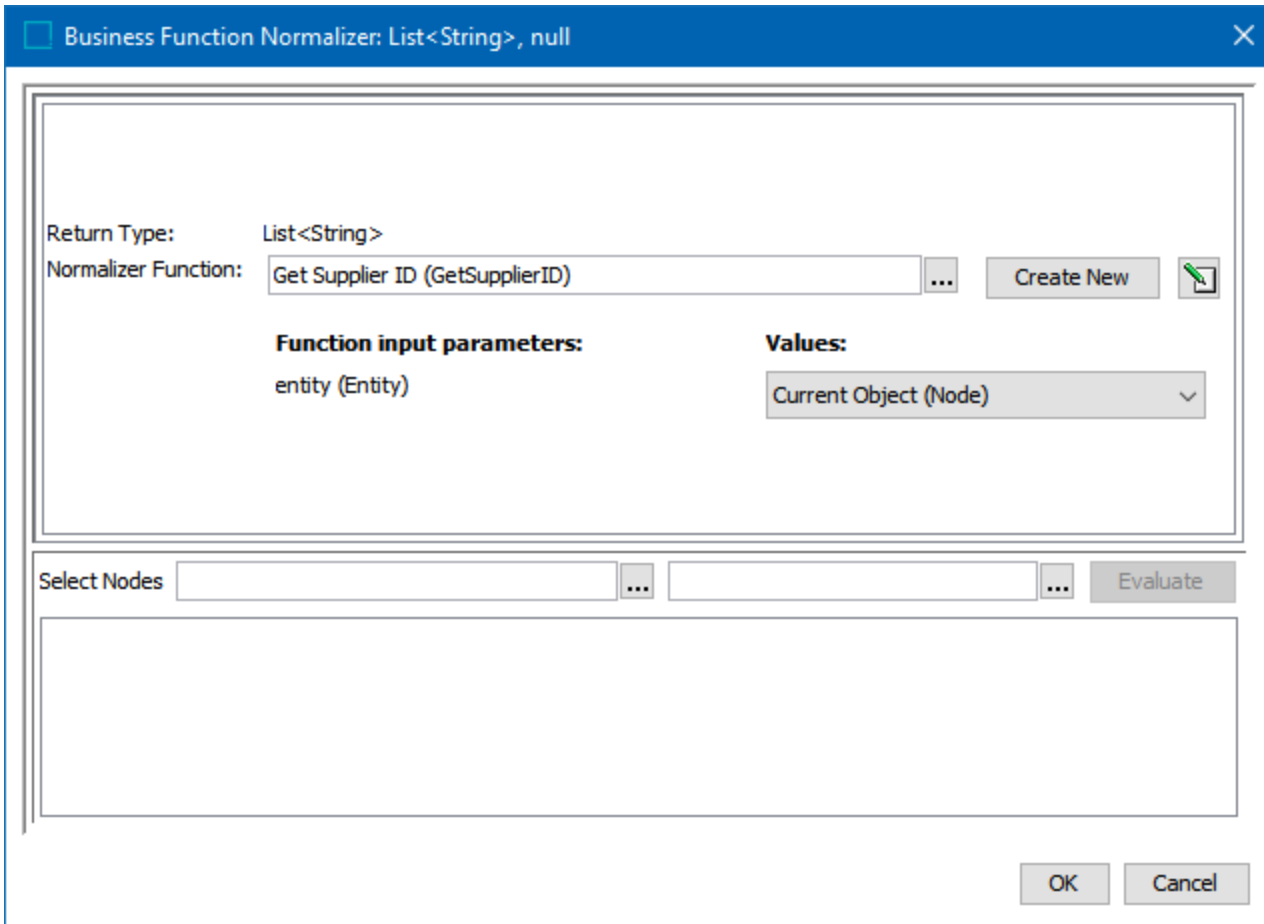
Attribute: ...

Select Nodes
 ...
 ...
 Evaluate

OK
Cancel

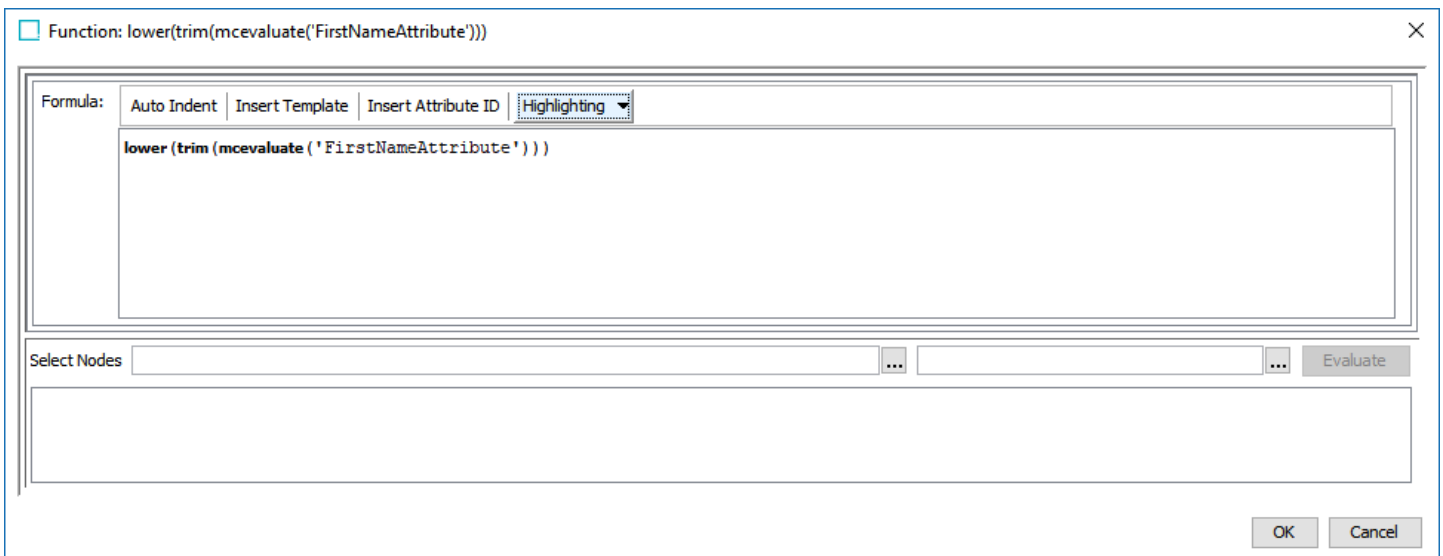
Business Function Normalizer

A Business Function Normalizer is the most versatile normalizer in the toolbox. Using a business function, it can take any number of values from the source records and produce a normalized data element. For more information, see the **Business Functions** topic of the **Business Rules** documentation.



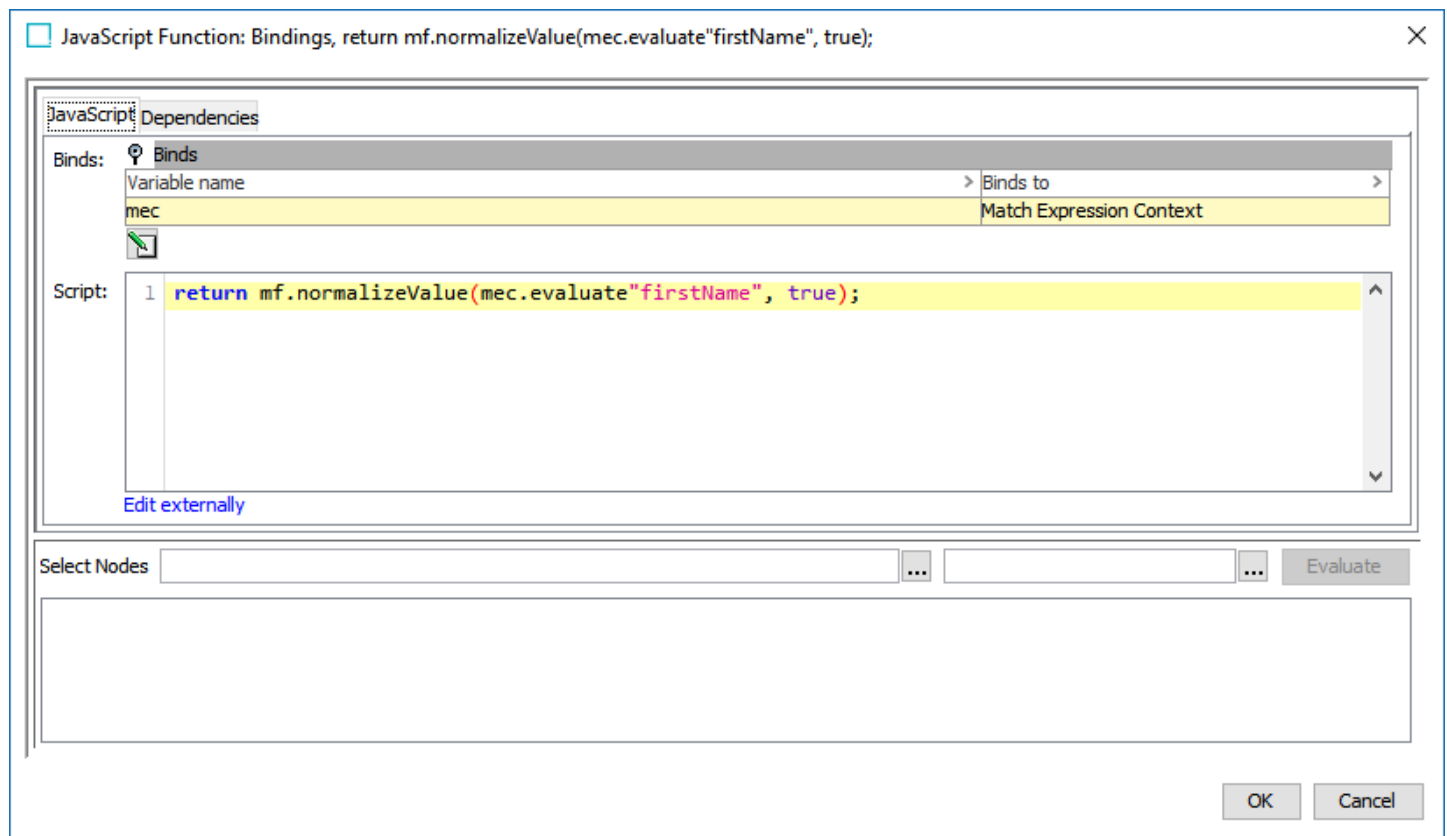
STEP Function

The element is called 'Function' on the dropdown list, and this option will normalize values via built-in STEP functions.



JavaScript Function

Normalized values produced via JavaScript functions.



JavaScript Function: Bindings, return mf.normalizeValue(mec.evaluate("firstName"), true);

JavaScript Dependencies

Variable name	Binds to
mec	Match Expression Context

Script:

```
1 return mf.normalizeValue(mec.evaluate("firstName"), true);
```

Select Nodes: [] [] [] Evaluate

OK Cancel

Party Data Normalizers

These normalizer templates are intended for matching of party data:

- Address Normalizer
- Email Normalizer
- Organization Name Normalizer
- Person Name Normalizer
- Phone Normalizer
- Words Normalizer

Address Normalizer

The Address Normalizer produces a normalized set of addresses for use in the corresponding Address Matcher.

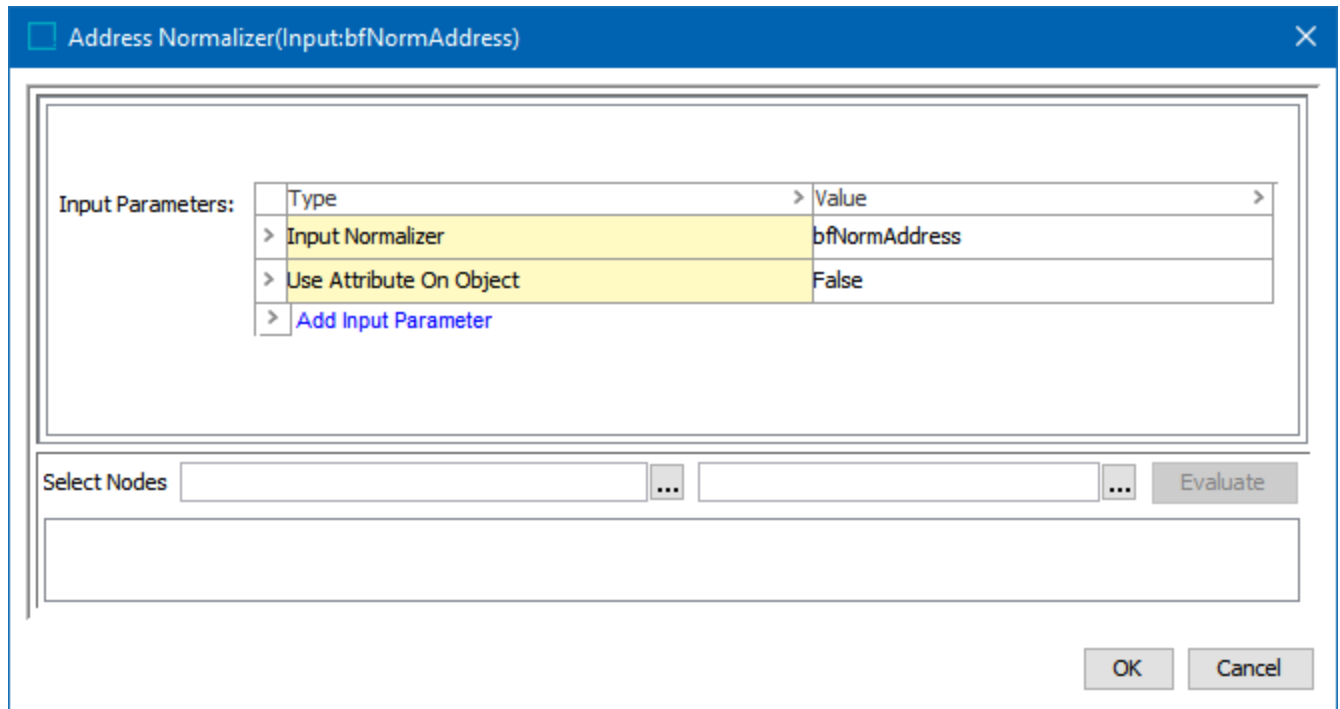
This data is provided by the input address element attributes mapped to the Address component model and include the following: Input City, Input Country, Input State, Input Street, Input Zip, Standardized City, Standardized Country, Standardized Country ISO Code, Standardized State, Standardized Street, Standardized Zip.

The output of the Address Normalizer is `java.util.Set<com.stibo.partydatamatching.domain.address.Address>`

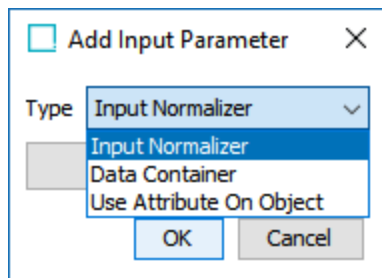
Note: If the postal code mapped to the corresponding standardized component model parameter is valid for the objects being compared, the decision table will output normalized values for address attributes mapped to the standardized attribute fields on the component model.

For more information on the Address Component Model, see the **Address Component Model** section of the **Data Integration** documentation.

1. To normalize customer address data, click the ellipsis button (...) in the Data column to access the configuration.



2. For Input Parameters, click the Add Input Parameter link, and in the dialog, select the input type from the dropdown.



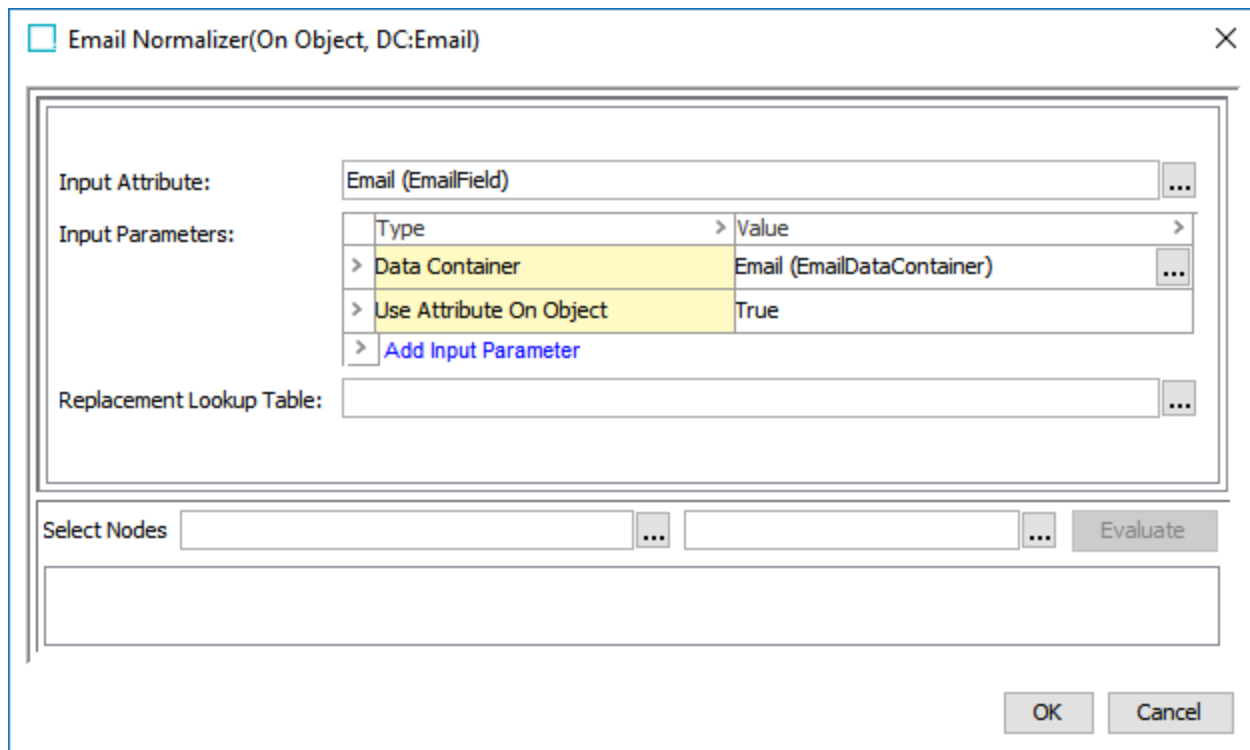
- Input Parameter: Select either a data container or the node object itself, which must hold the input attributes the Address Normalizer use, as defined by the component model. There is also the option of using an Input Normalizer, typically a business function written in JavaScript.
- Data Container: When you select 'Data Container,' click the ellipsis button (...) and browse or search for a data container type. The selected data container type will have its address data normalized when used in a matcher.
- Use Attribute on Object: When you select 'Use Attribute On Object,' click the dropdown and select 'True.' This input parameter will normalize address data that has been mapped to the address component model. This input parameter will be configured by default.

3. Click OK once the selection is made.

Email Normalizer

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

1. To normalize email data, click the ellipsis button (...) in the Data column to access the configuration.



Email Normalizer(On Object, DC:Email)

Input Attribute:

Input Parameters:

Type	Value
> Data Container	Email (EmailDataContainer)
> Use Attribute On Object	True
Add Input Parameter	

Replacement Lookup Table:

Select Nodes

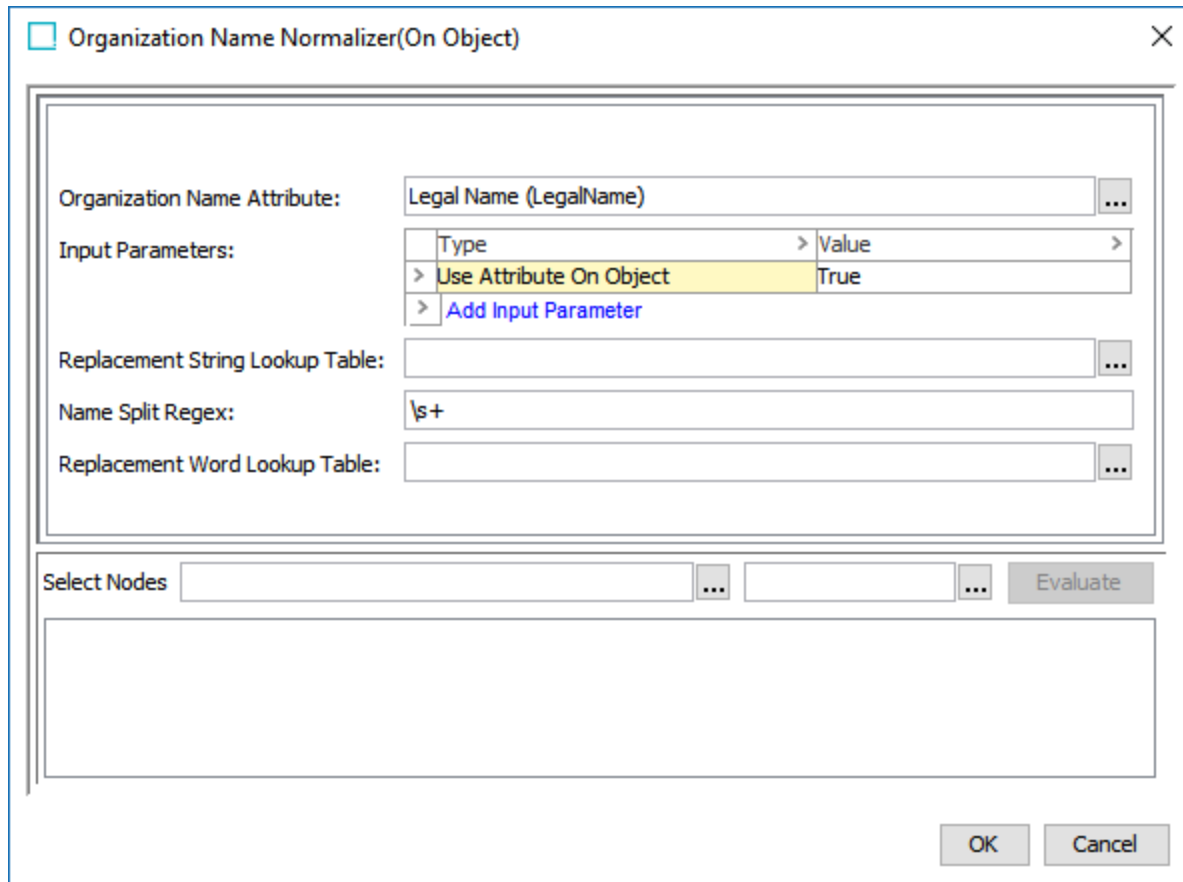
2. For the Input Attribute parameter, click the ellipsis button (...) and browse or search for an email attribute to normalize.
3. For Input Parameters, click the Add Input Parameter link, and in the Add Input Parameter popup dialog, select the input type from the dropdown.
 - Input Normalizer: Enter the ID of another Email Normalizer to use its output. Typically, this normalizer is written in JavaScript.

- Data Container: When you select 'Data Container,' click the ellipsis button (...) and browse or search for a data container. The selected data container will have its email data normalized when used in a matcher.
 - Use Attribute on Object: When you select 'Use Attribute On Object,' click the dropdown and select 'True.' This input parameter will normalize the email attribute mapped to the Input Attribute field. This input parameter will be configured by default.
4. For the Replacement Lookup Table parameter, click the ellipsis button (...) and select a lookup table. Typically, this is used to remove invalid email values.
 5. Click OK when finished.

Organization Name Normalizer

An Organization Name Normalizer can normalize organization name data for use in the corresponding Organization Name Matcher.

1. To normalize organization name data, click the ellipsis button (...) in the Data column to access the configuration.



The screenshot shows the 'Organization Name Normalizer(On Object)' dialog box. It contains the following fields and controls:

- Organization Name Attribute:** A text box containing 'Legal Name (LegalName)' with an ellipsis button (...).
- Input Parameters:** A table with columns 'Type' and 'Value'.

Type	Value
> Use Attribute On Object	True
> Add Input Parameter	
- Replacement String Lookup Table:** A text box with an ellipsis button (...).
- Name Split Regex:** A text box containing '\s+'.
- Replacement Word Lookup Table:** A text box with an ellipsis button (...).
- Select Nodes:** Two empty text boxes with ellipsis buttons (...).
- Evaluate:** A button.
- OK / Cancel:** Buttons at the bottom right.

2. For the Organization Name Attribute parameter, click the ellipsis button (...) and browse or search for an organization name attribute to normalize.

3. For Input Parameters, click the Add Input Parameter link, and in the Add Input Parameter popup dialog, select the input type from the dropdown.
 - Input Normalizer: Enter the ID of another Organization Name Normalizer to use its output. Typically, this normalizer is written in JavaScript.
 - Data Container: When you select 'Data Container,' click the ellipsis button (...) and browse or search for a data container. The selected data container will have its organization name data normalized when used in a matcher.
 - Use Attribute on Object: When you select 'Use Attribute On Object,' click the dropdown and select 'True.' This input parameter will normalize the organization name attribute mapped to the Organization Name Attribute field. This input parameter will be configured by default.
4. For the Replacement String Lookup Table parameter, click the ellipsis button (...) and select a lookup table. This is used to account for inconsistencies in organization names by defining semantically equivalent strings (especially the usage of apostrophes and quotations). For example, normalizing "'s' to be 's' would change the organization name ACME's into ACMEs. Normalizing 'n' to be and would change the organization name ACME'n'SON to ACME and SON.
5. For the Name Split Regex parameter, enter the RegEx used to split the value of the Organization Name Attribute into words.
6. For the Replacement Word Lookup Table parameter, click the ellipsis button (...) and select a lookup table. Typically, this is used to account for the inconsistent use of common words in organization names. For example, '&' can be replaced by 'and.'
7. Click OK when finished.

Person Name Normalizer

A Person Name Normalizer can normalize customer name data for use in the corresponding Person Name Matcher.

1. To normalize customer name data, click the ellipsis button (...) in the Data column to access the configuration.

Name Normalizer(On Object) ✕

First Name Attribute: ...

Middle Name Attribute: ...

Last Name Attribute: ...

Input Parameters:

Type	Value
> Use Attribute On Object	True
> Add Input Parameter	

Name Split Regex:

Replacement Word Lookup Table: ...

Normalize Accents:

Select Nodes

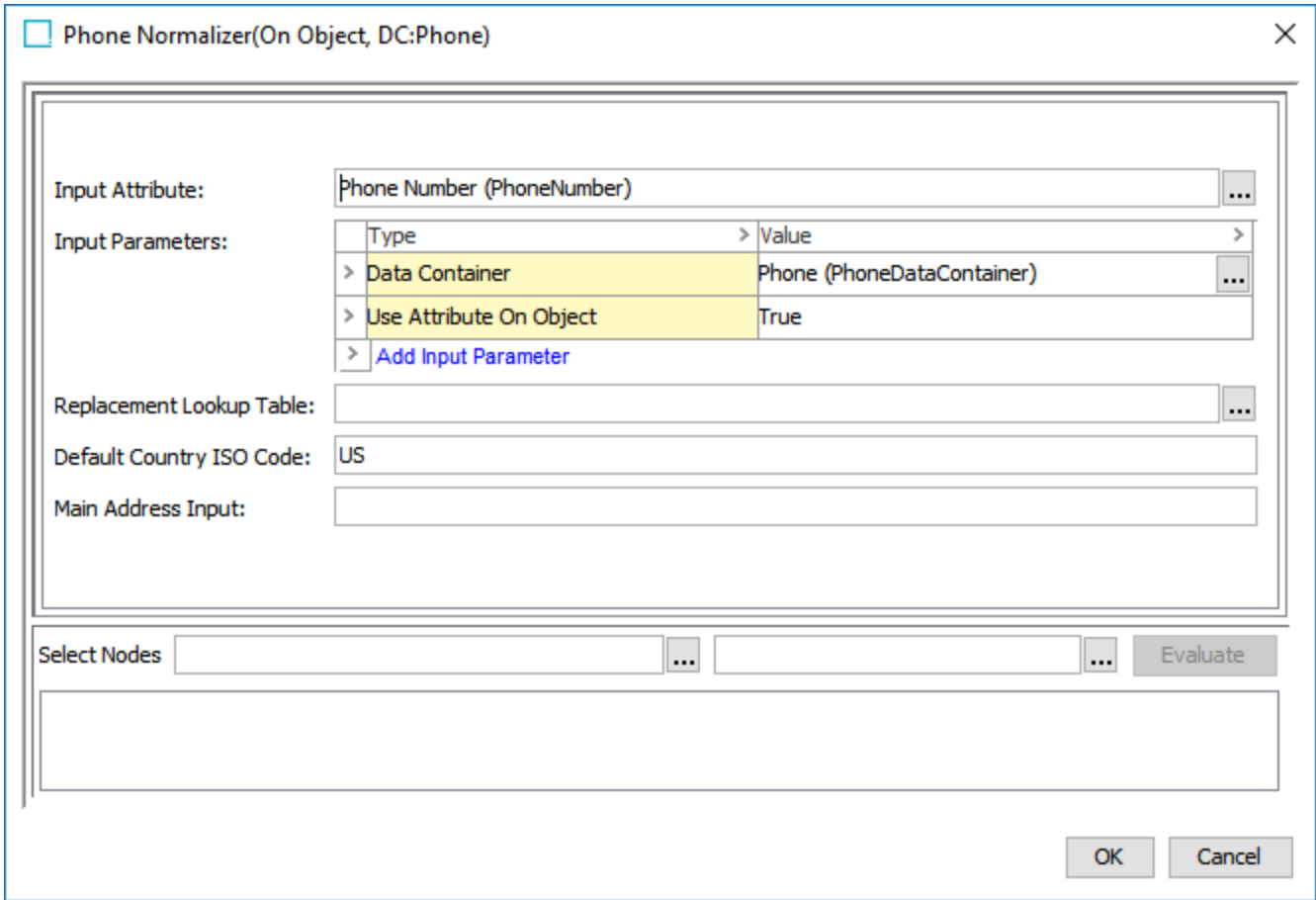
2. For the First Name Attribute parameter, click the ellipsis button (...) and browse or search for a first name attribute to normalize. Repeat this step for the Middle Name Attribute and Last Name Attribute parameters.
3. For Input Parameters, click the Add Input Parameter link, and in the Add Input Parameter popup dialog, select the input type from the dropdown.
 - Input Normalizer: Enter the ID of another Person Name Normalizer to use its output. Typically, this normalizer is written in JavaScript.
 - Data Container: When you select 'Data Container,' click the ellipsis button (...) and browse or search for a data container. The selected data container will have its name data normalized when used in a matcher.
 - Use Attribute on Object: When you select 'Use Attribute On Object,' click the dropdown and select 'True.' This input parameter will normalize the name attributes mapped to the First Name Attribute, Middle Name Attribute, and Last Name Attribute fields, and output them as one value. This input parameter will be configured by default.
4. For the Name Split Regex parameter, enter the RegEx used to split the First Name, Middle Name, and Last Name values into words.
5. For the Replacement Word Lookup Table parameter, click the ellipsis button (...) and select a lookup table. Typically, this is used to remove unwanted words from names. For example, 'Mr.,' 'Dr.,' or 'Von.'

6. Check the Normalizer Accents checkbox if accented characters should be normalized.
7. Click OK when finished.

Phone Normalizer

A Phone Normalizer can normalize phone data for use in the corresponding Phone Matcher.

1. To normalize customer phone data, click the ellipsis button (...) in the Data column to access the configuration.



2. For the Input Attribute parameter, click the ellipsis button (...) and browse or search for a phone attribute to normalize.
3. For Input Parameters, click the Add Input Parameter link, and in the Add Input Parameter popup dialog, select the input type from the dropdown.
 - Input Normalizer: Enter the ID of another Phone Normalizer to use its output. Typically, this normalizer is written in JavaScript.
 - Data Container: When you select 'Data Container,' click the ellipsis button (...) and browse or search for a data container. The selected data container will have its phone data normalized when used in a matcher.

- Use Attribute on Object: When you select 'Use Attribute On Object,' click the dropdown and select 'True.' This input parameter will normalize the phone attribute mapped to the Input Attribute field. This input parameter will be configured by default.
4. For the Replacement Lookup Table parameter, click the ellipsis button (...) and select a lookup table. Typically, this is used to remove invalid phone values.
 5. For the Default Country ISO Code parameter, enter a two-letter ISO code string.
 6. For the Main Address Input parameter, enter the ID of an Address Normalizer. The Country ISO Code value of the normalizer output is used in place of the Default ISO Code, if one exists.

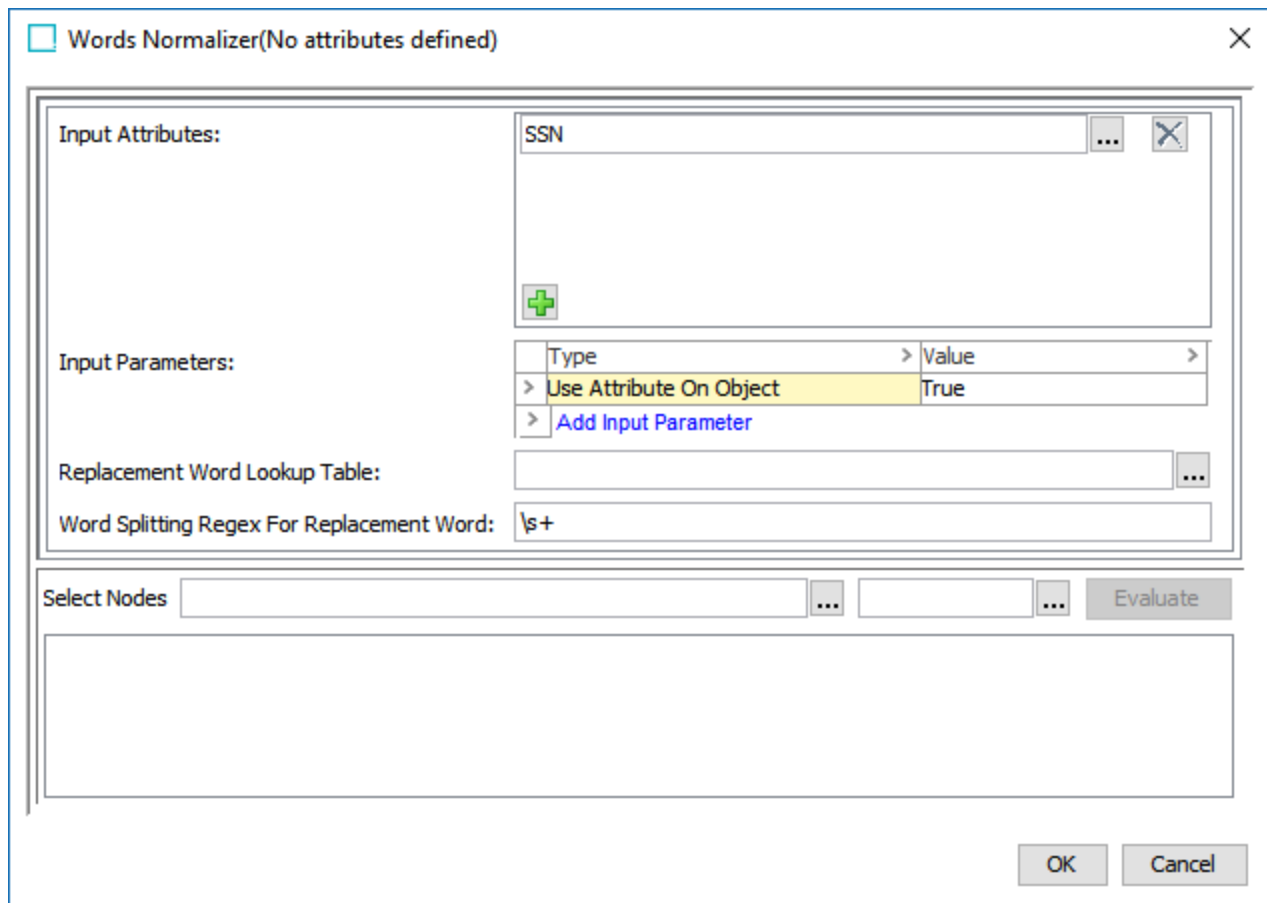
Note: To use this parameter, write a JavaScript address normalizer that outputs a Country ISO code.

7. Click OK when finished.

Words Normalizer

A Words Normalizer can normalize attribute data for use in the corresponding Words Matcher. Multiple attributes can be mapped to the same normalizer. When the corresponding Words Matcher is applied, all mapped attributes will be evaluated.

1. To add a words normalizer, click the ellipsis button (...) in the Data column to access the configuration.



Words Normalizer(No attributes defined)

Input Attributes: SSN

Input Parameters:

Type	Value
> Use Attribute On Object	True
> Add Input Parameter	

Replacement Word Lookup Table:

Word Splitting Regex For Replacement Word: \s+

Select Nodes: [] [] Evaluate

OK Cancel

2. For the Input Attribute parameter, click the plus sign (+) to create a new entry, then click the ellipsis button (...) and browse or search for attributes whose values should be normalized.
3. For Input Parameters, click the Add Input Parameter link, and in the Add Input Parameter popup dialog, select the input type from the dropdown.
 - Input Normalizer: Enter the ID of another Words Normalizer to use its output. Typically, this normalizer is written in JavaScript.
 - Data Container: When you select 'Data Container,' click the ellipsis button (...) and browse or search for a data container.
 - Use Attribute on Object: When you select 'Use Attribute On Object,' click the dropdown and select 'True.' This input parameter will normalize the attribute mapped to the Input Attribute field. This input parameter will be configured by default.
4. For the Replacement Word Lookup Table parameter, click the ellipsis button (...) and select a lookup table. Typically, this is used to remove invalid values.
5. For the Word Splitting RegEx For Replacement Word parameter, enter the RegEx used to split the attribute values into individual words.
6. Click OK when finished.

Customer Data JavaScript Normalizers

For especially complicated solutions, it is possible to expand the capabilities of a customer data normalizer via JavaScript. These JavaScript normalizers can be written to input the normalized values of basic customer data normalizer(s) and manipulate the data in ways the standard normalizer could not. In other words, a JavaScript normalizer inputs a set of strings / values from a basic customer data normalizer and outputs a new set of strings / values.

Note: The JavaScript normalizer should output a completely new set of strings / values and should not overwrite existing strings / values.

These normalizers can also be built completely from scratch rather than enhancing an existing customer data normalizer.

In many situations the more complex JavaScript normalizer would be cited by a corresponding matcher, rather than the basic customer data normalizer.

A typical customer data JavaScript normalizer complies with the following steps:

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 something that can be iterated, such as a set, and returns that set.

JavaScript Function: Bindings, var input = mc.evaluate("addressNormalizer");var address = input.iterator().next(); // There are only one address...

JavaScript Dependencies	
Variable name	Bind to
pdm	Party Data Matching
manager	STEP Manager
mc	Match Expression Context

```

1  var input = mc.evaluate("addressNormalizer");
2  var address = input.iterator().next(); // There is only one address
3  var countryISO = address.getCountryISO();
4
5  // check if country can be made 2 chars ISO3166 alpha2
6  countryISO = replaceLongCountry(countryISO);
7
8  var newAddress = pdm.createAddress().setCountry(address.getCountry()).setRegion(address.getRegion());
9
10 var set = new java.util.HashSet();
11 set.add(newAddress);
12
13 return set;
14
15 function replaceLongCountry(country) {
16     var lCountry = country.toLowerCase();
17     if(lCountry.equals("usa")) return "us";
18     if(lCountry.equals("united states")) return "us";
19     if(lCountry.equals("united states of america")) return "us";
20     return country; // not found, return what was inputted
21 }
22
23

```

Select Nodes: Evaluate

OK Cancel

Match Criteria Matchers

A matcher compares the values of a data element from the two records being compared by the matching algorithm and produces a match score.

The Matcher match score is used in the Rules result formula to calculate the final Match Score. It is also used to enable or disable rules. A rule may specify that it is only relevant, for example, if the address match is above 70, or even if address match is 'True' (depending on the Matchers inner score threshold definition). Many matchers allow setting a default threshold for what is considered 'True' or 'False' in a match rule condition.

Many matchers come with different weights and metrics, allowing detailed calibration of the algorithm to specific datasets. It is significant to do this calibration during match tuning.

The following matchers are available:

Standard Matchers

- Business Function Matcher
- STEP Function Matcher
- JavaScript Function Matcher

Party Data Matchers

- Address Matcher
- Email Matcher
- Organization Name Matcher
- Person Name Matcher
- Phone Matcher
- Words Matcher

Configuration

Use the following steps to configure a matcher for a matching algorithm.

1. To add a matcher to the table, click the 'Add Matcher' link.

Matchers		
ID	Matcher	Comment
> name	Name Matcher(normName)	
> address	Address Matcher(normAddress)	
> email	Email Matcher(normEmail)	
> phone	Phone Matcher(normPhone)	
> Add Matcher		

2. In the Define Matcher dialog, enter an ID for the matcher and use the Matcher Type dropdown(s) to define the match type, then click the 'Add Matcher' button.

Define Matcher
×

ID

Matcher Type

Matcher
▼

Business Function Matcher
▼

Business Function Matcher

Function

JavaScript Function

Address Matcher

Email Matcher

Organization Name Matcher

Person Name Matcher

Phone Matcher

Words Matcher

Important: No two data element, matchers, or match code generators should have the same ID.

3. Populate the Matcher column of the table:

- For matchers, click the ellipsis button (...) to access the configuration. Configuration steps vary depending on the type of matcher selected. For more information on these matchers, see the Standard Matchers and Customer Data Matchers sections below.

Decision Table
✕

Data Elements

ID	Data Elements	Comment
> normName	Name Normalizer (On Object)	
> normAddress	Address Normalizer (On Object, DC:Main Address)	
> normEmail	Email Normalizer (On Object, DC:Email)	
> normPhone	Phone Normalizer (On Object, DC:Phone)	

[Add Data Element](#)

Matchers

ID	Matcher	Comment
> name	Name Matcher (normName)	
> address	Address Matcher (normAddress)	
> email	Email Matcher (normEmail)	
> phone	Phone Matcher (normPhone)	

[Add Matcher](#)

Rules

Edit Conditions Rules Strategy Max

#	address >70	email >70	name >70	phone >70	Result	Comment
> 1					{address*30.0 + name*30.0} / 60.0	
> 2					{name*30.0 + email*30.0} / 60.0	
> 3					{phone*30.0 + email*30.0} / 60.0	

[Add Rule](#)

Match Code Generators

Active	ID	Match Code Generator	Comment
<input checked="" type="checkbox"/>	emailMatchCode	Email Match Code Generator: normEmail, EMAIL #	
<input checked="" type="checkbox"/>	phoneMatchCode	Phone Match Code Generator: normPhone, PHONE #	
<input checked="" type="checkbox"/>	nameAndAddress	Person Name and Address Match Code Generator: normName, null, normAddress, INDIVIDUAL#, true, 0, ...	

[Add Match Code Generator](#)

Match Code Filter

ID	Match Code Filter	Comment
----	-------------------	---------

[Add Match Code Filter](#)

Evaluator

Select Nodes Evaluate

Save Cancel

Standard Matchers

When created as a JavaScript or STEP function, 'mcevaluate' and 'evaluate' are used to assess elements from the data and matcher sections of the decision table and compare their results.


Business Function Matcher

The Business Function Matcher uses a business function to return a match score.

Business Function Matcher: MatchResult, 70, null

Return Type: MatchResult

Condition Threshold: 70

Matcher Function: OrganisationAccountGroupMatch (OrganisationAccountGroupMatch) ... Create New 

Function input parameters: Values:

firstNode (Node)	(First) Current Object (Node) v
secondNode (Node)	(Second) Current Object (Node) v

Select Nodes: Aarone Kirk (558762) ... Aarone Kirk (558774) ... Evaluate

OK Cancel

The business function is typically written in JavaScript.

Edit Operation
✕

JavaScript Function

Bounds:

Variable name	Bounds to
manager	STEP Manager
logger	Logger

Messages:

Variable name	Message	Translations

Input Parameters:

Parameter name	Type	Description
firstNode	Node	
secondNod	Node	

Return Type:

Return Type
Double

JavaScript:

```

1 //var entityHome = manager.getEntityHome();
2 //var targetNode = entityHome.getEntityByID("134537");
3 //var sourceNode = entityHome.getEntityByID("134545");
4
5 //compareReferences(sourceNode, targetNode, "SAPCustomerAccountGroup");
6 return compareReferences(firstNode, secondNod, "SAPCustomerAccountGroup");
7
8
9 function compareReferences(firstNode, secondNode, refTypeID){
10     var refType = manager.getReferenceTypeHome().getReferenceTypeByID(refTypeID);
11     var firstNodeReferences = firstNode.getReferences(refType);
12     var secondNodeReferences = secondNode.getReferences(refType);
13     if(firstNodeReferences && secondNodeReferences && (firstNodeReferences.size()>0 && secondNodeReferenc
14     var firstNodeReference = firstNodeReferences.get(0);
15     var secondNodeReference = secondNodeReferences.get(0);
16     if(firstNode.getReferences(refType).size()==0 && secondNod.getReferences(refType).size()==0){
17         logger.info("OrganisationAccountGroupMatcher true, no references");
18         return new java.lang.Double(100);
19     }
20 }

```

Edit externally

Save Test JavaScript Cancel

JavaScript Function Matcher

JavaScript Function: Bindings, var email1 = mec.evaluate("normEmail", "first");var email2 = mec.evaluate("normEmail", "second");return (email1 && email2 && email1 == email2) ? 1 : 0; X

JavaScript Dependencies

Variable name	Bindings	Matches to
mec		Match Expression Context

Script:

```

1 var email1 = mec.evaluate("normEmail", "first");
2 var email2 = mec.evaluate("normEmail", "second");
3 return (email1 && email2 && email1 == email2) ? 100 : 0;
4
5
6

```

[Edit externally](#)

Select Nodes Evaluate

OK Cancel

This JavaScript matcher above implements a basic email matcher that performs a plain comparison of the emails by comparing normalized email addresses as text strings.

Note: The matcher does not deal with any special cases such as where the normalizer returns strings that are obviously not emails, like empty strings. Resolving such cases is expected to be handled by the normalizer.

STEP Function Matcher

The STEP Function Matcher uses the language of calculated attributes to produce the match score.

Function: matchingLevenshteinDistance(mcevaluate('Phone', 'first'), mcevaluate('Phone', 'second')) X

Formula: Auto Indent | Insert Template | Insert Attribute ID | **Highlighting** ▼

matchingLevenshteinDistance (mcevaluate ('Phone', 'first'), mcevaluate ('Phone', 'second'))

Select Nodes Evaluate

OK Cancel

Customer Data Matchers

These matcher templates are intended for use in customer data solutions:

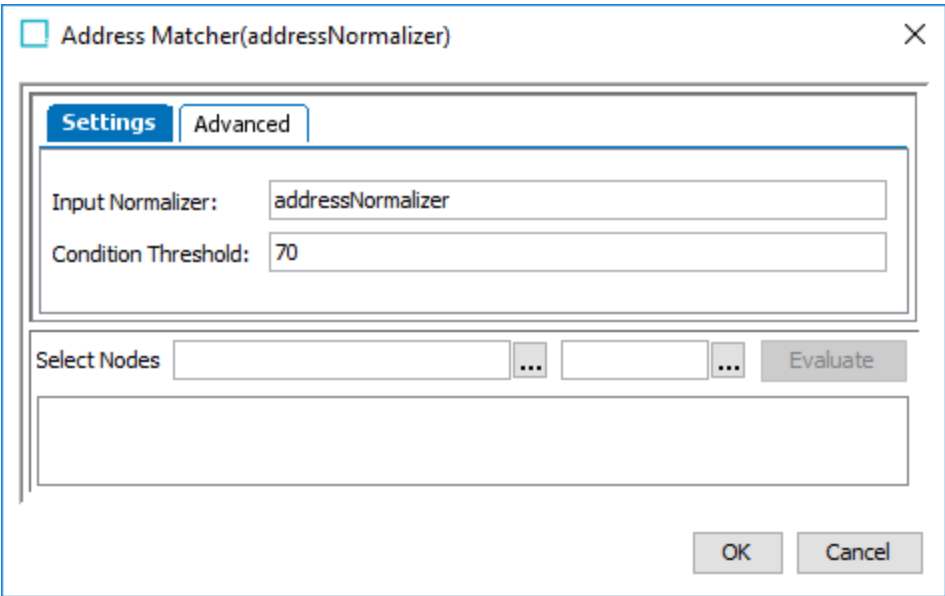
- Address Matcher
- Email Matcher
- Organization Name Matcher
- Person Name Matcher
- Phone Matcher
- Words Matcher

Some matchers give access to lookup tables. For more information on lookup tables, see the **Transformation Lookup Tables** topic in online help **Resource Materials** documentation.

Address Matcher

The Address Matcher compares the normalized address data of two objects and outputs a match score (also called the 'rank score' in Web UI) based on the weighted sum of relevant data elements and match factors. When applied to a rule, the resulting match score is evaluated against a condition threshold and returns 'True' if it meets or exceeds the minimum requirement of the threshold. If the combination of street and postcode, or street and city is a match, the Address Matcher will return a match score indicating a match.

The Address Matcher configuration is split into two tabs: Settings, where the corresponding normalizer is mapped and the condition threshold is established, and Advanced, where different weights are applied to the relevant data elements and match factors.



1. To configure a matcher for customer address data, click the ellipsis button (...) in the Matcher column to access the configuration.

2. The Address Matcher configuration dialog will open in the 'Settings' tab.
3. In the Input Normalizer parameter, enter the ID of the address normalizer the matcher applies to. This field is case sensitive.
4. In the Condition Threshold parameter, enter the minimum score a matcher must achieve to return 'True' on a decision table rule. By default, this score is set to '70.'

Note: An empty Condition Threshold should be used if a variable threshold is required between different rules. For example, one rule requires the matcher to return a score greater than '70,' and another rule needs it to be greater than '75.'

5. Navigate to the 'Advanced' tab. All but one of the parameters included on this tab require that a weight be defined. The matcher considers the individual weights of these elements when they are factored together for the match score (also called the 'rank score' in Web UI).

The screenshot shows the 'Address Matcher(addressNormalizer)' dialog box with the 'Advanced' tab selected. The dialog contains several input fields for weights and factors, and a 'Street Word Splitter Regex' field. At the bottom, there are 'Select Nodes' fields, an 'Evaluate' button, and 'OK' and 'Cancel' buttons.

Parameter	Value
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+

A few things to note:

- The final score is a weighted sum of street and postcode, or street and city.
- The value of Street is split into individual words based on the Street Word Splitter Regex.
- The words for the Street value are split between numbers and text and are compared separately.
- Text words are paired up using Exact, Metaphone 3, and Edit Distance. Text words that are not paired are handled as Missing words.
- Number words are paired up using Exact and Edit Distance. Number words that are not paired are handled as Missing Words.

Required parameters include:

- **Postcode and City Weight:** The relative weight of the Postcode / City score versus the Street score.
- **Street Weight:** The relative weight of the Street score versus the Postcode / City. The Street score is a weighted sum of the Number Words score and the Text Words score
- **Text Word Weight:** The relative weight of the Text Words score versus the Number Words score.
- **Number Words Weight:** The relative weight of the Number Words score versus the Text Words score.
- **Text Exact Word Match Factor:** Determines how pairs that are exact matches should influence the final score.
- **Text Edit Distance Word Match Factor:** Determines how words that are paired via edit distance influence the final score.
- **Number Exact Word Match Factor:** Determines how pairs that are exact matches should influence the final score.
- **Number Edit Distance Word Match Factor:** Determines how words that are paired via edit distance influence the final score.
- **Missing Word Factor:** Determines how much unpaired / missing words should penalize the final result.
- **Word Out Of Order Factor:** Determines how much words that appear out of order should penalize the final result.

6. In the Street Word Splitter Regex parameter, enter the RegEx used to split the Street value into words.

7. Click **OK** when finished.

Email Matcher

The Email Matcher compares the normalized email data of two objects and outputs a match score. When applied to a rule, the resulting match score is evaluated against a condition threshold and returns 'True' if it meets or exceeds the minimum requirement of the threshold. If the email values are a match, the email matcher will return a match score indicting a match.

1. To configure a matcher for customer email data, click the ellipsis button (...) in the Matcher column to access the configuration.
2. In the Input Normalizer parameter, enter the ID of the email normalizer this matcher applies to. This field is case sensitive.
3. In the Condition Threshold parameter, enter the minimum score a matcher must achieve in order to return 'True' on a decision table rule. By default, this score is set to '70.'

Note: An empty Condition Threshold should be used if a variable threshold is required between different rules. For example, one rule requires the matcher to return a score greater than '70,' and another rule needs it to be greater than '75.'

4. Click **OK** when finished.

Organization Name Matcher

The Organization Name Matcher compares the normalized organization name data of two objects and outputs a match score based on the weighted sum of relevant data elements and match factors. When applied to a rule, the resulting match score is evaluated against a condition threshold and returns 'True' if it meets or exceeds the minimum requirement of the threshold. If the organization name values are a match, the organization name matcher will return a match score indicating a match.

1. To configure a matcher for organization name data, click the ellipsis button (...) in the Matcher column to access the configuration.

Organization Name Matcher(OrgnameNORM) ✕

Input Normalizer:	<input type="text" value="OrgnameNORM"/>
Word Alias Table:	<input type="text" value="Org Name Equi Tokens (Org Name Equi Tokens)"/> ...
Exact Word Match Factor:	<input type="text" value="0.999"/>
Alias Word Match Factor:	<input type="text" value="0.9"/>
Concatenation Word Match Factor:	<input type="text" value="0.9"/>
Edit Distance Word Match Factor:	<input type="text" value="0.8"/>
Acronym Word Match Factor:	<input type="text" value="0.8"/>
Missing Word Factor:	<input type="text" value="0.75"/>
Word Out Of Order Factor:	<input type="text" value="1.0"/>
Unmatched Word Factor Table:	<input type="text" value="Org Name Missing Token Factor (Org Name Missing Token Factor)"/> ...
Name Word Splitter Regex:	<input type="text" value="\s+"/>
Condition Threshold:	<input type="text" value="70"/>

Select Nodes

2. In the Input Normalizer parameter, enter the ID of the organization name normalizer this matcher applies to. This field is case sensitive.
3. In the Word Alias Table parameter, click the ellipsis button (...) and select a lookup table to use for substituting certain words. This substitution takes place after the string has been cut into individual words via the Splitter Regex. This parameter should be used to match words that have the same or similar meaning. For example, legal terms such as 'inc' and 'incorporated.'
4. In the Exact Word Match Factor parameter, determine how pairs that are exact matches should influence the final score.
5. In the Alias Word Match Factor parameter, determine how words paired together via aliases should influence the final score.

6. In the Concatenation Word Match Factor parameter, determine how concatenated organization names paired with non-concatenated organization names impact the final score. For example, this match factor could be configured to match 'ACME Systems' with 'ACMESystems.'
7. In the Edit Distance Word Match Factor parameter, determine how words that are paired via edit distance influence the final score. Typically, this match factor is used to catch spelling errors.
8. In the Acronym Word Match Factor parameter, determine how an organization name paired together based off of an acronym influences the final score. For example, the acronym ACME America Inc. could be matched with the organization name Advanced Cellular Medical Engineering of America.
9. In the Missing Word Factor parameter, determine how much unpaired / missing words should penalize the final result.
10. In the Word Out Of Order Factor parameter, determine how much words that appear out of order should penalize the final result.
11. In the Unmatched Word Factor Table parameter, click the ellipsis button (...) and select the relevant lookup table.

Note: The Unmatched Word Factor Table is a lookup table that assigns factors to certain words.

By default, unpaired / missing words will penalize the final score using the Missing Word Factor parameter. However, if an unpaired / missing word appears in this table, it will penalize the final score using the factor in the table rather than the factor configured in Missing Word Factor parameter. This can be used to reduce or increase the penalty that certain words, depending on their significance, have on the final score if they are missing.

12. In the Name Word Splitter RegEx parameter, enter the RegEx used to split the organization name value into words.
13. In the Condition Threshold parameter, enter the minimum score a matcher must achieve to return 'True' on a decision table rule. By default, this score is set to '70.'

Note: An empty Condition Threshold should be used if a variable threshold is required between different rules. For example, one rule requires the matcher to return a score greater than '70,' and another rule needs it to be greater than '75.'

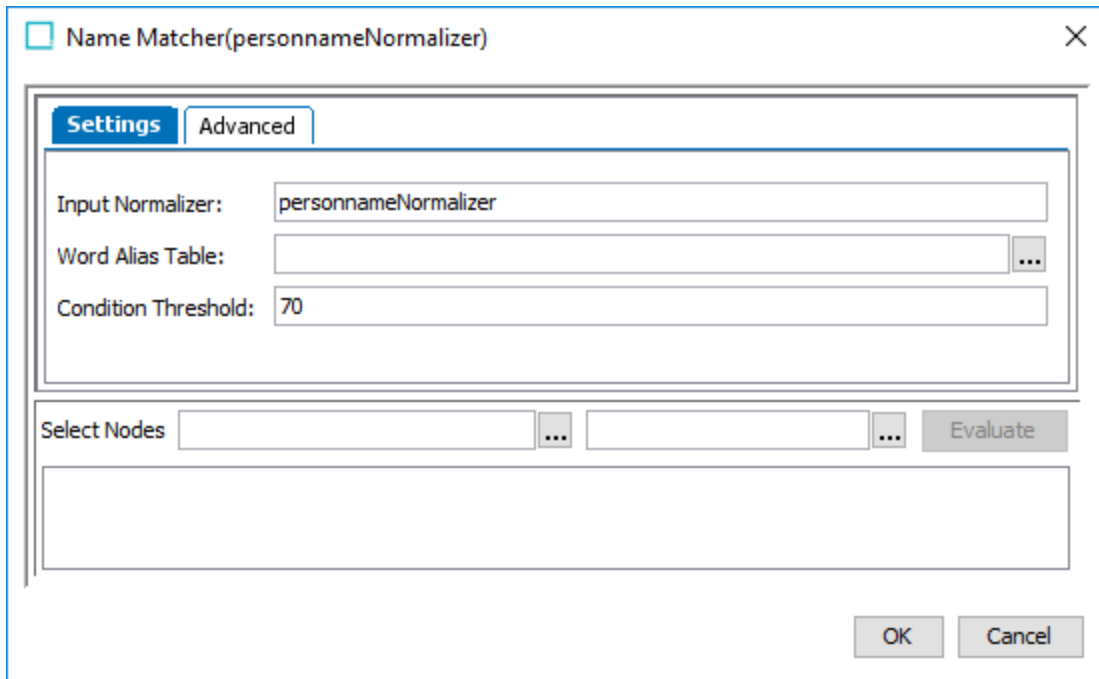
14. Click **OK** when finished.

Person Name Matcher

The Person Name Matcher compares the normalized name data of two objects and outputs a match score based on the weighted sum of relevant data elements and match factors. When applied to a rule, the resulting match score is evaluated against a condition threshold and returns 'True' if it meets or exceeds the minimum requirement of the threshold. If the combination of first name and middle name, and middle name and last name is a match, the Person Name Matcher will return a match score indicating a match.

Note: If customer names are represented in a single field rather than split into first name and last name, use the Words Normalizer / Matcher instead. Middle name is not required to use this matcher.

The Person Name Matcher configuration is split into two tabs: Settings, where the corresponding normalizer is mapped and the condition threshold is established, and Advanced, where different weights are applied to the relevant data elements and match factors.



1. To configure a matcher for customer name data, click the ellipsis button (...) in the Matcher column to access the configuration. The Person Name Matcher configuration dialog will open in the 'Settings' tab.
2. In the Input Normalizer parameter, enter the ID of the person name normalizer this matcher applies to. This field is case sensitive.
3. In the Word Alias Table parameter, click the ellipsis button (...) and select a lookup table to use for substituting certain words. This substitution takes place after the string has been cut into individual words via the Splitter Regex.
4. In the Condition Threshold parameter, enter the minimum score a matcher must achieve to return 'True' on a decision table rule. By default, this score is set to '70.'

Note: An empty Condition Threshold should be used if a variable threshold is required between different rules. For example, one rule requires the matcher to return a score greater than '70,' and another rule needs it to be greater than '75.'

5. Navigate to the 'Advanced' tab. All but two of the parameters included on this tab require that a weight be defined. The matcher considers the individual weights of these elements when they are factored together for the match score.

Not Configured
✕

Settings

Advanced

First Name Weight:	<input type="text" value="1.0"/>
Last Name Weight:	<input type="text" value="1.0"/>
Exact Word Match Factor:	<input type="text" value="1.0"/>
Alias Word Match Factor:	<input type="text" value="0.9"/>
Metaphone3 Word Match Factor:	<input type="text" value="0.5"/>
Edit Distance Word Match Factor:	<input type="text" value="0.4"/>
Initials Match Factor:	<input type="text" value="0.3"/>
Missing Word Factor:	<input type="text" value="0.95"/>
Word Out Of Order Factor:	<input type="text" value="1.0"/>
Unmatched Word Factor Table:	<input type="text"/> ...
Name Word Splitter Regex:	<input type="text" value="\s+"/>

Select Nodes Evaluate

OK Cancel

A couple things to note:

- The final score is a weighted sum of the combined first name and middle name, and the combined middle name and last name.
- The First Name, Middle Name, and Last Name values are split into individual words based on the Name Word Splitter Regex.

Required parameters include:

- **First Name Weight:** The relative weight of the first name / middle name score versus the middle name / last name score.
- **Last Name Weight:** The relative weight of the middle name / last name score versus the first name / middle name score.
- **Exact Word Match Factor:** Determines how pairs that are exact matches should influence the final score.

- **Alias Word Match Factor:** Determines how words paired together via aliases should influence the final score.
 - **Metaphone3 Word Match Factor:** Determines how words paired together via Metaphone 3 should influence the final score.
 - **Edit Distance Word Match Factor:** Determines how words that are paired via edit distance influence the final score.
 - **Initials Match Factor:** Determines how words paired together via initials influence the final score.
 - **Missing Word Factor:** Determines how much unpaired / missing words should penalize the final result.
 - **Word Out Of Order Factor:** Determines how much words that appear out of order should penalize the final result.
7. In the Unmatched Word Factor Table parameter, click the ellipsis button (...) and select the relevant lookup table.

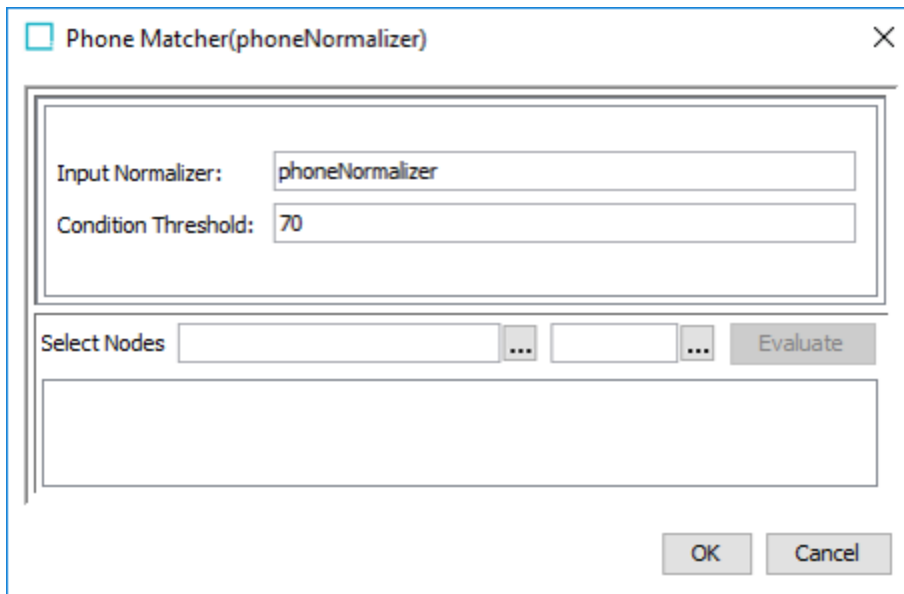
Note: The Unmatched Word Factor Table is a lookup table that assigns factors to certain words.

By default, unpaired / missing words will penalize the final score using the Missing Word Factor parameter. However, if an unpaired / missing word appears in this table, it will penalize the final score using the factor in the table rather than the factor configured in Missing Word Factor parameter. This can be used to reduce or increase the penalty that certain words, depending on their significance, have on the final score if they are missing.

- 8. In the Name Word Splitter Regex parameter, enter the RegEx used to split the first name, middle name, and last name values into words.
- 9. Click **OK** when finished.

Phone Matcher

The Phone Matcher compares the normalized phone data of two objects and outputs a match score. When applied to a rule, the resulting match score is evaluated against a condition threshold and returns 'True' if it meets or exceeds the minimum requirement of the threshold. If the phone values are a match, the Phone Matcher will return a match score indicating a match.



1. To configure a matcher for customer phone data, click the ellipsis button (...) in the Matcher column to access the configuration.
2. In the Input Normalizer parameter, enter the ID of the phone normalizer this matcher applies to. This field is case sensitive.
3. In the Condition Threshold parameter, enter the minimum score a matcher must achieve to return 'True' on a decision table rule. By default, this score is set to '70.'

Note: An empty Condition Threshold should be used if a variable threshold is required between different rules. For example, one rule requires the matcher to return a score greater than '70,' and another rule needs it to be greater than '75.'

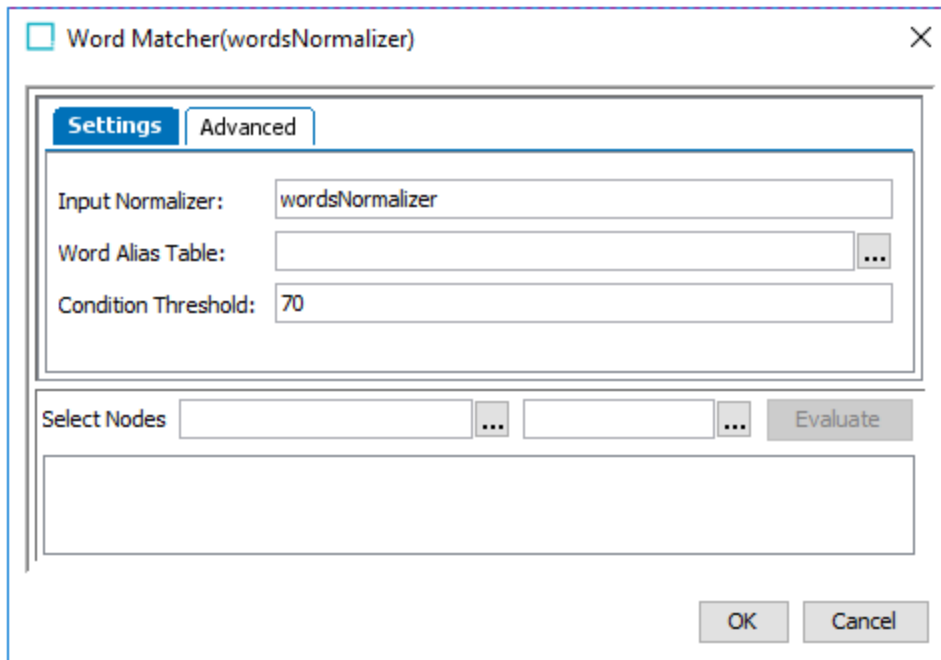
4. Click **OK** when finished.

Word Matcher

The Word Matcher compares the normalized words data of two objects and outputs a match score based on the weighted sum of relevant data elements and match factors. When applied to a rule, the resulting match score is evaluated against a condition threshold and returns 'True' if it meets or exceeds the minimum requirement of the threshold. If the word values are a match, the Words Matcher will return a match score indicating a match.

Note: The Word Normalizer / Matcher is a generic multi-word matcher that can represent a wide range of data, such as, customer names and social security numbers.

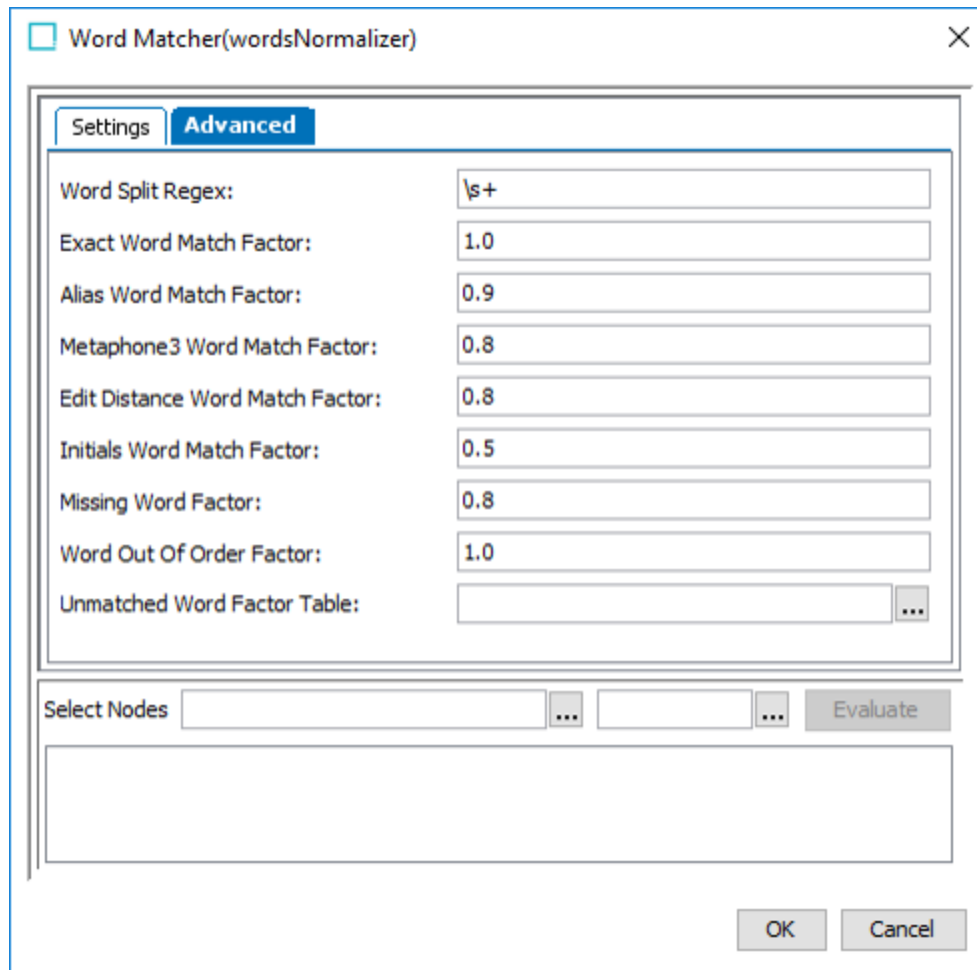
The Word Matcher configuration include these tabs: **Settings** where the corresponding normalizer is mapped and the condition threshold is established, and **Advanced** where different weights are applied to the relevant data elements and match factors.



1. To configure a matcher for different word normalizers, click the ellipsis button (...) in the Matcher column to access the configuration. The Word Matcher configuration dialog will open in the 'Settings' tab.
2. In the Input Normalizer parameter, enter the ID of the words normalizer this matcher applies to. This field is case sensitive.
3. In the Word Alias Table parameter, click the ellipsis button (...) and select a lookup table to use for substituting certain words. This substitution takes place after the string has been cut into individual words via the Splitter Regex.
4. In the Condition Threshold parameter, enter the minimum score a matcher must achieve to return 'True' on a decision table rule. By default, this score is set to '70.'

Note: An empty Condition Threshold should be used if a variable threshold is required between different rules. For example, one rule requires the matcher to return a score greater than '70,' and another rule needs it to be greater than '75.'

5. Navigate to the 'Advanced' tab. All but two of the parameters included on this tab require that a weight be defined. The matcher considers the individual weights of these elements when they are factored together for the match score.



Word Matcher(wordsNormalizer) [X]

Settings **Advanced**

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

Initials Word Match Factor: 0.5

Missing Word Factor: 0.8

Word Out Of Order Factor: 1.0

Unmatched Word Factor Table: [] ...

Select Nodes [] ... [] ... Evaluate

OK Cancel

6. In the Word Splitter RegEx parameter, enter the RegEx used to split the word values into separate words.
7. Required weighed parameters include:
 - **Exact Word Match Factor:** Determines how pairs that are exact matches should influence the final score.
 - **Alias Word Match Factor:** Determines how words paired together via aliases should influence the final score.
 - **Metaphone3 Word Match Factor:** Determines how words paired together via Metaphone 3 should influence the final score.
 - **Edit Distance Word Match Factor:** Determines how words that are paired via edit distance influence the final score.
 - **Initials Word Match Factor:** Determines how words paired together via initials influence the final score.
 - **Missing Word Factor:** Determines how much unpaired / missing words should penalize the final result.
 - **Word Out Of Order Factor:** Determines how much words that appear out of order should penalize the final result.

9. In the Unmatched Word Factor Table parameter, click the ellipsis button (...) and select the relevant lookup table.

Note: The Unmatched Word Factor Table is a lookup table that assigns factors to certain words.

By default, unpaired / missing words will penalize the final score using the Missing Word Factor parameter. However, if an unpaired / missing word appears in this table, it will penalize the final score using the factor in the table rather than the factor configured in Missing Word Factor parameter. This can be used to reduce or increase the penalty that certain words, depending on their significance, have on the final score if they are missing.

10. Click **OK** when finished.

Extending Customer Data Matchers With JavaScript

For especially complicated solutions, it is possible to expand the capabilities of a customer data matcher via JavaScript. These work in much the same way as basic customer data matchers but allow for more flexibility and expanded functionality.

A typical customer data JavaScript matcher complies with the following basic steps:

1. Uses `mc.evaluate` to retrieve the output of a desired normalizer.

Note: 'mc' is a bind to the match expression context.

2. Uses an iterator to access the set of values / strings of both objects being matched.
3. Compares those objects and outputs a match score.

For more information on customer data JavaScript normalizers, see the **Match Criteria Data Elements** topic of this documentation.

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.

The rules strategy determines which rule provides the final score. With rules strategy 'First', the first rule with no condition evaluating to false, provides the score. With rules strategy 'Max', the rule with the highest score, with no condition evaluating to false, provides the score.

Rules							
Edit Conditions		Rules Strategy Max					
#	address >70	email >70	name >70	phone >70	Result	Comment	
1					$(\text{address} * 30.0 + \text{name} * 30.0) / 60.0$	Customers with the same name and address are matched.	
2					$(\text{name} * 30.0 + \text{email} * 30.0) / 60.0$	Customers with the same name and email are matched.	
3					$(\text{phone} * 30.0 + \text{email} * 30.0) / 60.0$	Customers with the same phone and email are matched.	
Add Rule							

In the image above, the strategy is to return the result of the rule with the maximum score via Rules Strategy Max. There are three rules: the first combines address and name, the second combines the name and email, and the last combines phone and email. This allow one of the three to be missing and still return a match score of 100.

Rules							
Edit Conditions		Rules Strategy First					
#	address >70	email >70	name >70	phone >70	ssn >70	Result	Comment
1	True		True			$(\text{address} * 30.0 + \text{name} * 30.0) / 60.0$	
2					True	ssn	
Add Rule							

In the previous image, the Rules Strategy First has two rules. The first combines address and name into a common score, while the second ensures that records sharing a social security number are always matched. If the name and address both are 'True' (that is, the threshold set in their respective matchers is attained), the address + name rule is used, and the ssn rule is never even evaluated.

Matchers are, optionally, 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.

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

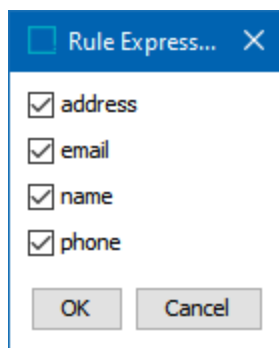
- With the 'Max' rules strategy, the result of all rules with true conditions are calculated, and the maximum result is reported as the match criteria score.

- With the 'First' rules strategy, the result of the top-most rule with true conditions is reported as the match criteria score.

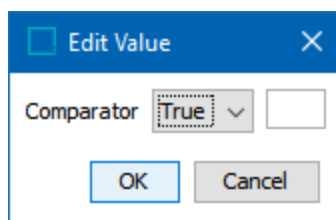
In the above images, the '>70' value displayed in the top row of the three conditions has been provided by the corresponding matcher configurations, each of which states that if it exceeds a condition threshold of '70' it will return 'True.' In some cases, such as when using a more function-based decision table, the threshold will not be established in the matcher configuration and must instead be defined in the table cell.

Configuration

1. Edit Conditions: Matchers can be added / removed from the rules table by clicking the Edit Conditions button and selecting the desired matchers from the 'Rule Expression' dialog.



2. Rules Strategy: The rules strategy determines how the rules are applied when generating a match result. Two options are available from the dropdown: 'First' and 'Max.' Selecting 'First' tells the decision table to look at the rules from top to bottom and base the results on the first rule in which all conditions return 'True.' Alternatively, selecting 'Max' tells the decision table to evaluate all conditions that return 'True' and combine their maximum results.
3. Add Rule: Click the Add Rule link in order to add a row to the table.
4. Rule Condition: If a particular condition should be included in a rule, click the ellipsis button () and add a comparator via the 'Edit Value' popup dialog.



Alternatively, the comparator can be entered directly into the cell. If the condition threshold is defined in the matcher configuration, select 'True.' If it is not defined, manually enter the condition threshold.

5. Rule Result: Each decision table rule requires an expression that drives the logic of the resulting score. In most cases, this is as simple as assigning weights to all conditions relevant to the rule. To create an expression, enter the expression directly into the cell.

Alternatively, click the ellipsis button (...) to access the 'Rule Expression' dialog and click the table radio button. Enter the desired weights for the relevant conditions and click OK. This will automatically generate the relevant expression.

Rule Expression

(address*30.0 + name*30.0) / 60.0

Table

ID	Weight
name	30.0
address	30.0
email	
phone	

OK Cancel

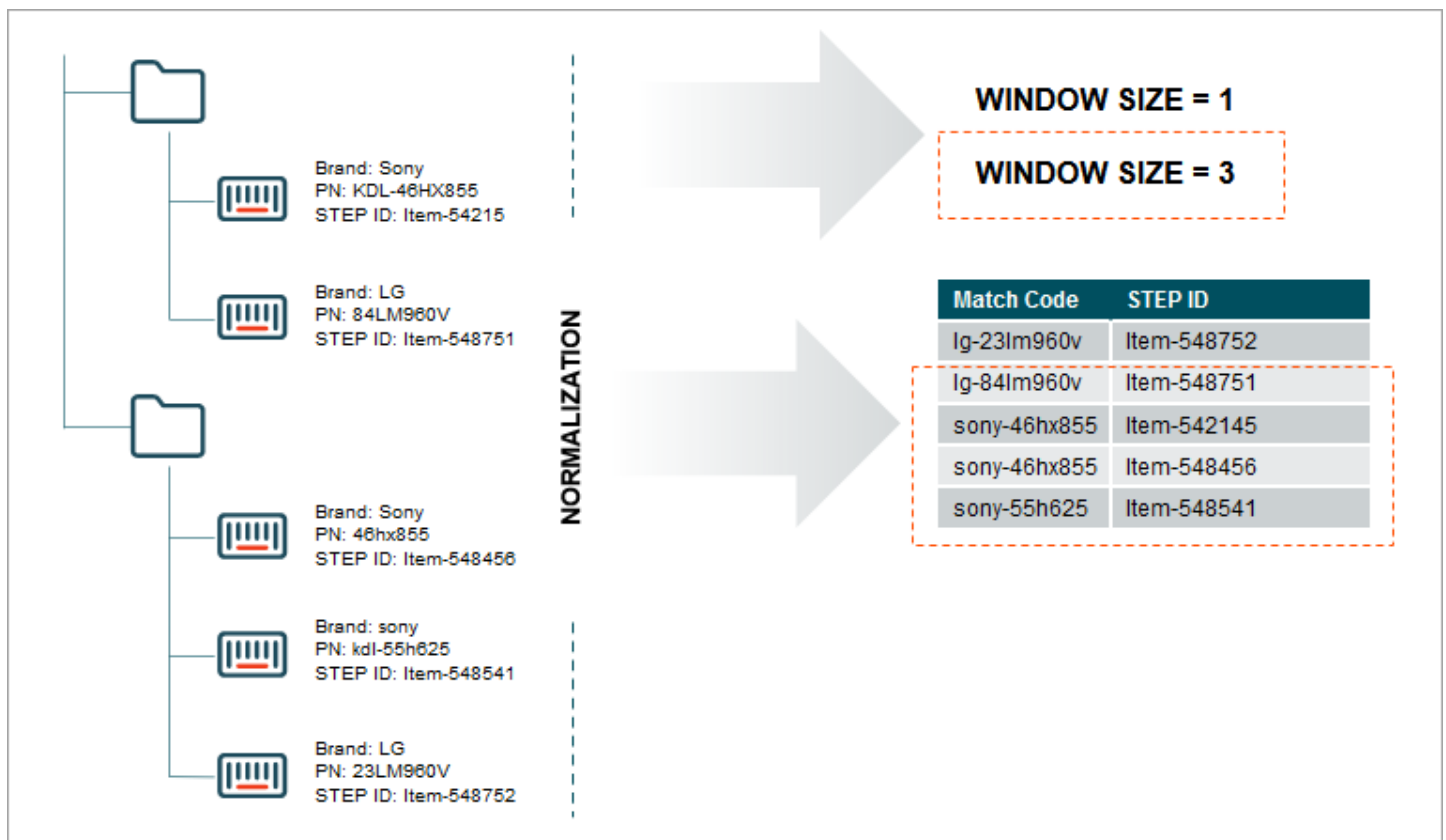
Legacy Match Criteria Without Embedded Match Codes

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.



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.

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: $\text{lowest number} / \text{highest number} * 100$. 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, see the **JavaScript Binds** topic of the online help **Resource Materials** documentation.

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.

The screenshot displays the STIBO SYSTEMS configuration interface. On the left, a tree view shows the navigation structure, with 'Match Code Anon Values' selected under 'Matching Lookup Tables'. The main area is titled 'Match Code Anon Values rev.1.0 - Transformation Lookup Table'. It contains two main sections:

- Transformation Lookup Table:** A table with the following data:

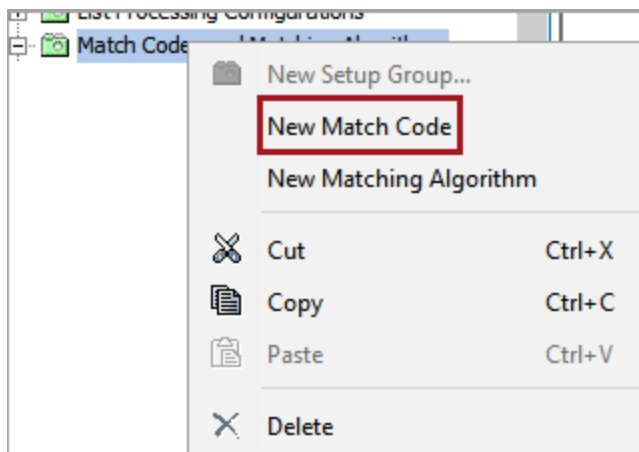
Name	Value
ID	MatchCodeAnonValues
Name	Match Code Anon Values
Object Type	Transformation Lookup Table
Revision	1.0 Last edited by DAGI on Thu Sep 03 10:39:04 CEST 2020
Approved	Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Configurations/Matching Lookup Tables/Match Code Anon Values
Asset URL Attribute	URL
Keywords	abc
OriginalRecord	
- Lookup Table:** A section for defining the mapping between 'From' and 'To' values. It includes checkboxes for '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)', and 'Ignore Case'. Below these is a table:

From	To
> Co	
> Inc	
> Limited	
> Ltd	
>	

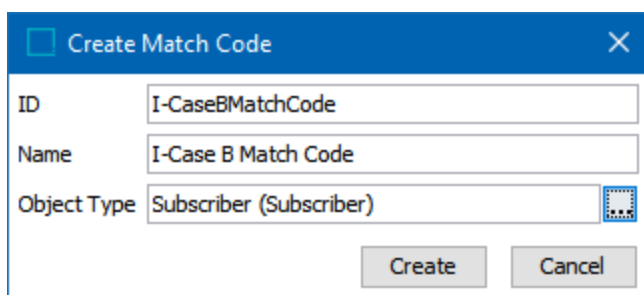
 At the bottom of this section, there is an 'Add Row' button and a '5 Rows' indicator.

At the bottom right of the configuration area, there are buttons for 'Import From Clipboard' and 'Apply'. The status bar at the bottom left shows 'Ready'.

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.



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.

I Case B Match Code - Match Code

Match Code | Match Code Values | Statistics | Log

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...`

- In the **Match Code Window Size** parameter, specify the window size to be used by the matching algorithm. See the **Window Size** section above for details.
- 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.
- 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.
- In the **Match Code Workspace** parameter, specify the workspace to run the match code formula. By default, Main workspace is selected.
- 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.
- In the **Match Code Formula** parameter, click the ellipsis button (...) to open the formula editor and add your match code formula. See 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, see 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, see the **Text Functions** topic in the online help **Resource Materials** documentation.

Method	Description
<code>context.soundex('Stibo')</code>	Returns the Soundex.
<code>context.metaphone3('Stibo')</code>	Returns the primary value for the Metaphone 3.
<code>context.metaphone3alternate('Stibo')</code>	Returns the alternate value for the Metaphone 3.
<code>lth.getLookupTableValue('<asset-id>', 'LookupValue')</code>	For more information, see the Transformation Lookup Tables topic in the online help Resource Materials documentation.

Java Script
Dependencies
✕

Ⓜ Binds

Variable name	> Binds to
node	Current Object

```

1 var normFirstName = mf.normalizeValue(node.getValue("S-FirstNames").getSimpleValue(), true);
2 var normLastName = mf.normalizeValue(node.getValue("S-LastName").getSimpleValue(), false);
3 var normCountry = mf.normalizeValue(node.getValue("S-Country").getSimpleValue(), false);
4 var normZip = mf.normalizeValue(node.getValue("S-ZIP").getSimpleValue(), false);
5
6 var nameAddr = "";
7 if(normFirstName && normLastName && normCountry && normZip) {
8     nameAddr = normFirstName + ":" + normLastName + ":" + normCountry + ":" + normZip;
9 }
10 //
11 var mail = node.getValue("S-Email").getSimpleValue();
12 var phone = node.getValue("S-Phone").getSimpleValue();
13
14 var mcArr = [];
15 if(nameAddr) mcArr.push("NAMEADDR-" + nameAddr);
16 if(mail) mcArr.push("MAIL-" + mail);
17 if(phone) mcArr.push("PHONE-" + phone);
18
19 if(mcArr.length > 0) return mcArr;
20 else return "";

```

[Edit externally](#)

Evaluation Node ... Evaluate

OK Cancel

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.'

```

1      concatenate(
2          replacevaluebylookup("ManufacturerNormalization", value
3          ("Manufacturer")),
4          ":",
5          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:

```

1      listconcatenate(
2          concatenate("MAN-", value("Manufacturer")),
3          concatenate("MPN-", value("ManufacturerPartNumber"))
4      )

```

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.

```

1      {
2          man:= value("Manufacturer"),
3          mpn:= value("ManufacturerPartNumber")
4      }
5      listconcatenate(
6          if(len(man)!=0, concatenate("MAN-", man), ""),
7          if(len(mpn)!=0, concatenate("MPN-", mpn), "")
8      )

```

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.

```

1      function normalizeValue(value) {
2      if(value) {
3      var normVal = value + "";
4          normVal = normVal.toLowerCase();
5          normVal = normVal.replace(/[^\\w]|_/g, "");
6      return normVal;
7      }
8      else {
9      return "";
10     }
11    }

```

normalizeStreet

This example demonstrates how to access lookup tables. For more information on lookup tables, see 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.

```

1      function normalizeStreet(input, lookupTableHome) {
2      var output = "";
3      if(input) {
4          input = input + "";
5          input = input.toLowerCase();
6          input = input.replace(/[\.\,|#]_|/g, "");
7      var inArr = input.split(" ");
8      var outArr = [];
9      for(var i = 0; i < inArr.length; i++) {
1         outArr.push(lookupTableHome.getLookupTableValue
0         ("AddressAbbreviations", inArr[i]));
1         }
1         }
1         for(var j = 0; j < outArr.length; j++) {
2         }
1         output += outArr[j];
3         }
1         if(j != outArr.length - 1) {
4         }
1         output += " ";
5         }
1         }
6         }
1         }
7         }
1         }
8         }
1         return output;
9         }
2         }
0     }

```

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	
<input type="checkbox"/>	Replace with default value when no matches are found (Value Substitution only):
<input checked="" type="checkbox"/>	Replace with a source value when no matches are found and default value is empty (Value Substitution only)
<input checked="" type="checkbox"/>	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 'coreMatchingFunctions.'

```

1      var street1 = mec.evaluate("normStreet", "first");
2      var street2 = mec.evaluate("normStreet", "second");
3      return coreMatchingFunctions.levenshteinDistance(street1,
               street2);

```

☐ Edit Operation
✕

Execute JavaScript ▾

Binds:

Variable name	>	Binds to	>
matchExpressionContext		Match Expression Context	
coreMatchingFunctions		Matching Functions	

Messages:

Variable name	>	Message	>	Translations	>

JavaScript:

```

1 var street1 = matchExpressionContext.evaluate("normStreet", "first");
2 var street2 = matchExpressionContext.evaluate("normStreet", "second");
3
4 return coreMatchingFunctions.levenshteinDistance(street1,street2);

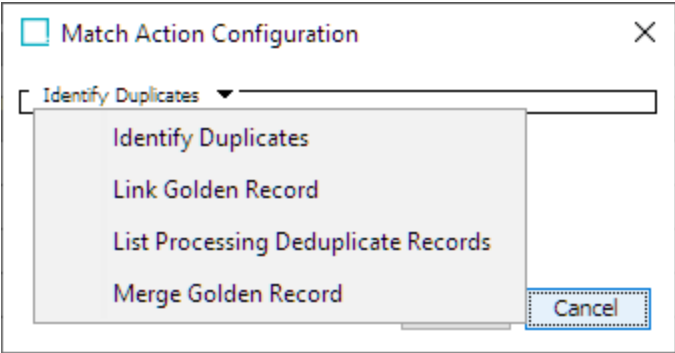
```

[Edit externally](#)

Save Test JavaScript Cancel

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, see 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

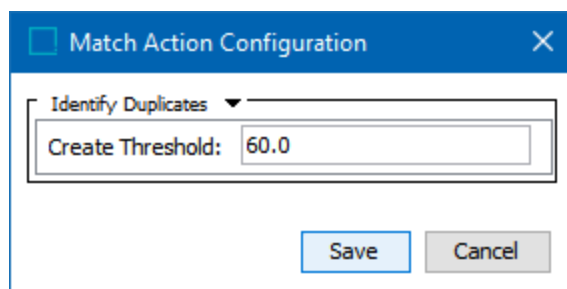
The Identify Duplicates match action helps determine if duplicates exist in a dataset and allows users to manually confirm, reject, merge, and delete duplicates with limited impact on existing functionality.

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, see 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.



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, see 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

- **Configuring a Deduplication Clerical Review** topic

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'.

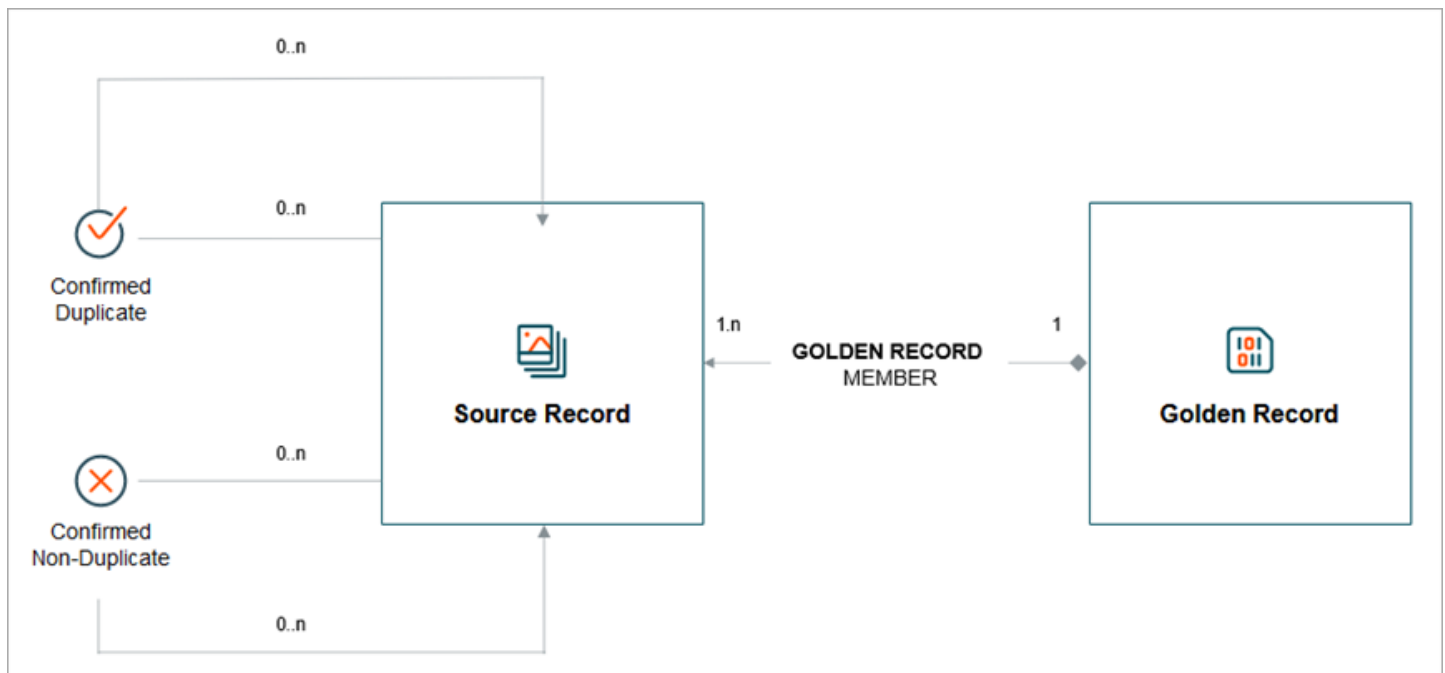
- In Product MDM, Match and Link is commonly used in automating the creation and maintenance of sell-side products as golden records, based on buy-side products as source records.
- In Customer MDM, Match and Link is commonly used for resolving household entities as golden records using individual customer entities as source records.

A detailed setup using Match and Link is described in the **PIM for Retail Data Onboarding** topic of the **PIM for Retail** section of the **Product MDM Solution Enablement** 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.

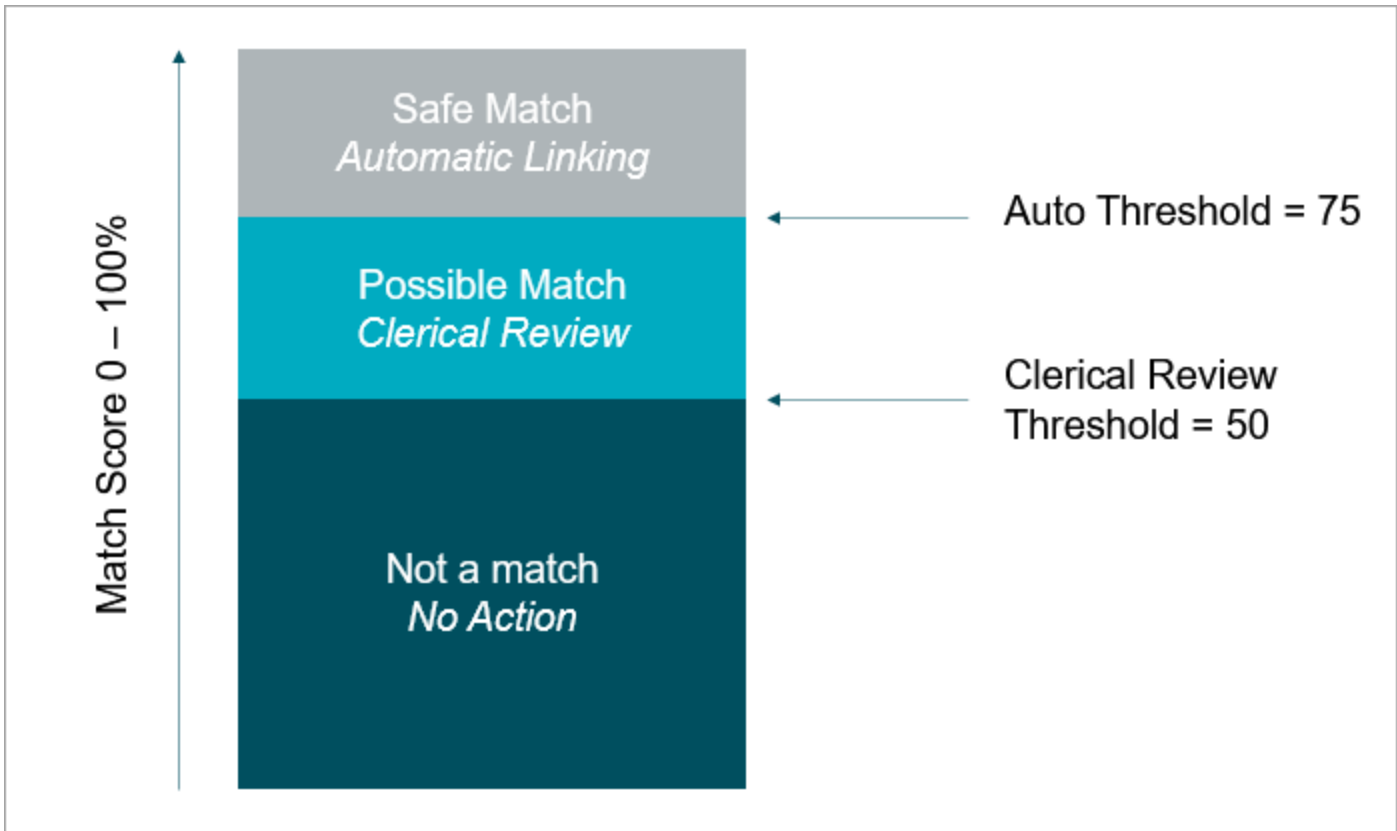
Match Score

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.

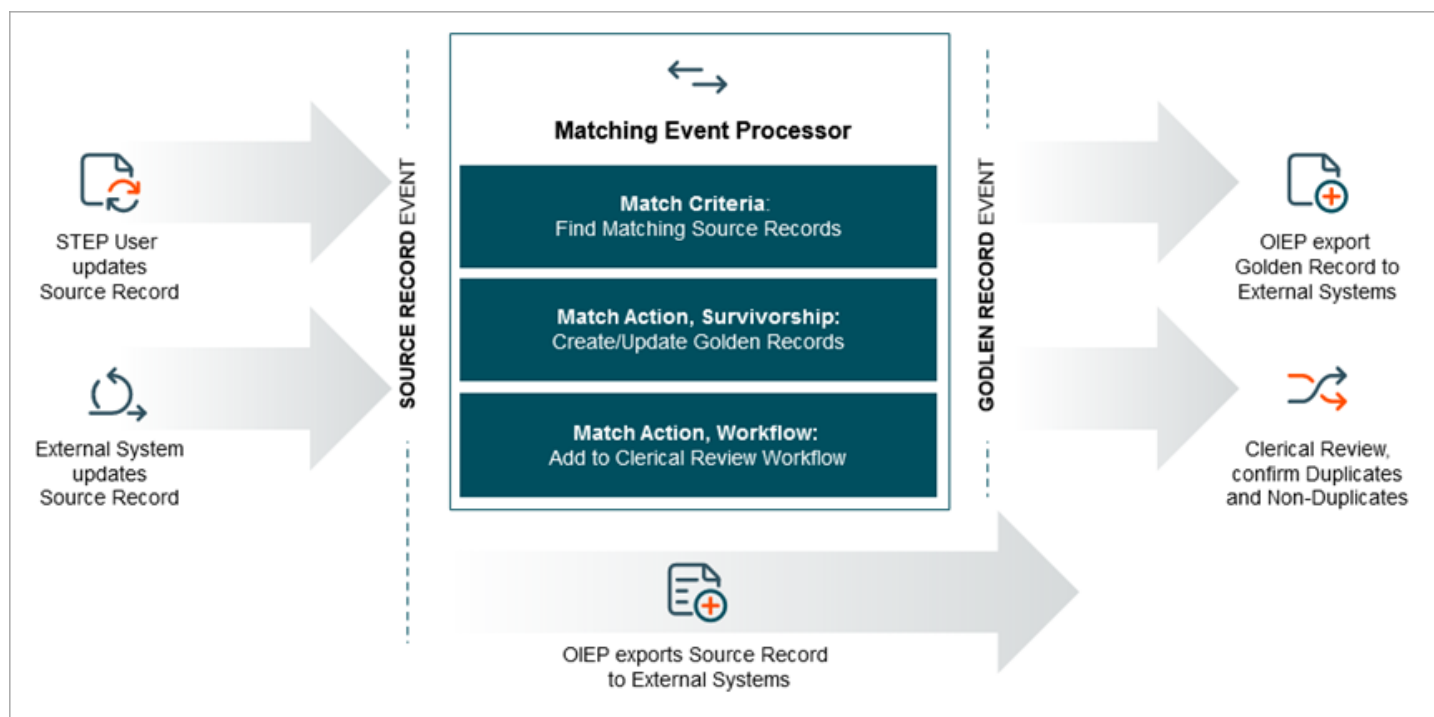
Potential duplicates enter the selected clerical review workflow where a user then sets one of the following reference types:

- **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.



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, see **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, see 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.

Name	Value	Description
Match Tuning Asset Object Types		
	MTC-CSV	Asset object types which can be used for storing uploaded data files for Match Tuning Configurations
	XML File	
Matchable Object Types		
	Address	Object types which can be matched using Match Codes and Matching Algorithms
	CD_Customer	
	Contact	
	Customer	
	Customer Record	
	External Item	
	External Item2	
	External Item Enrichment Record	
	GR	
	Individual Customer	
	Prospect	
	Subscriber	
Confirmed Justification Attribute		
	Justification	Attribute used for storing justification comment on confirmed relations
Data Source Attribute		
	Source	Attribute used for storing ID of Data Source on source-member records (optional as only used for source records in linked golden records setup)
Duplicate Reference Types		
	Confirmed Duplicate Contact	Reference types used throughout this system as duplicate types
	MergeDup	
	External Item Duplicate	
	Subscriber Duplicate	
	Confirmed Duplicate Address	
	Confirmed Duplicate Prospect	
	Confirmed Duplicate Individual	
Non-Duplicate Reference Types		
	Confirmed Non Duplicate Contact	Reference types used throughout this system as non-duplicate types
	MergeNonDup	
	Subscriber Non Duplicate	
	External Item Non Duplicate	
	Confirmed Non Duplicate Address	
	Confirmed Non Duplicate Prospect	
	Confirmed Non Duplicate Individual	

- 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	MTC-CSV XML File	Asset object types which can be used for storing uploaded data files for Match Tuning Configurations
Matchable Object Types	Address	Object types which can be matched using Match Codes and Matching Algorithms
	CD_Customer	
	Contact	
	Customer	
	Customer Record	
	External Item	
	External Item2	
	External Item Enrichment Record	
	Individual Customer	
	Prospect	
Subscriber		
Confirmed Justification Attribute	Justification	Attribute used for storing justification comment on confirmed relations
Data Source Attribute	Source	Attribute used for storing ID of Data Source on source-member records (optional as only used for source records in linked golden records setup)
Duplicate Reference Types	Confirmed Duplicate Contact	Reference types used throughout this system as duplicate types
	MergeDup	
	External Item Duplicate	
	Subscriber Duplicate	
	Confirmed Duplicate Address	
	Confirmed Duplicate Prospect	
Non-Duplicate Reference Types	Confirmed Duplicate Individual	Reference types used throughout this system as non-duplicate types
	Confirmed Non Duplicate Contact	
	MergeNonDup	
	Subscriber Non Duplicate	
	External Item Non Duplicate	
	Confirmed Non Duplicate Address	
	Confirmed Non Duplicate Prospect	
	Confirmed Non Duplicate Individual	

Save Restore live settings Save pending Cancel

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 (X) 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.

- 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.
 - **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.

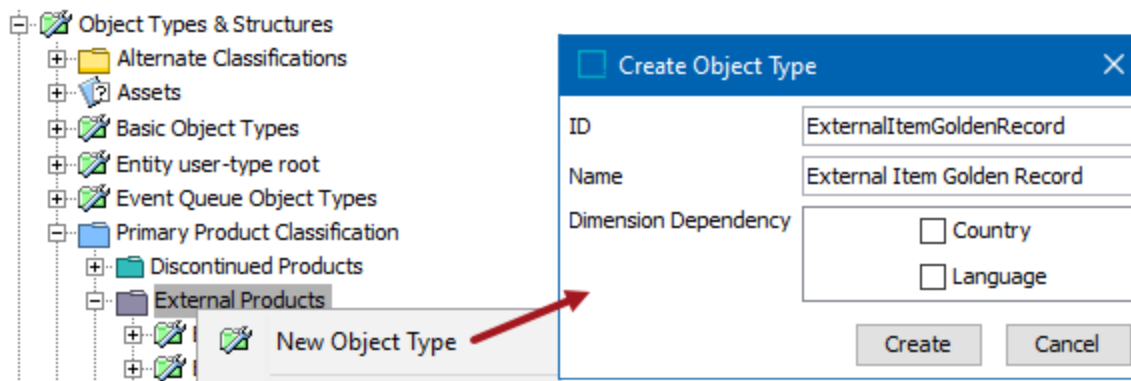
Configuring the 'Link Golden Record' Object Type

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.

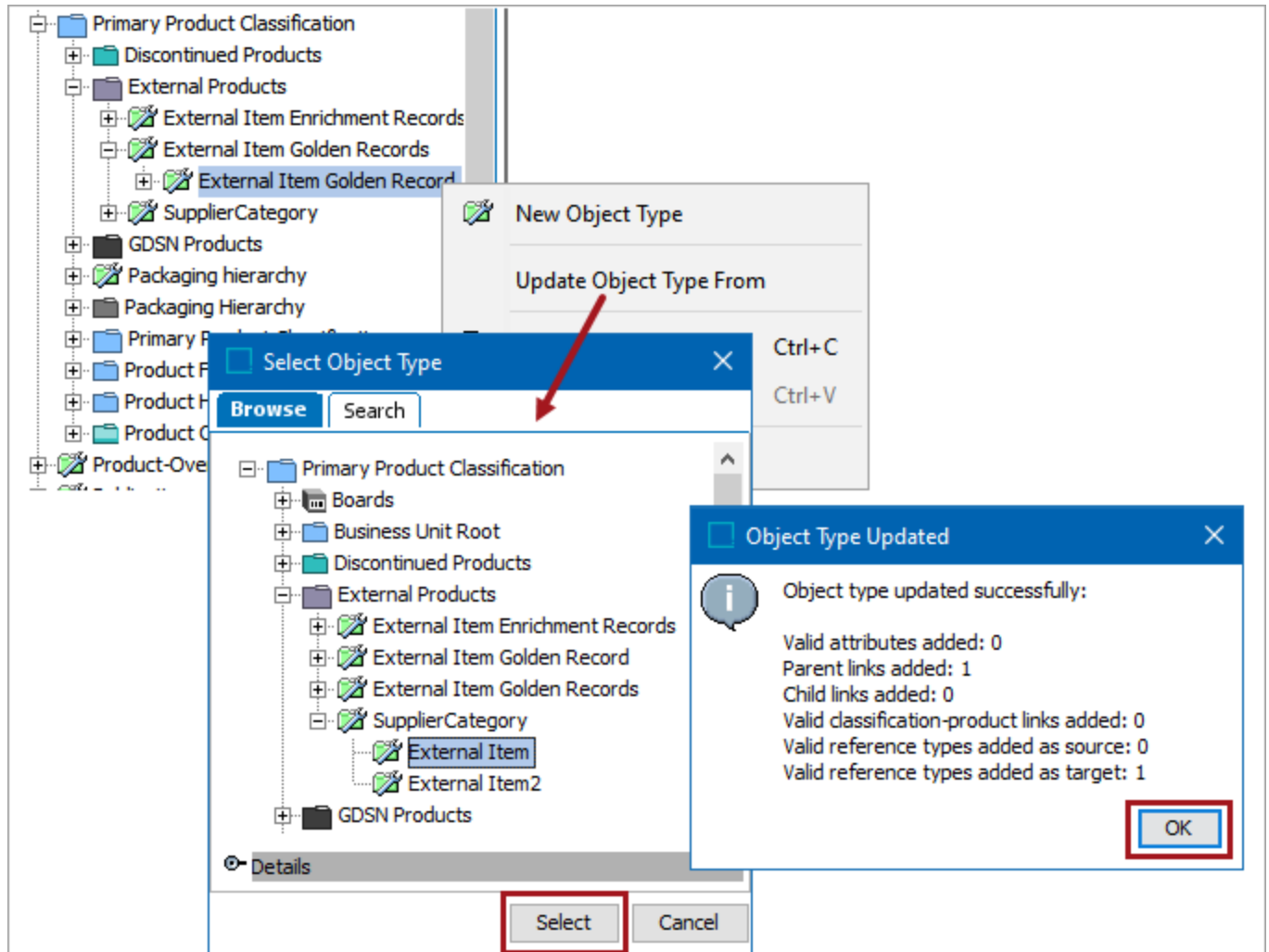
Important: The 'link golden record' object type must be different from the object type used for source objects.

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.



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.

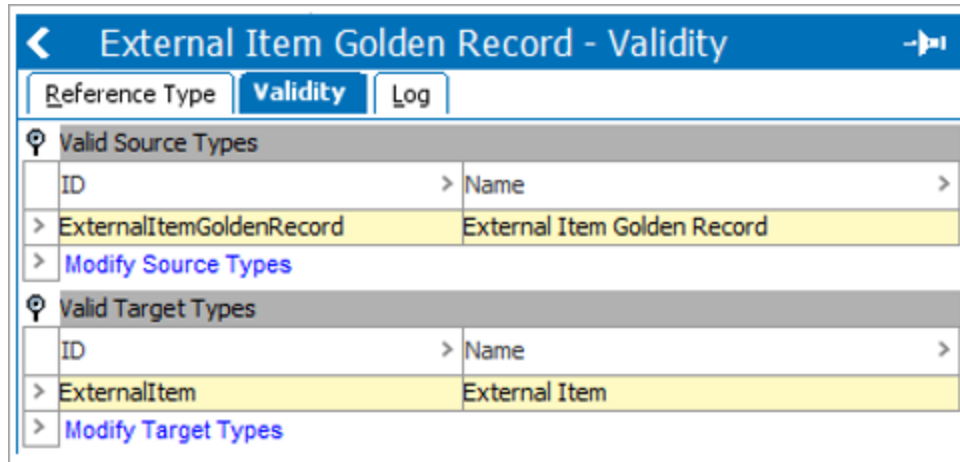


3. On the Description flipper, set the **ID Pattern** parameter to use the **[id]** variable.

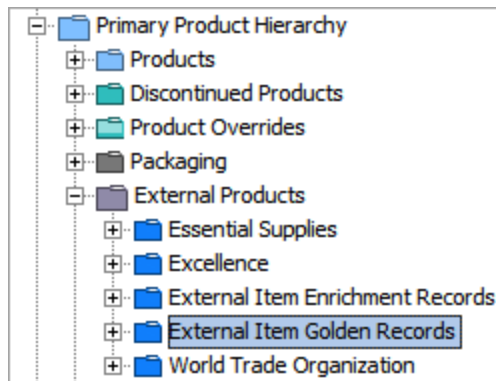
External Item Golden Record - Object Type		
Object Type		References
Log		
Description		
Name	>	Value
ID	>	ExternalItemGoldenRecord
Name	>	External Item Golden Record
Last edited by	>	2021-01-27 09:54:17 by USERE
Name Pattern	>	
ID Pattern	>	ExternalItemGoldenRecord-[id]

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).



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.



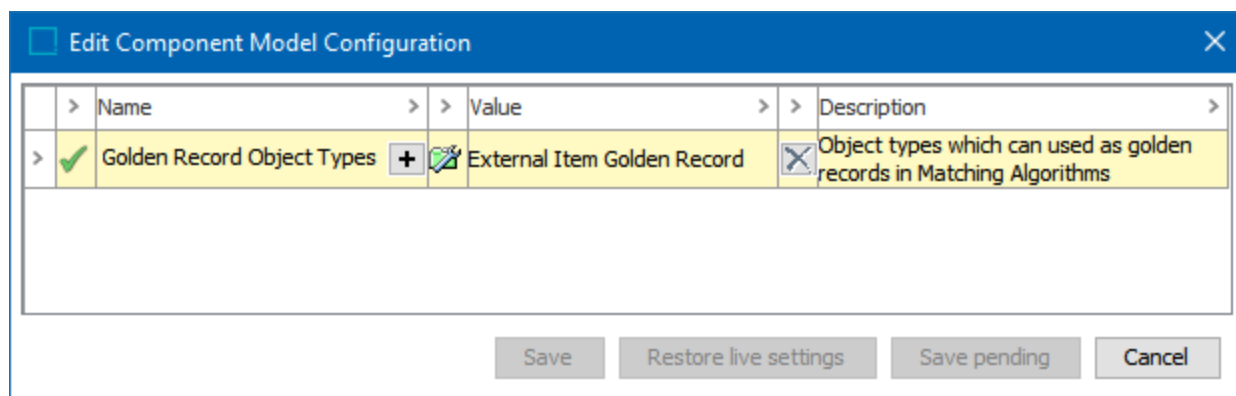
Configuring the Matching - Link Golden Record Component Model

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.



- 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.

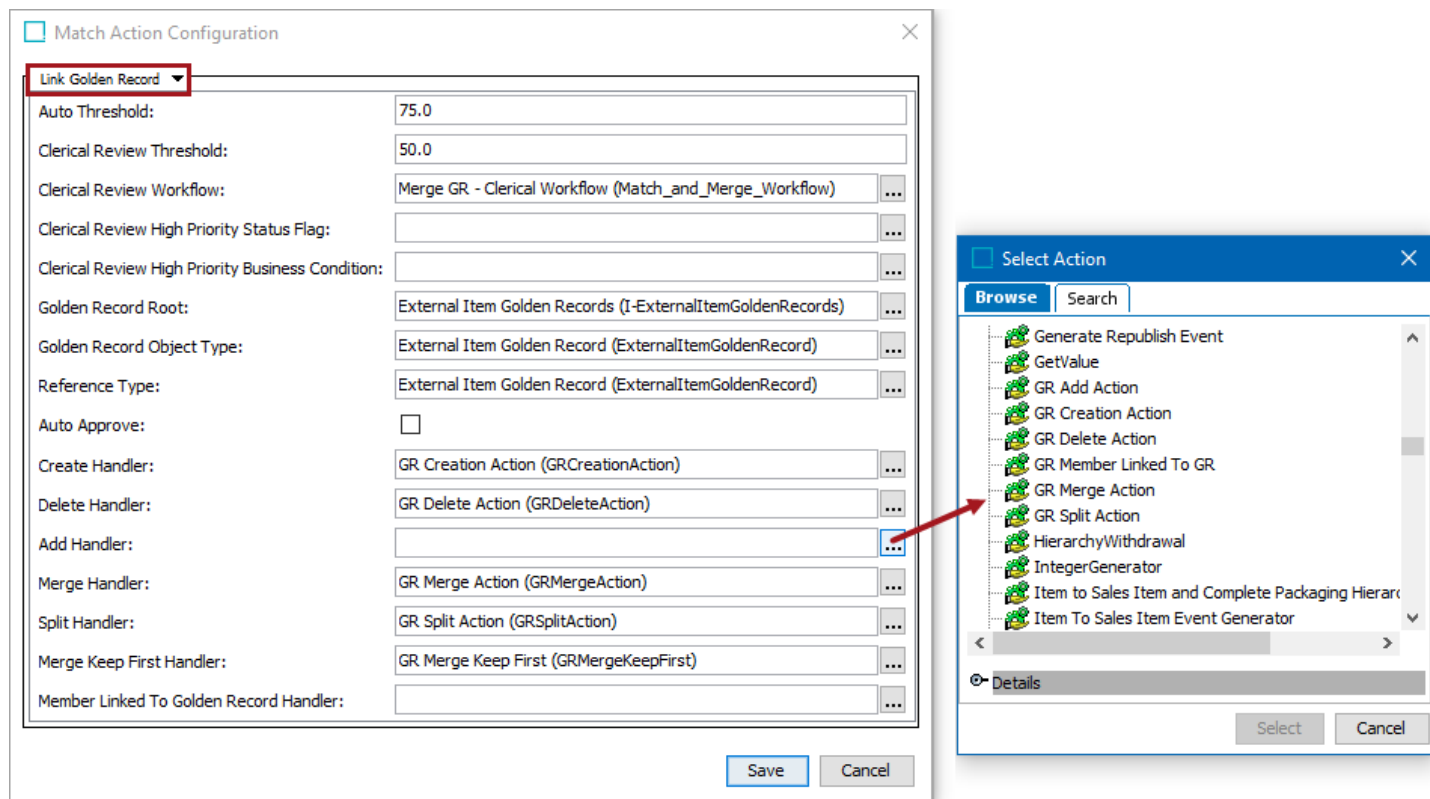
Working with a golden record setup often requires specific actions to handle a golden record change (created, deleted, merged, split, etc.). In these cases, the matching algorithm can be configured to call a business rule via a handler in order to allow for more granular processing of events. For example, 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.



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, see 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:

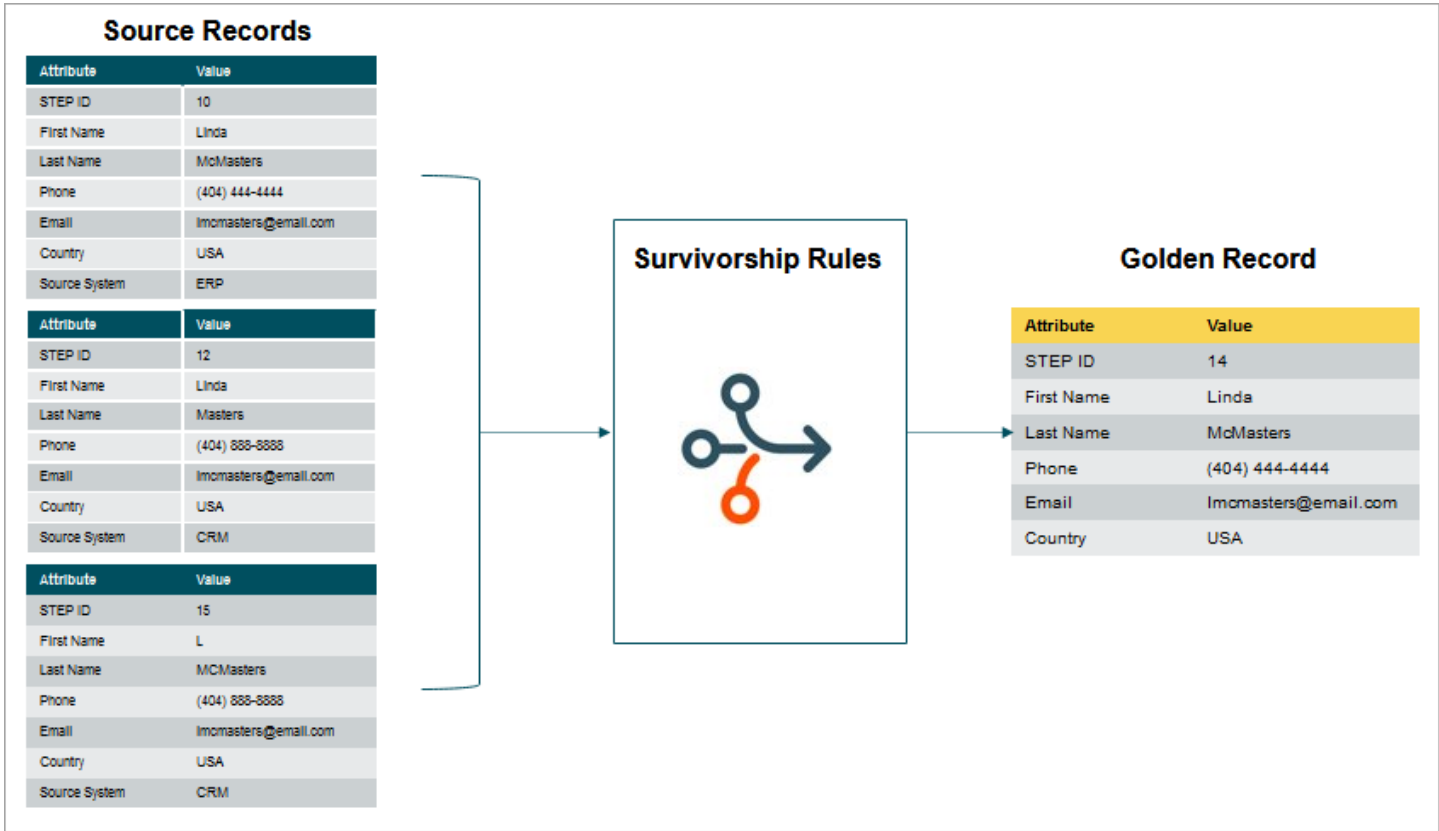
- For the **Create Handler** parameter, the selected business action runs on the golden record after it has been created and has initial source object links, but before survivorship rules run.
- For the **Delete Handler** parameter, the selected business action runs after the golden record is deleted. 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 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.



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, see 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, see 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, see 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, see 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 / Super User Guide** 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 / Super User Guide** documentation.

For more information, see the **Maintaining an Event Processor** topic of the **System Setup / Super User Guide** 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, see **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.

Node	Duplicate Candidate
> Sean Duke	Sean Duke
> Anthony Cooley	Tony Cooley
> Bob Franklin	Robert Franklin

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.

- Confirmed Non Duplicates

[Matching Algorithm](#) | **[Match Result](#)** | [Score Distribution](#) | [Statistics](#) | [Confirmed Duplicates](#) | [Confirmed Non Duplicates](#) | [Log](#)

[Pair Export](#) | [Pair Export Confirmed](#) | [Pair Import Confirmed](#)

Showing page 1 Sort Ascending [Add Additional Matching Algorithm](#)

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.

- Confirmed Non Duplicates

[Statistics](#) | [Confirmed Duplicates](#) | **[Confirmed Non Duplicates](#)** | [Log](#)

[Matching Algorithm](#) | [Match Result](#) | [Score Distribution](#)

Showing page 1

Node 1	> Non Duplicate	> Date	> Justification
> Amos Charles III	Austin Copeland	Wed May 18 12:51:11 EDT 2016	
> Aline			

Compare

	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.

The screenshot shows the 'Confirmed Non Duplicates' interface. At the top, there are tabs for 'Matching Algorithm', 'Match Result', 'Score Distribution', 'Statistics', 'Confirmed Duplicates', 'Confirmed Non Duplicates', and 'Log'. Below the tabs are buttons for 'Pair Export', 'Pair Export Confirmed', and 'Pair Import Confirmed'. A status bar indicates 'Showing page 1' and 'Sort Ascending'. A table lists duplicate candidates with columns for Node, Duplicate Candidate, Date, and Score (%). A dialog box titled 'Select Matching Algorithm' is open, showing a dropdown menu with 'Case B Compare Algorithm' selected and 'OK' and 'Cancel' buttons.

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

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.

The screenshot shows a table with columns for Node, Duplicate Candidate, Date, and Score (%). The 'Sean Duke' row is highlighted. A context menu is open over the first column, showing three options: 'Compare', 'Confirm Duplicate', and 'Reject Duplicate'. The 'Confirm Duplicate' option is highlighted in blue.

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.

The screenshot shows the 'Confirm Duplicate' dialog box. It has a title bar with a close button. The main area contains the text 'Please type in reason for confirming objects as duplicates:' followed by a text input field. At the bottom, there are 'OK' and 'Cancel' buttons.

- 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 sees 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

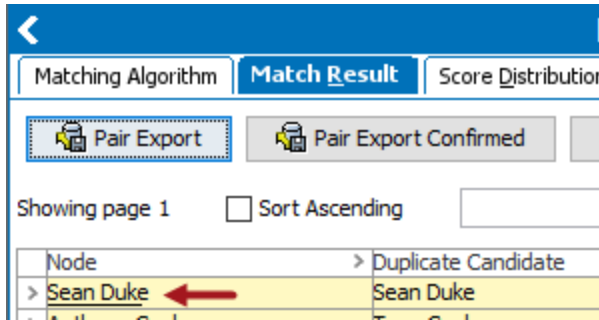
Duplicate information can also be viewed directly on each link golden record source record in the Tree.

Choose a method to view the object:

- In the Tree, select the relevant source record and click the 'Matching' tab.

No Title			
Subscriber	Data Containers	References	Referenced By
Matching			
Data Profile			
Proof View			
Stat			
☰ 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.



- For the link golden record, the 'Link Golden Record' tab display the source records that are linked to it.

Jackson, Hudsonville rev.0.2 - Link Golden Record

Household | Data Containers | References | Referenced By | **Link Golden Record** | Matching | Proof View | Status | ST

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)		RJKNSN	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.

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)	
[All Elements]	Anthony C	Merge result	Tony Cooley (Delete)	Keep this instead
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...

- 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, see the **Adding Widgets to a Homepage** topic in the **Web User Interfaces / Web UI Setup and User Guide** 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

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.

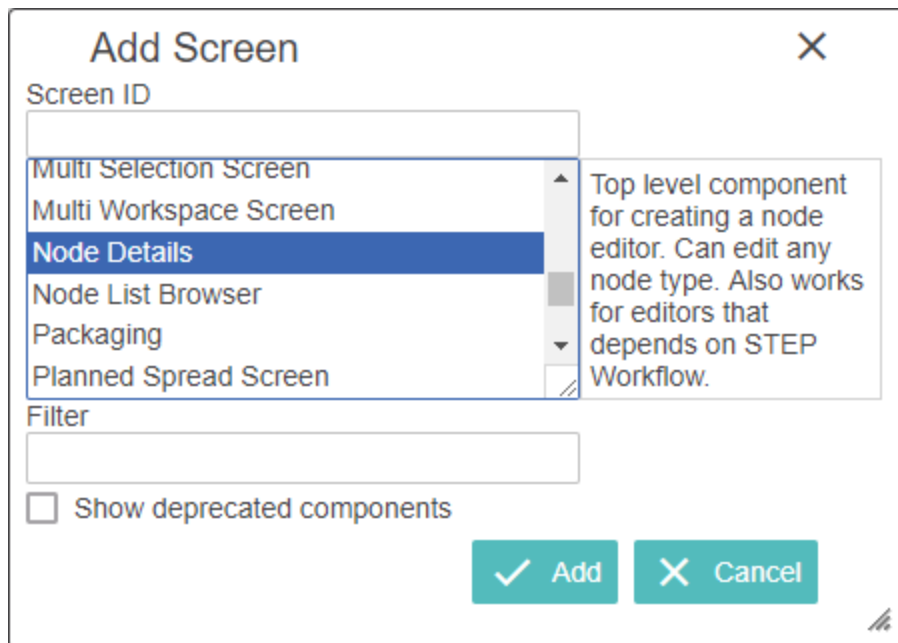
Prerequisites

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, see the **Workflows** documentation.
3. For details on configuring the Web UI for Merge Golden Record clerical reviews, see 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.

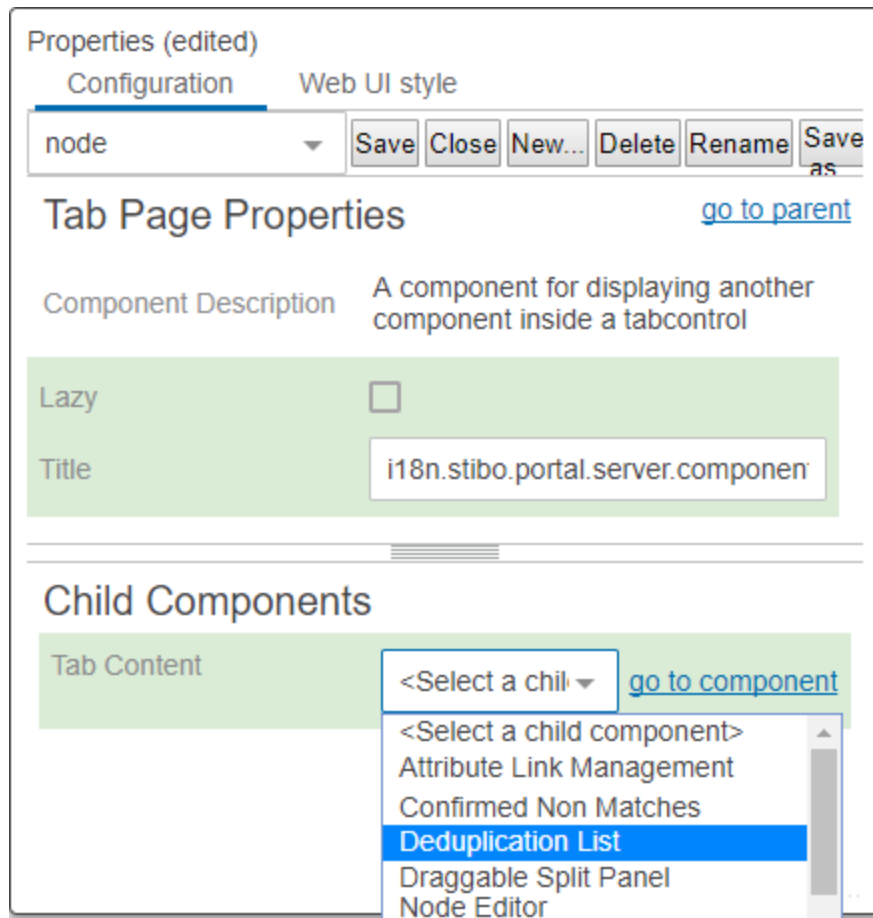
1. Log in to the Web UI and click the gear wheel icon (⚙️) to enter design mode.
2. Click **New**, and then select the **Node Details** screen type.



3. Enter a **Screen ID** and click **OK**.
4. In the **Child components** area, in the **Main** dropdown, select **Tab Control**.

The screenshot shows a configuration window titled 'Properties' with two tabs: 'Configuration' and 'Web UI style'. The 'Configuration' tab is active, showing a dropdown menu with 'node' selected. To the right of the dropdown are buttons for 'Save', 'Close', 'New...', 'Delete', 'Rename', and 'Save as'. Below this is a section titled 'Node Details Properties'. The main section is 'Child Components', which contains three rows: 'Below Title', 'Main', and 'Buttons'. Each row has a dropdown menu with '<Select a chil...' and a blue link labeled 'go to component'. The 'Main' row is highlighted in green.

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**.



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 Duplicate From Grid**

Action and Reject Duplicate From Grid Action.

17. Click **Save** to save the changes.

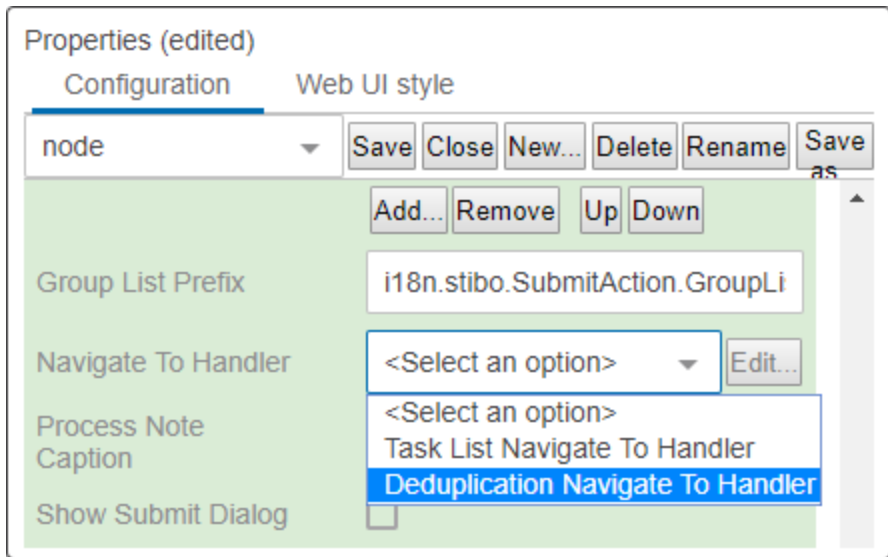
The 'Hide Equal' and 'Mark Different' actions will automatically appear alongside any other actions configured on the Deduplication List. For more information, see the **Comparing Data Using Hide Equal and Mark Different** section of the **Web User Interfaces / Using a Web UI** documentation.

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 see the **Navigate To Handler** list, drag the sizing handle until you can see all properties.

6. In the **Navigate To Handler** list, select **Deduplication Navigate To Handler**.



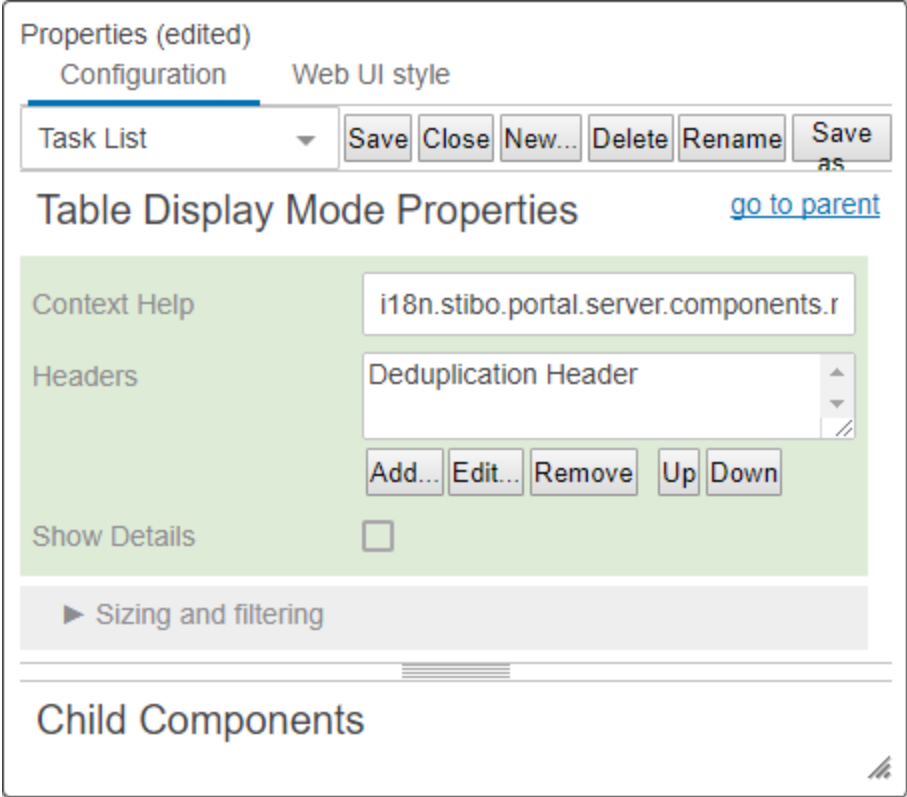
7. Deduplication Navigate To Handler Properties screen will be displayed.
8. In the **Matching Algorithm ID** field, specify the ID of the relevant matching algorithm.
9. In **State ID**, enter the review state of the clerical review workflow, and then click **Save**.

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.



- 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.

Properties (edited)

Configuration Web UI style

Task List Save Close New... Delete Rename Save as

Node List Properties [go to parent](#)

Hide Standard Buttons

ID*

Include Labels

Lookup Screen Type For Navigation

Page Size

Use Details Overlay

Child Components

Display Modes

Add.. Remove Up Down

Actions

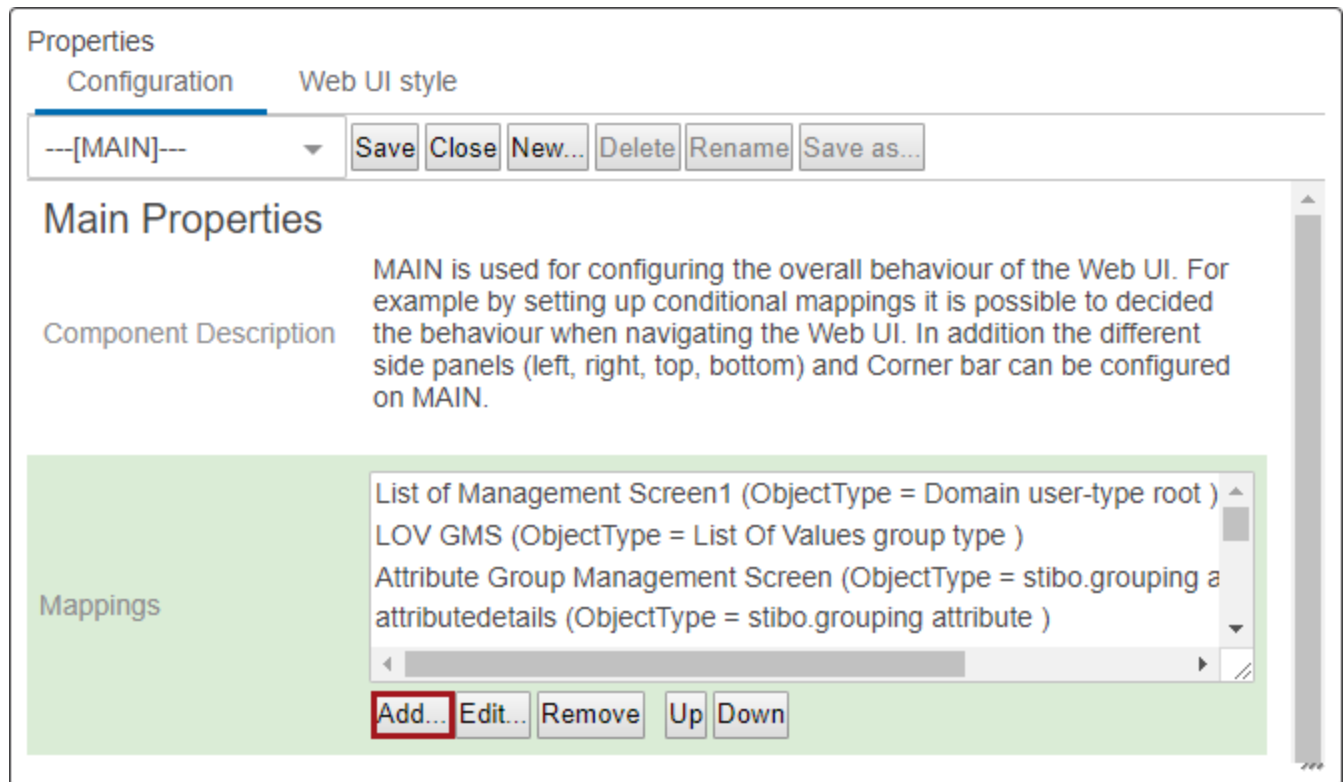
Add.. Remove Up Down

7. Click **Save**.

For information about individual table properties, see '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**.



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.

Workflow Condition Properties

Component Description A condition that is true if the node is in the specified STEP Workflow state and optionally flagged

*Workflow Details**

ClericalWorkflow2 ▼

Review ▼

<Select a status flag> ▼

✓ Add
✕ Cancel

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 see where each inherited value originated. A matching algorithm and corresponding action button(s) are required, and users can customize the table formatting.

Item Category Details

Basic Overview **Record**

Select all
 Clear filter
 Navigate to merge nodes screen
 Unlink duplicates
 Unlink single record from golden

	<input type="checkbox"/> I EI00001a	<input type="checkbox"/> (ER-179131)	<input type="checkbox"/> (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 / Web UI Setup and User Guide** documentation.

Configuration

Configure this component on a node details screen.

Properties

Configuration Web UI style

Node details Save Close New... Delete Rename Save as...

Golden Record Linked Members Properties [go to parent](#)

Component Description A component for displaying a tab with a golden record and its member records

Headers

ID Header (true)
Name Header
Attribute Value Group Header (false / false / fa

Add... Edit... Remove Up Down

*** Matching Algorithm** Matchingalgo ...

Property Direction HORIZONTAL

Show Group Headers

Show Only Valid Attributes

Use Immediate Save

Advanced

Dimensions Compare Display Mode Dimensions Edit...

Enable Freeze Panes

Child Components

Actions

Merge Confirmed Match From Grid Action
Unlink Duplicate From Grid Action
Unlink Single Record From Golden Action

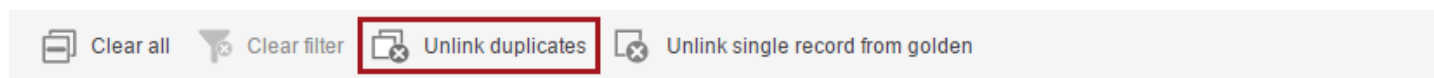
Add.. Remove Up Down

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.



	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	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.

	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	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)
<input checked="" type="checkbox"/> (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, see the **Potential Duplicates List** topic.

For more information about how to merge confirmed matches via workbench, see the **Match and Link in Workbench** topic.

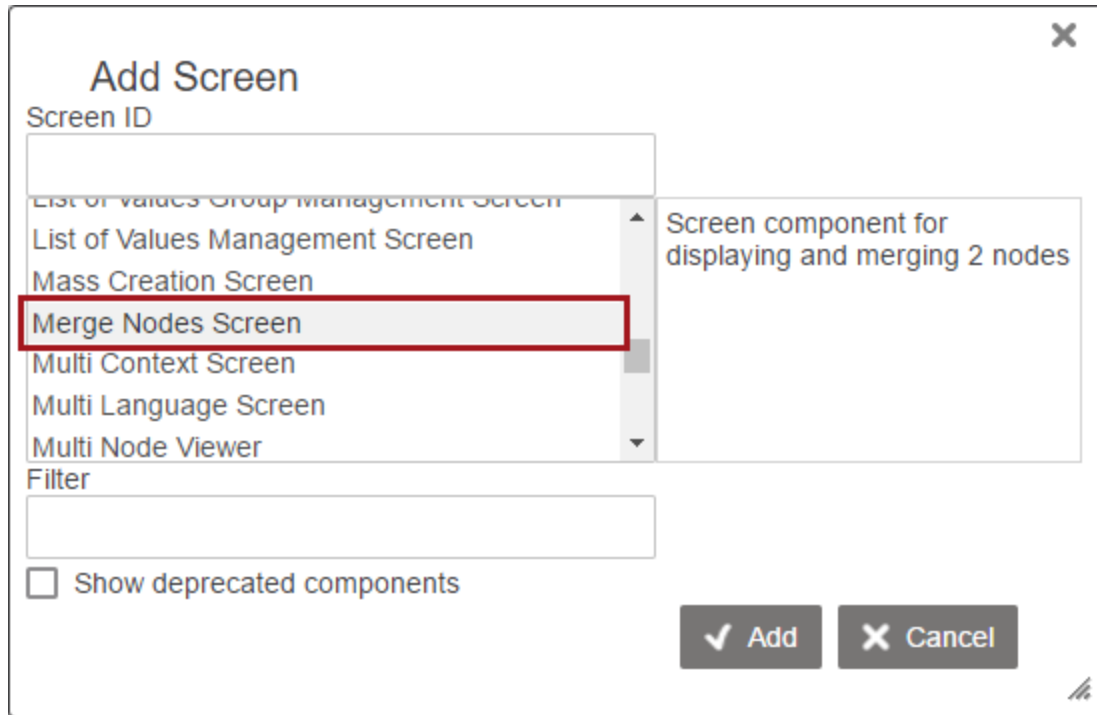
Note: Although the Merge Nodes Screen can be used to merge any nodes of the same super type, it is intended for use with matching algorithms and the Match Action for Identify Duplicates in conjunction with the Merge Confirmed Match From Grid action in the Confirmed Matches component.

Configuration

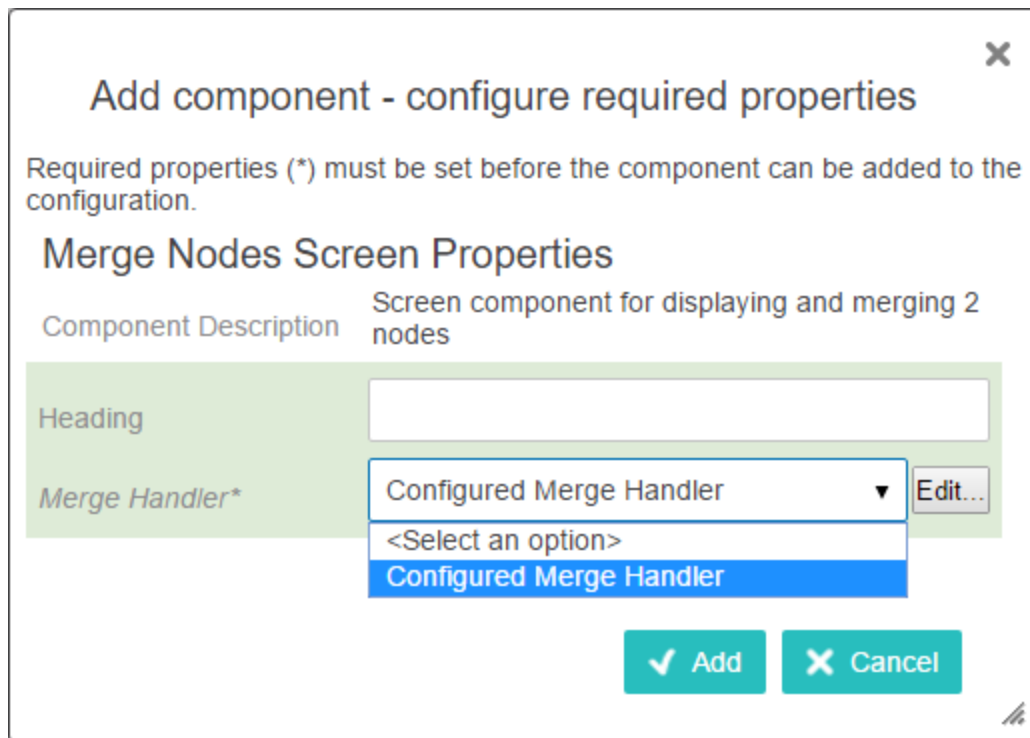
The following setup is required to enable the merge confirmed matches functionality in Web UI.

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**.



3. In the 'Merge Nodes Screen Properties' dialog, from the 'Merge Handler' dropdown select 'Configured Merge Handler' and click the **Add** button.



4. Click the **Save** button.

Map Merge Node Screen

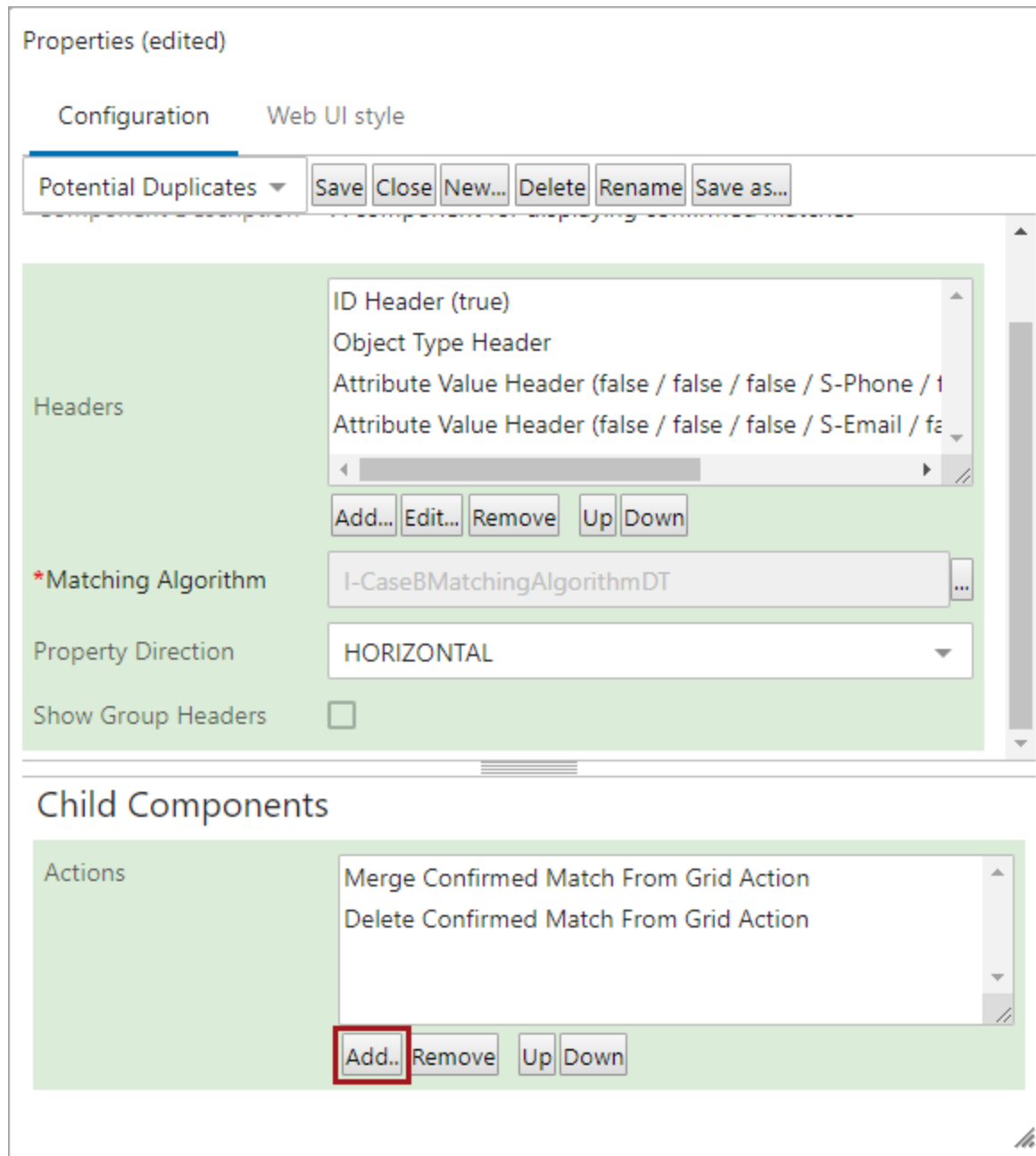
1. In design mode, navigate to [MAIN].
2. Under 'Mappings', click **Add**.
3. On the Screen Mapping dialog, under 'Conditions', click **Add**, select 'Merge Duplicate Condition', and click **Add**.
4. From the 'Screen' dropdown, select the merge nodes screen configured in the previous section.
5. Click the **Add** button.

6. Adjust the priority of the screen as needed. For more information, see the **Mappings** topic in the **Web User Interfaces / Web UI Setup and User Guide** 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**.



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 duplicate from grid' button.

Subscriber Details

Basic Information and References
Potential Duplicates List
Confirmed Matches
Confirmed Non Matches

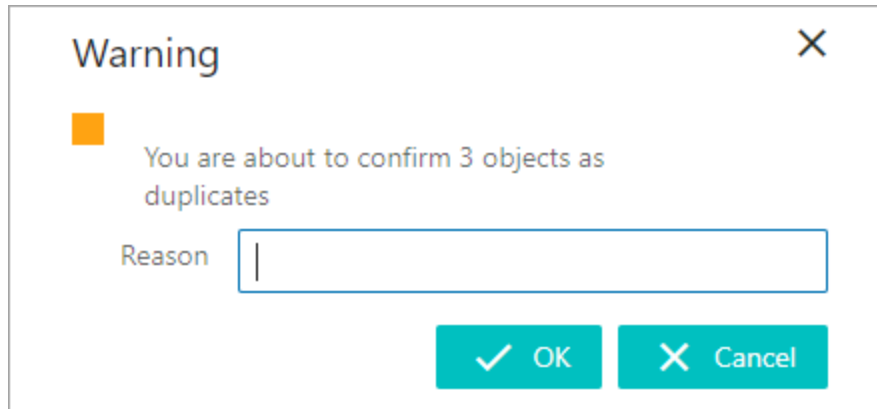
Clear all
 Hide Equal
 Mark Different
 Confirm duplicate from grid
 Reject duplicate from grid

	Score	Matching Algorithm	Name
<input checked="" type="checkbox"/> Anthony C	-	-	Anthony C
<input type="checkbox"/> Tony Cooley	89.206	I Case B Matching Algorithm DT	Tony Cooley
<input type="checkbox"/> Anthony Cooley	89.206	I Case B Matching Algorithm DT	Anthony Cooley

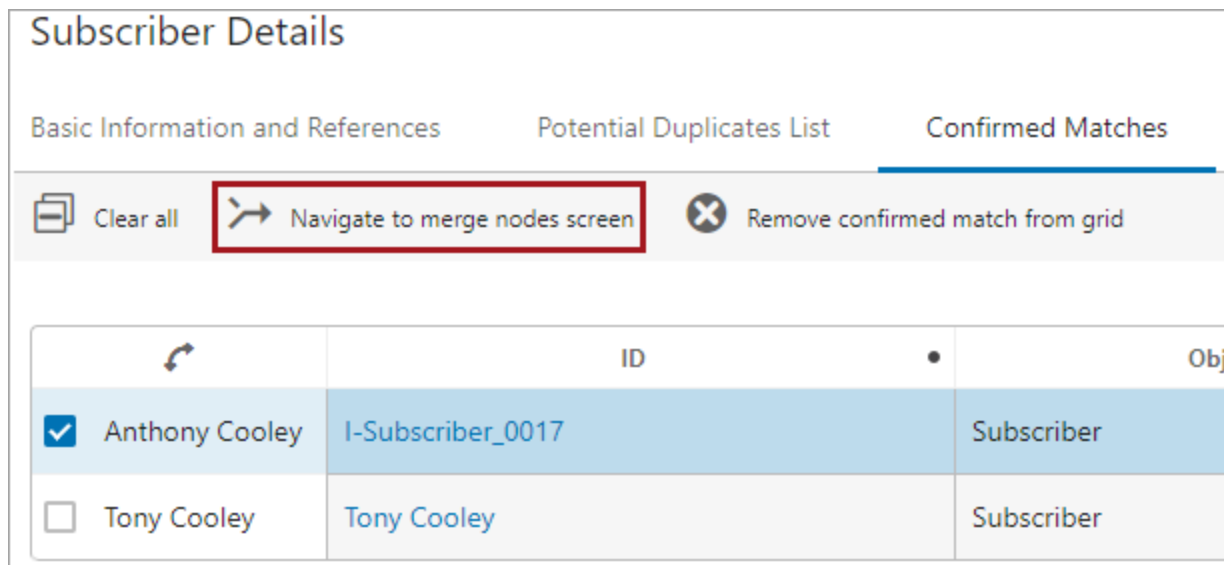
Number of items : 3

Save

3. Explain why the objects are duplicates and click **OK**.



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.



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)

Merge

- 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.

 Clear all
 Hide Equal
 Mark Different
 Confirm duplicate from grid
 Reject duplicate from grid

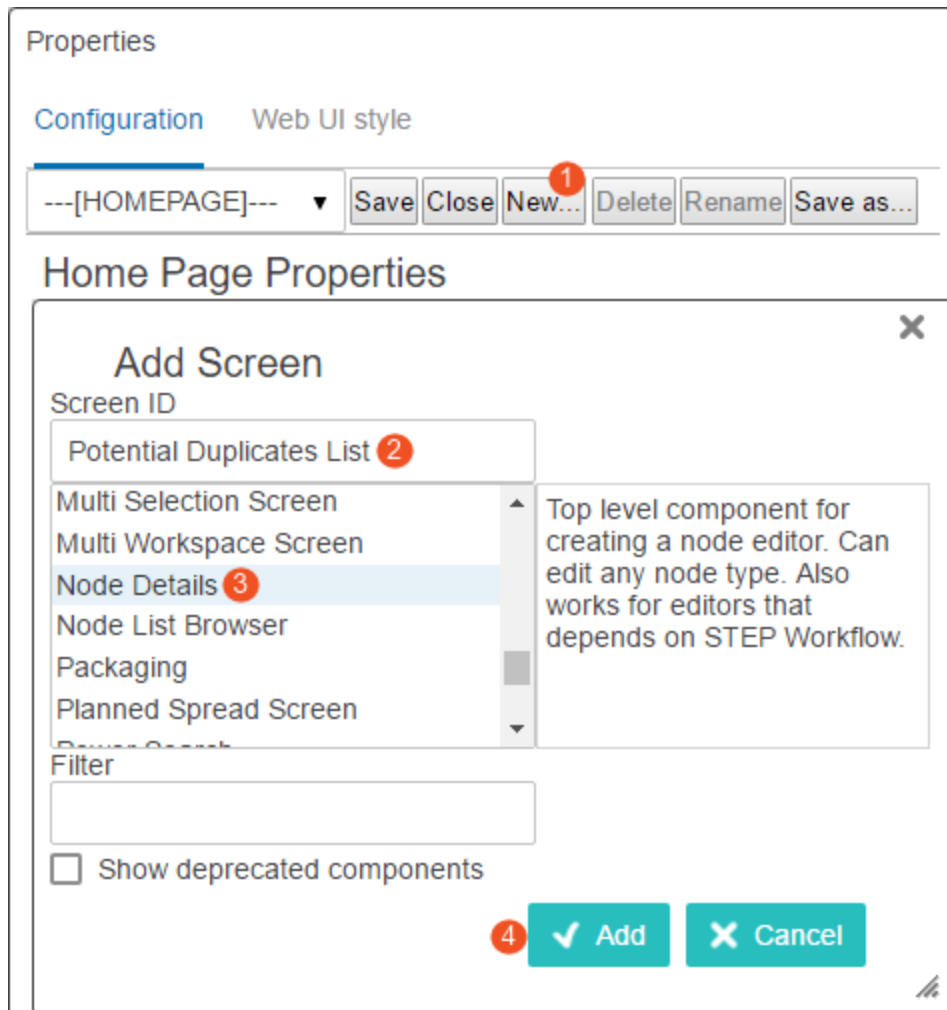
	Score	Matching Algorithm	ID	Object Type
<input checked="" type="checkbox"/> Miles Morales	-	-	CUS_265382	CD_Customer

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, see the **Match and Link** documentation.

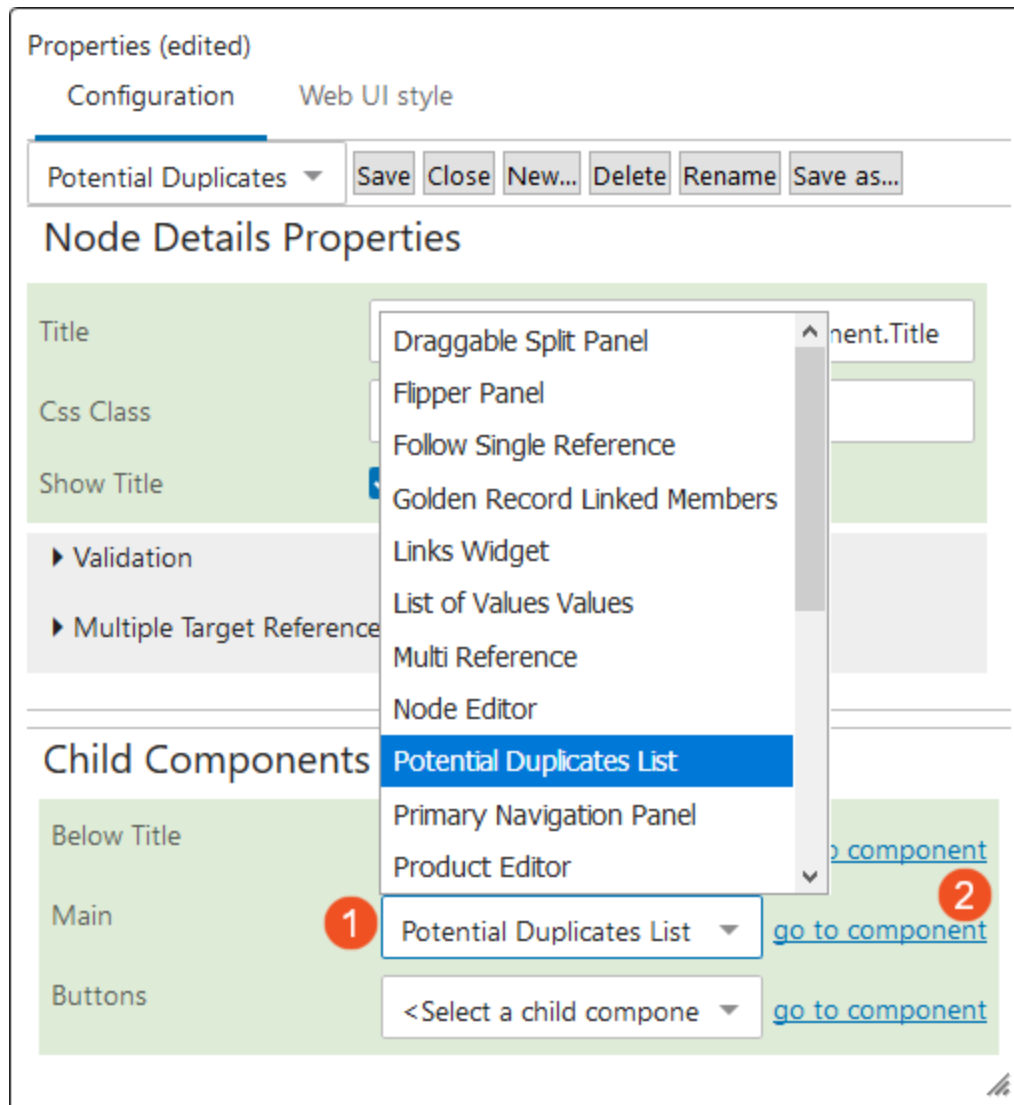
Configuring the Deduplication Table

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**.



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.



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.

Properties (edited)

Configuration Web UI style

Potential Duplicates ▾ Save Close New... Delete Rename Save as...

Potential Duplicates List Properties [go to parent](#)

Component Description A component for displaying a tab with a list of possible duplicates listview

Dimensions <Select an option> Edit...

Event

Headers Name Header
Object Type Header
Attribute Value Header (false / false / false ,
Attribute Value Header (false / false / false ,
Add... Edit... Remove Up Down

Hide Selection Buttons

Matching Algorithm PersonMatchAlgorithm ... Clear

Property Direction <Select a value>

Show Group Headers

Child Components

Actions Confirm Duplicate From Grid Action
Reject Duplicate From Grid Action
Add.. Remove Up Down

7. In the Child Components section, for the Actions parameter, add the Confirm and Reject actions.

The 'Hide Equal' and 'Mark Different' actions automatically display before other actions configured manually. For more information, see the **Comparing Data Using Hide Equal and Mark Different** section of the **Web User Interfaces / Web UI Setup and User Guide** documentation.

8. 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, see the **Mappings** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.
9. Save and close the Web UI Design Mode.

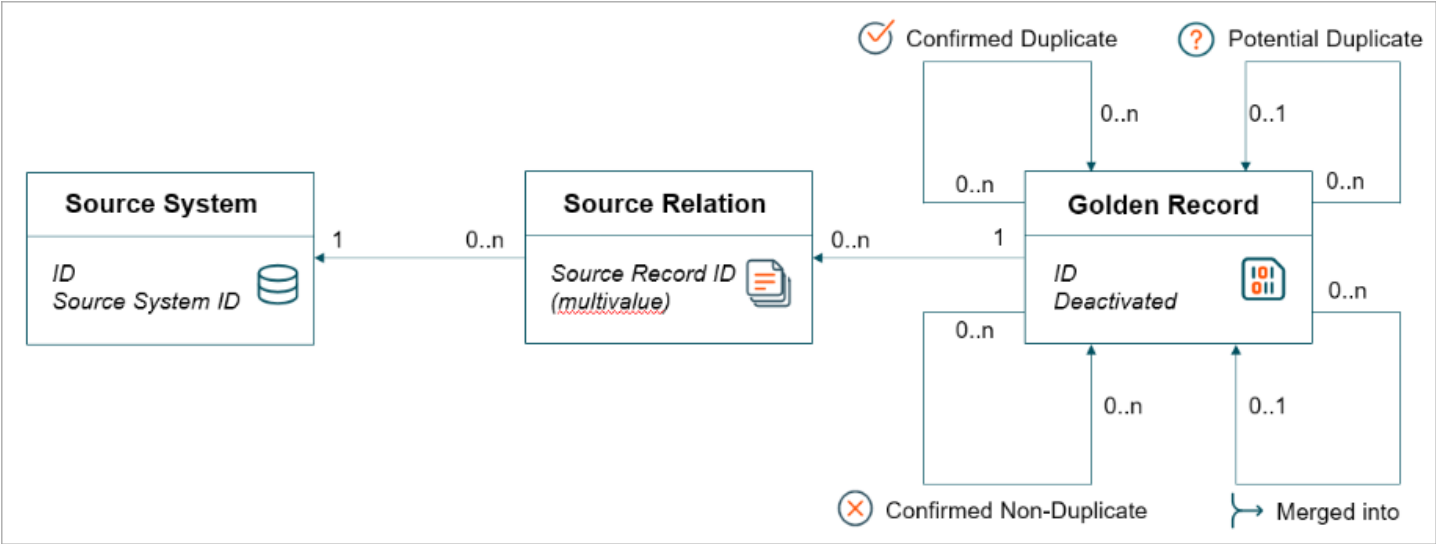
Match and Merge

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, see 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

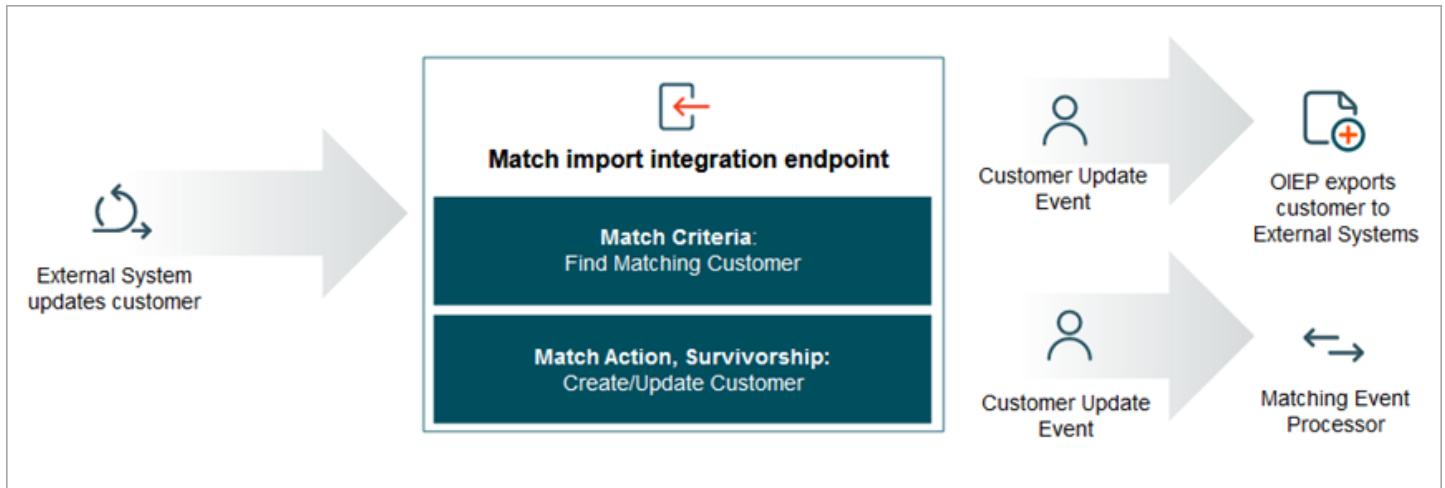


Unlike the Match and Link Match Action, in the Match and Merge Match Action the source record and golden record do not use separate object types. The source system is registered as an entity and the source relation is modeled as a reference from the golden record to that source system.

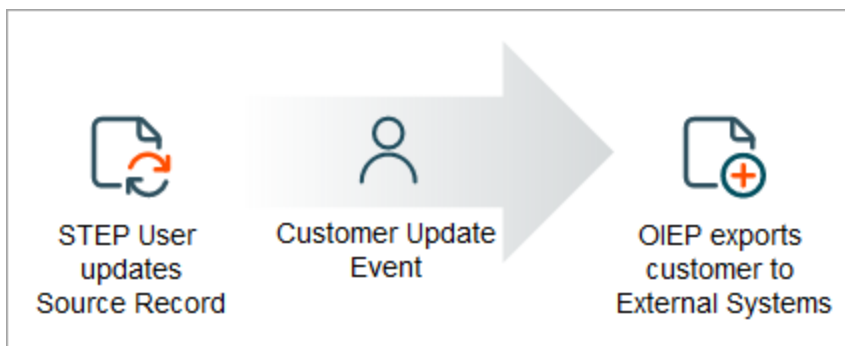
Information Flow

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.

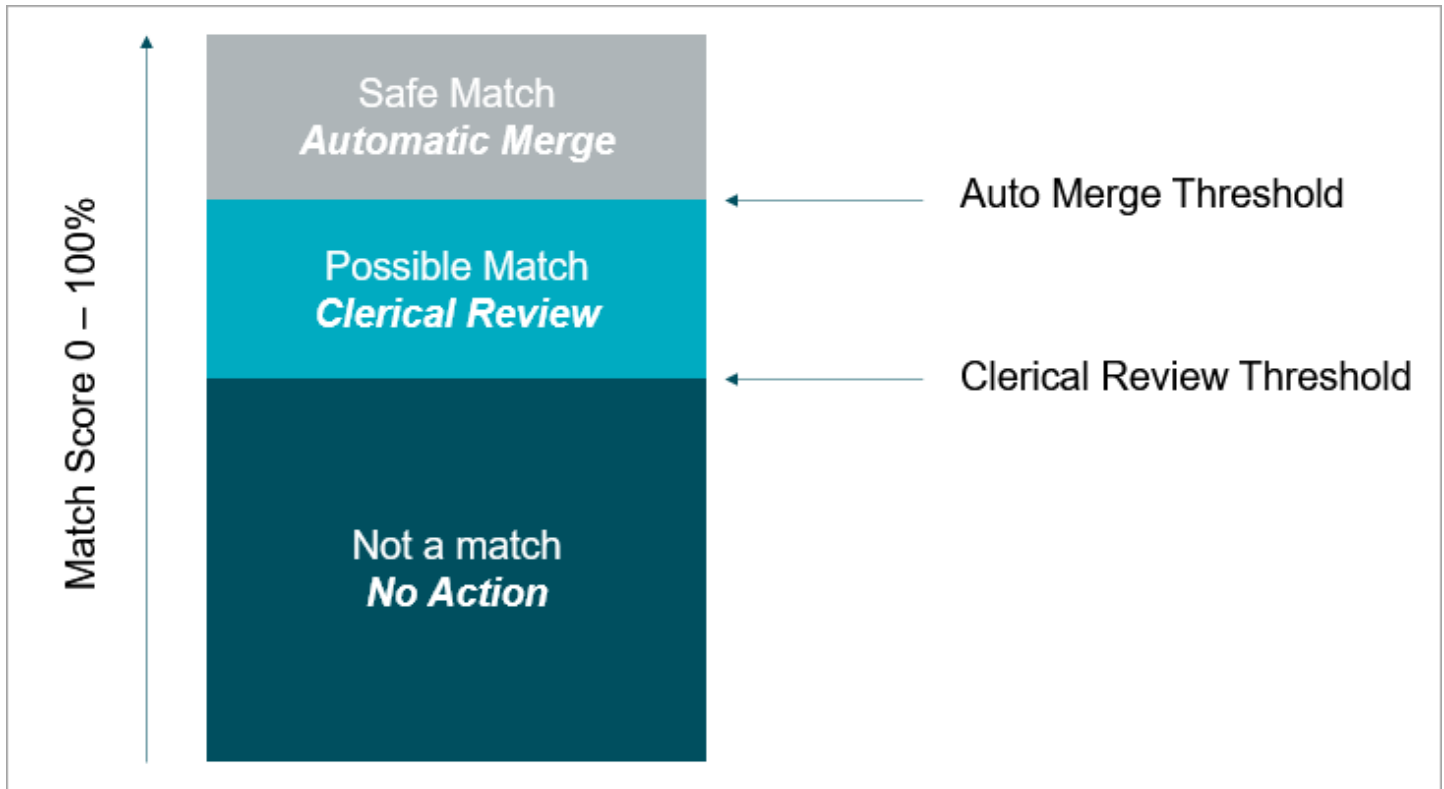
In both cases, the incoming source record is matched against the existing golden records, and if a match is found, the information from the source record is merged into the relevant golden record using survivorship rules. If this results in updated information, the customer record can be exported back 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.



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.

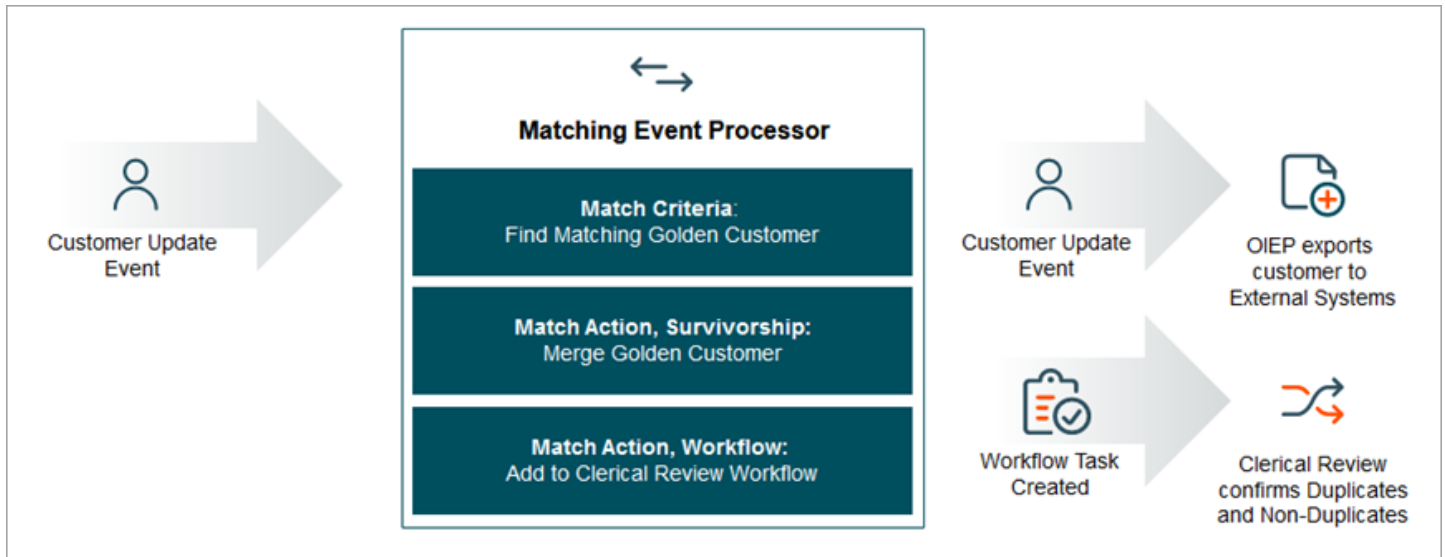


The matching process uses a 'match score' within three groups separated by thresholds to indicate the likelihood of a match.

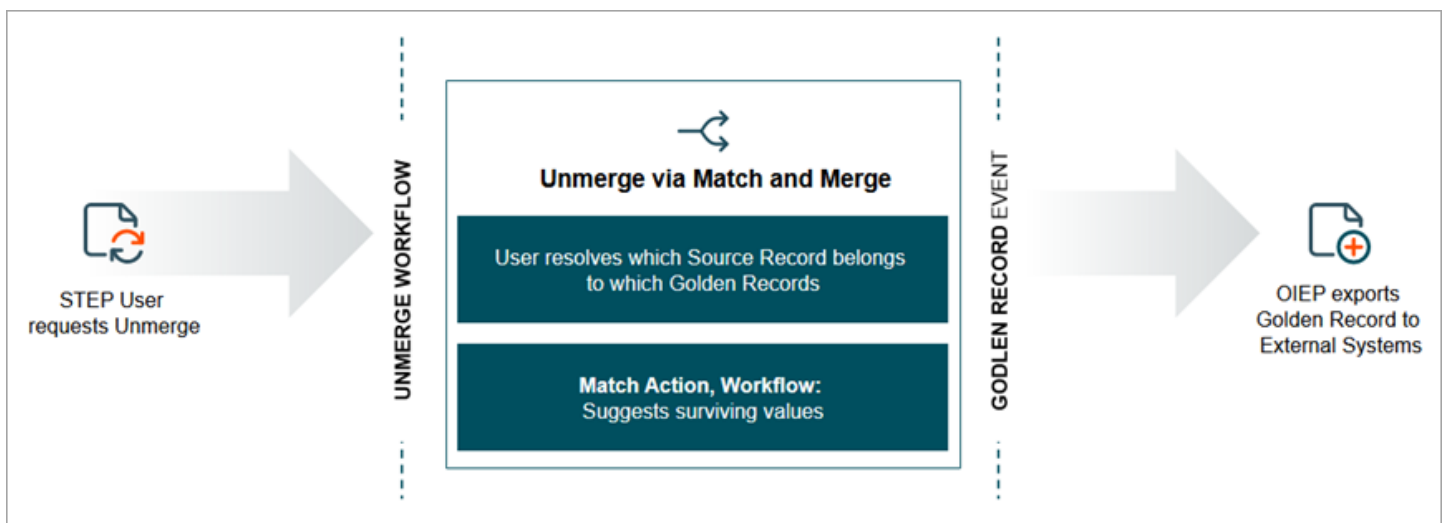


- A match score above the auto merge threshold (the highest threshold) is considered a match and the system automatically merges the data. During import, this results in the incoming data being merged directly into the existing golden record. If updates make two existing records match above the auto 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.



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.



Match and Merge Traceability

This functionality is used by a Match and Merge solution. For more information, see 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	<table border="1"> <thead> <tr> <th>Source Record</th> <th>Source System</th> <th>Created</th> <th>Last Updated</th> </tr> </thead> <tbody> <tr> <td>16320807-2367</td> <td>CRM Global</td> <td>10/14/2021</td> <td>10/14/2021</td> </tr> </tbody> </table>	Source Record	Source System	Created	Last Updated	16320807-2367	CRM Global	10/14/2021	10/14/2021
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)								

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, see the **Node Details Screen** topic of the **Web User Interfaces / Web UI Setup and User Guide** 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 <small>INDIVIDUAL CUSTOMER • ID: 248854</small>					
Overview Source Traceability History Household Confirmed Non Matches Household Deduplication					
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, see 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.

1 Oliver Johnson rev.7.0 - Status						
Individual Customer Data Containers References Referenced By Matching Data Profile Proof View Status State Log Tasks						
Revisions						
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=CustomerGR.378499	
> 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=CustomerGR.378497	
> 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=CustomerGR.378499	
> 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.

1 Olivia Johnson rev.3.0 - Status						
Individual Customer Data Containers References Referenced By Matching Data Profile Proof View Status State Log Tasks						
Revisions						
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=CustomerGR.378495	
> 2.0	Mon May 04 13:43:37 EDT 2020	Mon May 04 13:43:37 EDT 2020	X	USER	Source ::Merged into=CustomerGR.378495	
> 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, see 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, see 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 / Super User Guide** 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.

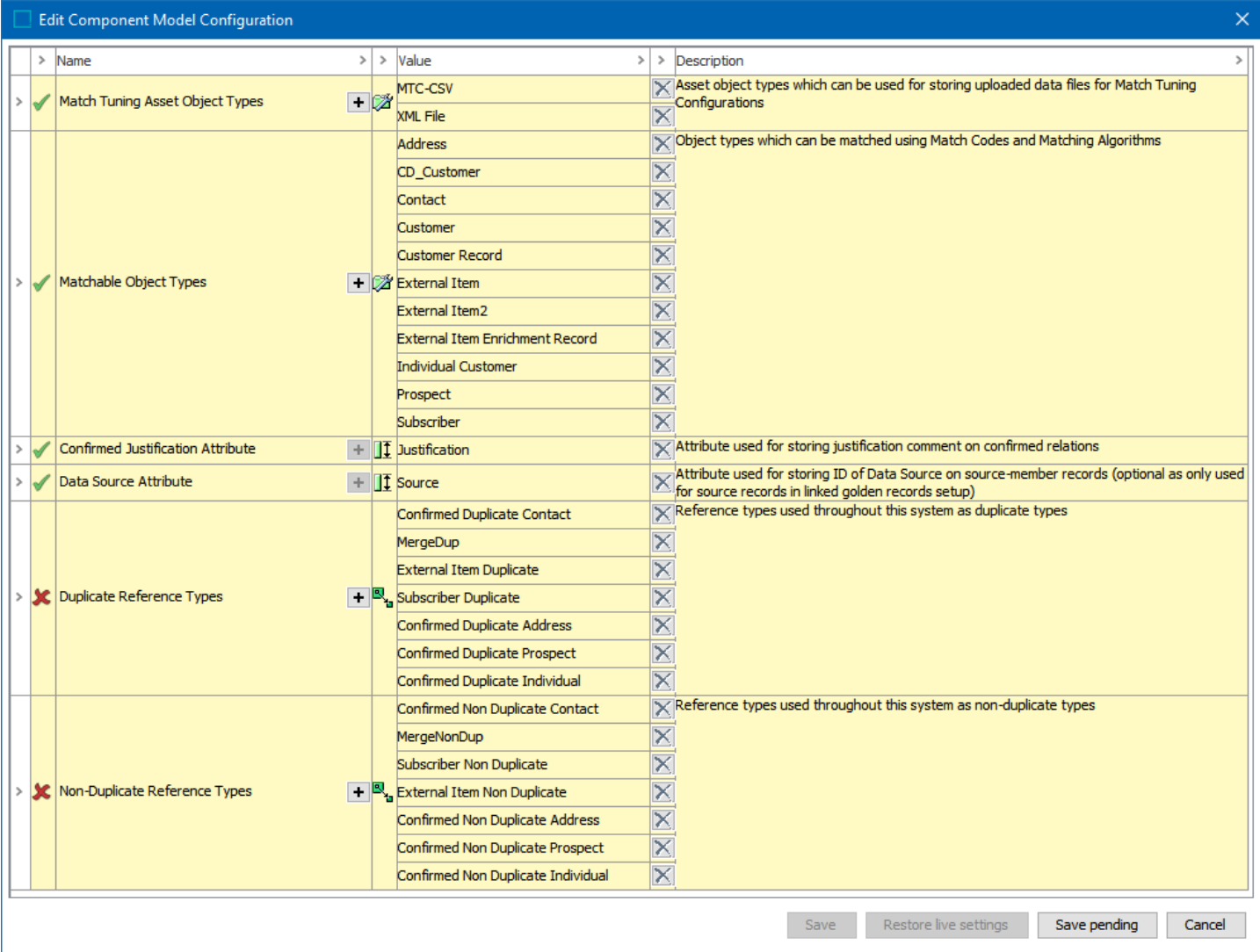
Configuration

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.

Name	Value	Description
Match Tuning Asset Object Types		
	MTC-CSV	Asset object types which can be used for storing uploaded data files for Match Tuning Configurations
	XML File	
Matchable Object Types		
	Address	Object types which can be matched using Match Codes and Matching Algorithms
	CD_Customer	
	Contact	
	Customer	
	Customer Record	
	External Item	
	External Item2	
	External Item Enrichment Record	
	GR	
	Individual Customer	
	Prospect	
	Subscriber	
Confirmed Justification Attribute		
	Justification	Attribute used for storing justification comment on confirmed relations
Data Source Attribute		
	Source	Attribute used for storing ID of Data Source on source-member records (optional as only used for source records in linked golden records setup)
Duplicate Reference Types		
	Confirmed Duplicate Contact	Reference types used throughout this system as duplicate types
	MergeDup	
	External Item Duplicate	
	Subscriber Duplicate	
	Confirmed Duplicate Address	
	Confirmed Duplicate Prospect	
	Confirmed Duplicate Individual	
Non-Duplicate Reference Types		
	Confirmed Non Duplicate Contact	Reference types used throughout this system as non-duplicate types
	MergeNonDup	
	Subscriber Non Duplicate	
	External Item Non Duplicate	
	Confirmed Non Duplicate Address	
	Confirmed Non Duplicate Prospect	
	Confirmed Non Duplicate Individual	

2. Click the **Edit** link (or the **Edit (pending changes)** link) to display the Edit Component Model Configuration dialog.



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 (X) 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.
 - **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.

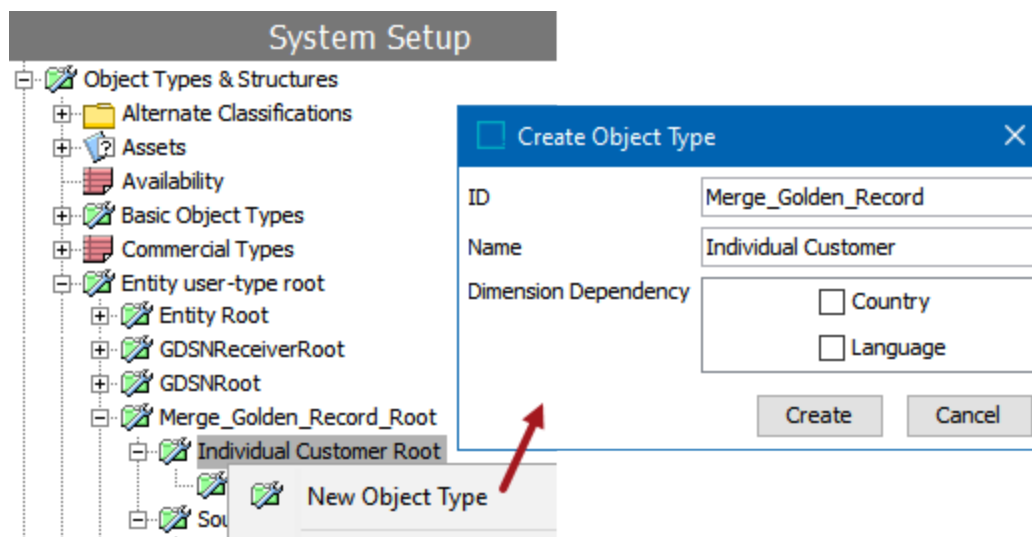
Configuring the Merge Golden Record Object Type

This functionality is used by a Match and Merge solution. For more information, see 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.



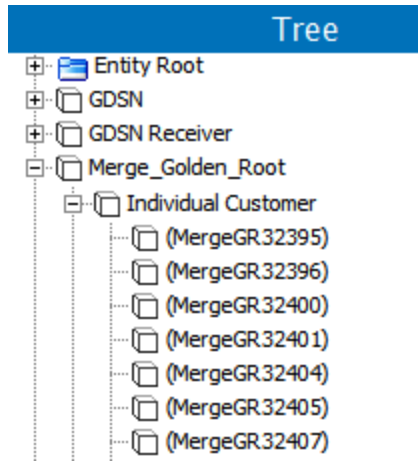
2. On the Description flipper, set the **ID Pattern** parameter to use the **[id]** variable. See the **Autogenerate Using Name Pattern and ID Pattern** topic in the **System Setup / Super User Guide** documentation.

Individual Customer - Object Type	
Object Type	References Log
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	
Reference Type	Validity Log
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.



Configuring the Matching - Merge Golden Record Component Model

This functionality is used by a Match and Merge solution. For more information, see 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: <ul style="list-style-type: none"> ID=true, value=Yes ID=false, value=No It is not recommended to reuse this LOV.	None	No	No
Potential Duplicate Match Algorithm ID	Yes	Text	None	No	Yes
Source	No	Text (40-character limit)	None	No	Yes

Attributes	Externally Maintained	Validation Base Type	Dimension Dependencies	Mandatory	Multivalued
Record ID Attribute					
Source System ID Attribute	No	Text	None	No	No

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	Yes	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-From Relation Reference Types	No	None	No	No	None	Merge Golden Records object types that should support unmerge	Same as the source types
Unmerge Reference Type	No	None	No	No	None	Merge Golden Records	Same as the source

Reference Types	Externally Maintained	Dimension Dependencies	Allow Multiple References	Mandatory	Inheritance	Valid Source Types	Valid Target Types
						object types that should support unmerge	types

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.

Name	Value	Description
	Customer Contact	Object types which can be used as merged golden records in Matching Algorithms
	Individual Customer	
	Organization Customer	
	Prospect	
> Golden Record Object Types	Supplier	
	Customer Contact	Golden Record Object Types for which Source Records will be stored when importing with Match and Merge Importer. This is an optional setting.
> Keep Source Records for Golden Record Object T...	Individual Customer	
	Organization Customer	
> Source System Object Type	Source System	Object type which can be used as source system for merged golden records
> Deactivated Attribute	Deactivated Record	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 Algorithm ID	PotentialDuplicateMatchAlgorithmID	Multi-valued attribute for storing matching algorithm IDs 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	Merged Into	Single valued reference types for linking a deactivated golden record to the surviving golden record when merging golden records
> Potential Duplicate Reference Type	PotentialDuplicate	Optional Reference type for relations between a potential duplicates in a clerical review task
> Source Relation Reference Type	CustomerSourceSystem	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, see 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 Algorithm ID** – Optional, In-Memory is required. Select the attribute to indicate which matching algorithm(s) identify 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, see 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, see 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

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.

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 entity. 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.

Match Criteria Tab

The Match Criteria tab defines how to compare two objects and evaluate to what degree they are similar.

Important: Create new match algorithms with embedded match codes as defined in the **Configuring Matching Algorithms** topic.

The Match Criteria Tab is separated into the following flippers:

- **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.
- **Matchers** do the actual comparisons of values. A matcher compares one logical aspect of the objects, assigning a equality percentage to that aspect based on the related values.
- **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.
- **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. For information on choosing match codes, see the **Selecting Match Codes** topic.
- **Match Code Filter** allows users to remove specific match code values based on a Transformation Lookup Table.

- **Evaluator** is the user's test tool and allows a user to execute the entire match criteria setup on two select records.

The screenshot displays the 'Individual Matching Tmp - Match Criteria' configuration window. It includes several sections:

- Data Elements:** Lists fields like normName, normAddress, normEmail, and normPhone with their respective normalizers.
- Matchers:** Lists matchers for name, address, email, and phone.
- Rules:** Shows three rules with weights of 70 and formulas like $(address * 30.0 + name * 30.0) / 60.0$.
- Match Code Generators:** Lists generators for emailMatchCode, phoneMatchCode, and nameAndAddress.
- Match Code Filter:** A section for filtering match codes.
- Evaluator:** The active section showing a comparison between two nodes. It includes a 'Matchers' table, a 'Common Match Codes: Yes' section, and a 'Match Code Generators' table with columns for 'First Node Result' and 'Second Node Result'.

Legacy Match Criteria Without Embedded Match Codes

For match algorithms without embedded match codes, see the **Match Codes** topic.

A matching algorithm with the Match Criteria flipper displayed indicates it is a legacy algorithm where the match codes are not embedded and must be created manually.

Use the following steps to configure:

1. Click the **Add Criterion** link to display the 'Select Match Criterion' dialog.
2. Specify a **Name**.
3. Choose a match criterion from the **Select Match Criterion** dropdown.
4. Click the **Add** button.
5. Click into the **Criterion** field and then click the ellipsis button (...) to open the editor.
6. Create the matching criterion and click **OK**.
7. Click into the **Weight** field and specify a weight for the criterion.

Match Criteria		
Name	Criterion	Weight
> DT	Decision Table: Sub Tables 0, Expressions 17, Rules 4	10.0
Add Criterion		

Match code generators are only available when match codes are not embedded into the matching algorithm.

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. It is not recommended to 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.

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.

Golden Record Clerical Review Task List						
ID	Name	Source Information	First	Last	Email	Phone Number
<input type="checkbox"/>	CustomerGR229245	Customer003	SAP	SAP_003	Theresa Lebowitz	theribo@email.com (615)497-5547
<input type="checkbox"/>	CustomerGR229247	Customer001	SAP	SAP_001	Theresa Lebowitz	tlebo@email.com (615)497-8898
<input checked="" type="checkbox"/>	CustomerGR229243	Customer004	SAP	SAP_004	Theresa Lebowitz	tlebowitz@email.com (615)497-3333
<input type="checkbox"/>	CustomerGR229244	Customer002	SAP	SAP_002	Theresa Lebowitz	tleebu@email.com (615)497-4138
<input type="checkbox"/>	CustomerGR229246	Customer005	SAP	SAP_005	Theresa Lebowitz	theresalebo@email.com (615)497-0121
<input type="checkbox"/>	MergeGR32407	(MergeGR32407)	SAP	GARED	FULLEN HACKETT	maureen.eizt.2015@gm... (418)687-8954
<input type="checkbox"/>	MergeGR32410	(MergeGR32410)	SAP	GARED	FULLEN HACKETT	breanta.alsip.ct184@gma... (229)490-7378

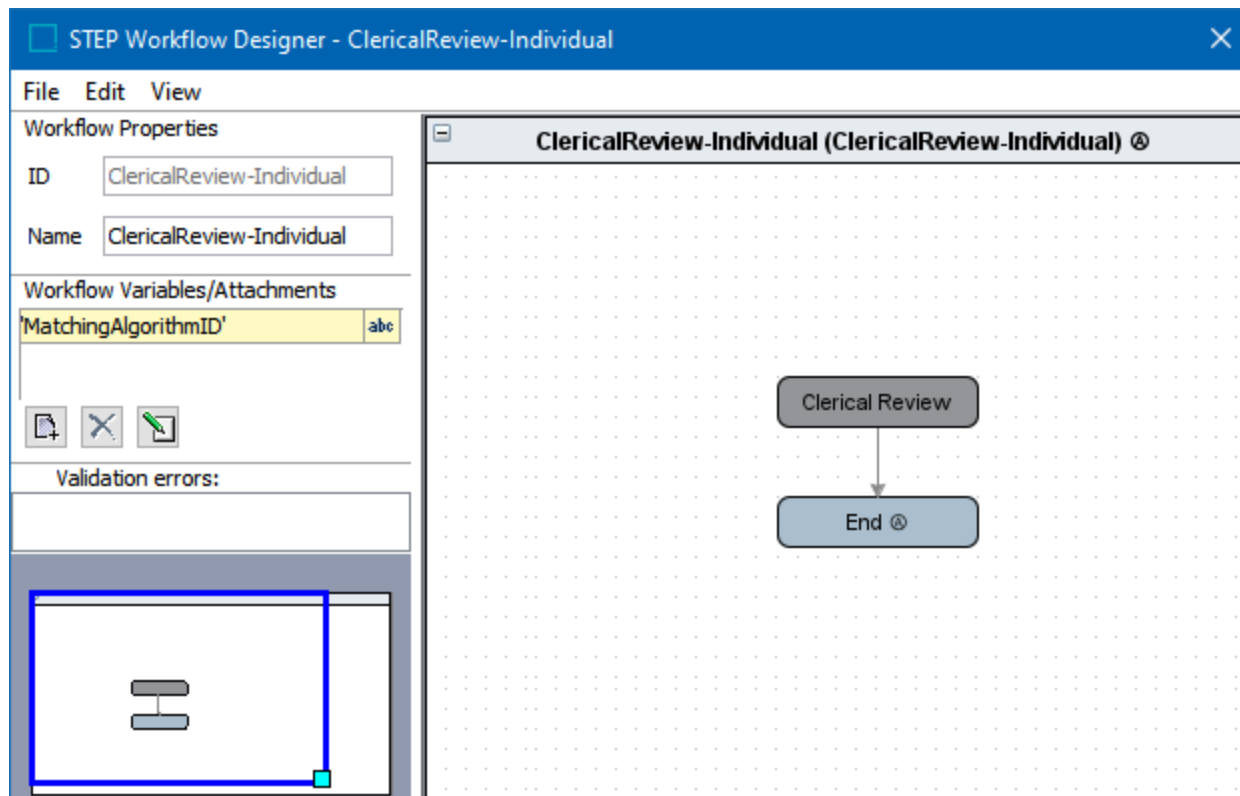
For information on setting up and using the golden record clerical review screen as well as the advanced merge feature, see the **Golden Record Clerical Review Task List** topic of the **Web User Interfaces / Web UI Setup and User Guide** documentation.

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, see 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.

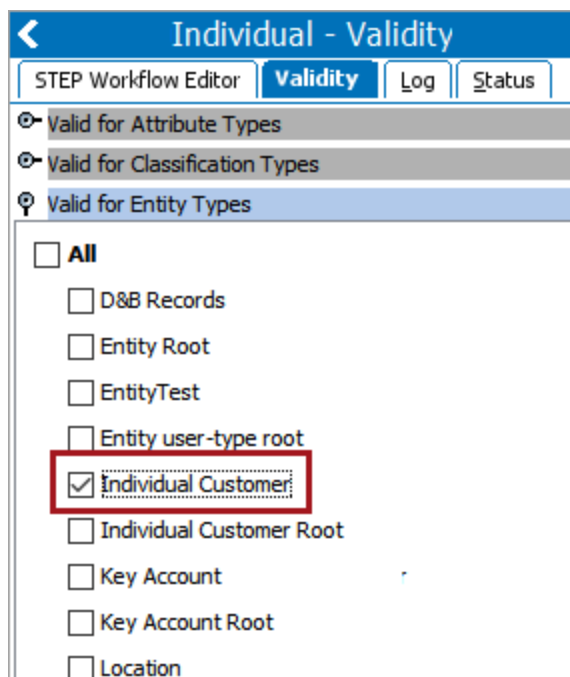


The following elements are available for the clerical review workflow:

- **Workflow** - (required) create a workflow as simple or elaborate as needed. For more information, see 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, see 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, see the **Creating a Business Rule or Library** topic in the **Business Rules** documentation.
- On the Validity tab of the workflow, select the merge object type.



Considerations

The following rules apply when using a clerical review workflow to configure the Merge Golden Record match action:

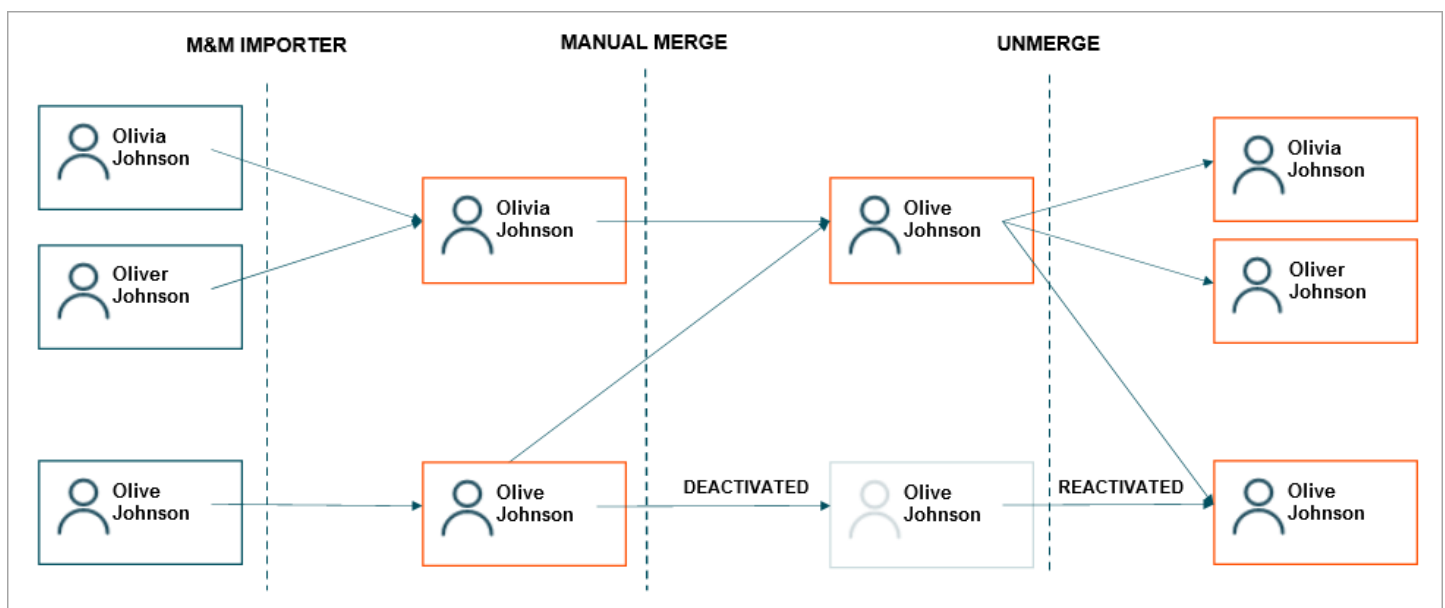
- If a status flag is configured, but a business condition is not configured, the status flag behaves as if a business condition evaluated to true.
- If a business condition is configured and a status flag is not configured, the business condition is ignored.
- Although the business condition runs as 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.

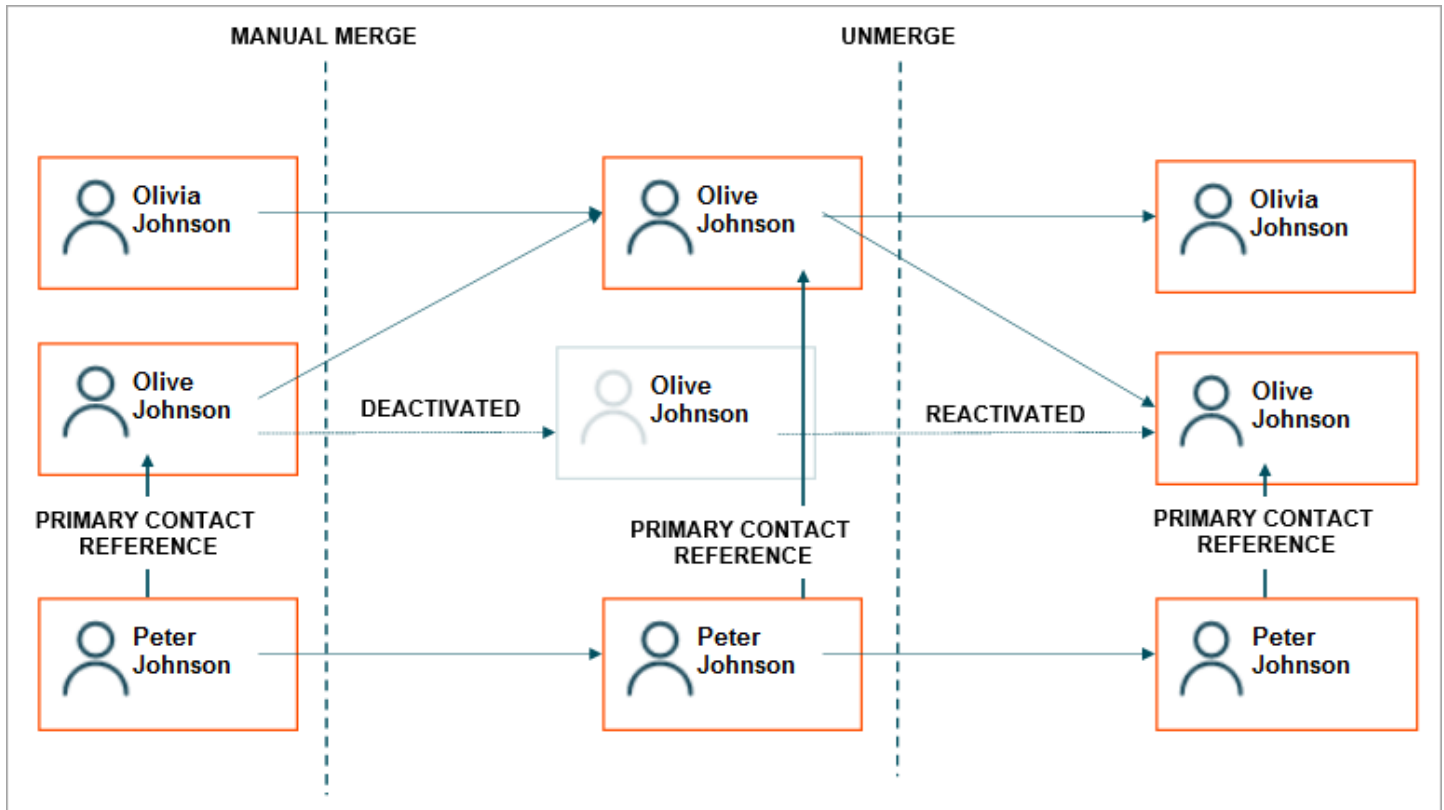
This functionality is available to incoming records with source record IDs as well as deactivated golden records. The unmerge operation restores relevant data back to the golden record and can be done as part of a workflow or as an ad hoc operation. The logic to revert the changes is not supported on 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, see 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, see 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, see **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, see the **Unmerging Golden Records** topic.

For information on an unmerge workflow, see 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, see 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, see the **Match and Merge Clerical Review - Unmerge** topic.

The Unmerge wizard in Web UI (shown below) provides a collaborative process for all unmerge operations.

Unmerge: Jeff Collins ID: 35005 1 Distribute Source Records 2 Select Surviving Values

↶ Reset all
→ Move to
🗑️ Reactivate Golden Record

	Original Golden Record 35005	New Golden Record																												
Sources ^	<input type="checkbox"/> SAP London - 8518 <input type="checkbox"/> SAP US - 2462 <input type="checkbox"/> Deactivated Golden Record 63003 <input type="checkbox"/> Dynamics Europe - 4323	Select a source record to move this new golden record.																												
Surviving Values ^	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">Name</td> <td>Jeff Collins</td> <td style="text-align: right;">2 unused</td> </tr> <tr> <td>First Name</td> <td>J.</td> <td style="text-align: right;">2 unused</td> </tr> <tr> <td>Last Name</td> <td>Collins</td> <td></td> </tr> <tr> <td>Credibility Score</td> <td>6</td> <td style="text-align: right;">2 unused</td> </tr> <tr> <td>Main Address</td> <td>305th Ave Hadley, Massachusetts, 01035 USA</td> <td style="text-align: right;">7 unused</td> </tr> <tr> <td rowspan="3">Phone</td> <td>Business: 555-6412</td> <td style="text-align: right;">3 unused</td> </tr> <tr> <td>Private: 514-7258</td> <td style="text-align: right;">4 unused</td> </tr> <tr> <td>Other: 514-5416</td> <td></td> </tr> <tr> <td>Email</td> <td>jeff.collins@yahoo.com</td> <td style="text-align: right;">4 sources for Email</td> </tr> <tr> <td>Company Code Data</td> <td>MAG Germany</td> <td style="text-align: right;">1 unused 3 unused for Company Code Data</td> </tr> </table>	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	
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																												

Cancel Unmerge
Select Surviving Values

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. See the **Node Details Screen** topic of the **Web User Interfaces / Web UI Setup and User Guide** 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

Css Class


Show Title

▶ Validation

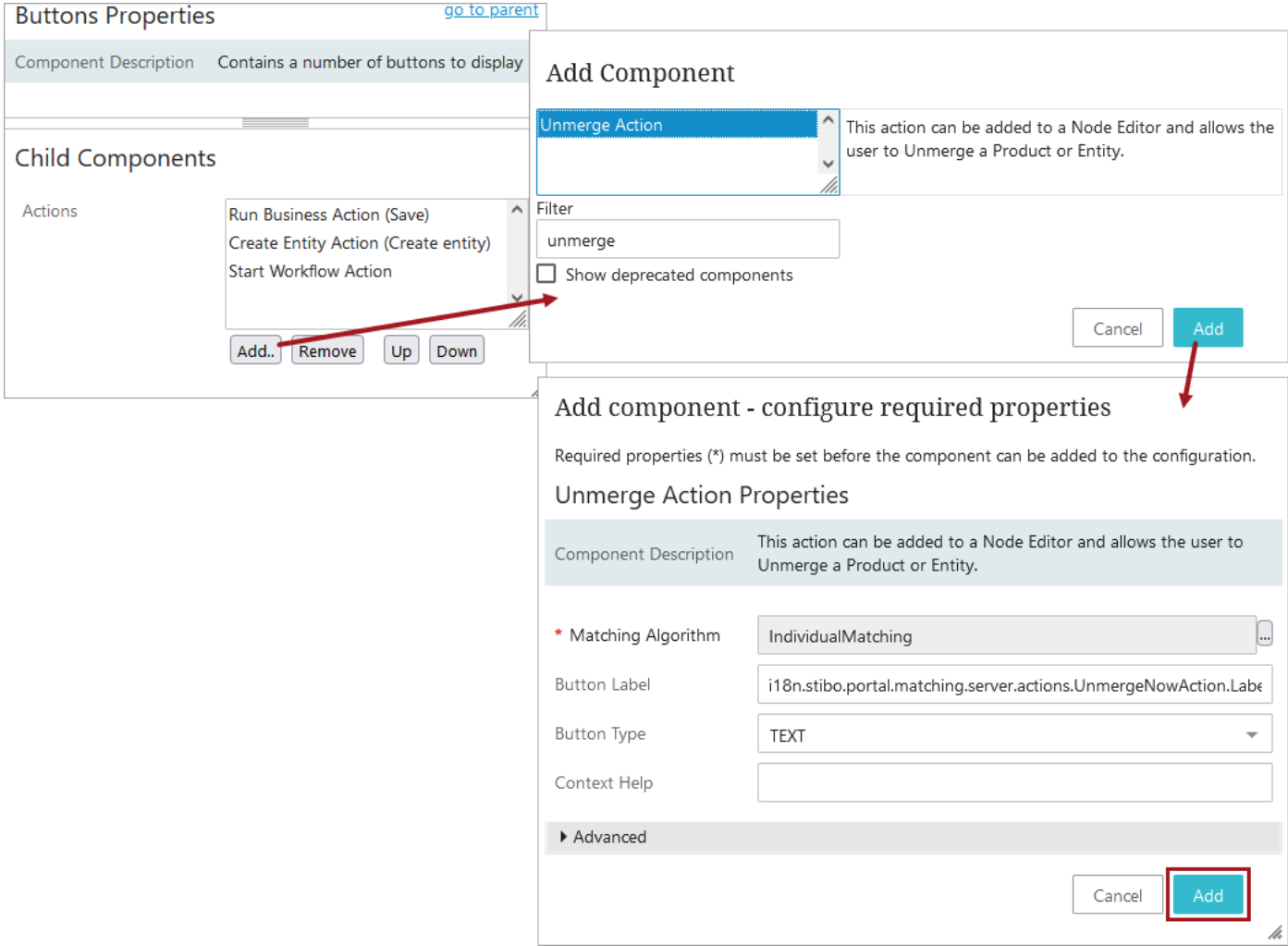
▶ Multiple Target References

Child Components

Below Title	<input style="width: 150px;" type="text" value="Entity Summary"/>	go to component
Main	<input style="width: 150px;" type="text" value="Tab Control"/>	go to component
Buttons	<input style="width: 150px;" type="text" value="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.



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, see 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, see 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.

Unmerge: Jeff Collins ID: 35005 **1** Distribute Source Records **2** Select Surviving Values

Reset all Move to Reactivate Golden Record

	Original Golden Record 35005	New Golden Record																																
Sources	<input type="checkbox"/> SAP London - 8518 <input type="checkbox"/> SAP US - 2462 <input type="checkbox"/> Deactivated Golden Record 63003 <input type="checkbox"/> Dynamics Europe - 4323	Select a source record to move this new golden record.																																
Surviving Values	<table border="1"> <tr> <td>Name</td> <td>Jeff Collins</td> <td>2 unused</td> </tr> <tr> <td>First Name</td> <td>J.</td> <td>2 unused</td> </tr> <tr> <td>Last Name</td> <td>Collins</td> <td></td> </tr> <tr> <td>Credibility Score</td> <td>6</td> <td>2 unused</td> </tr> <tr> <td>Main Address</td> <td>305th Ave Hadley, Massachusetts, 01035 USA</td> <td>7 unused</td> </tr> <tr> <td rowspan="3">Phone</td> <td>Business: 555-6412</td> <td>3 unused</td> </tr> <tr> <td>Private: 514-7258</td> <td>4 unused</td> </tr> <tr> <td>Other: 514-5416</td> <td></td> </tr> <tr> <td rowspan="2">Email</td> <td>jeff.collins@yahoo.com</td> <td></td> </tr> <tr> <td colspan="2">4 sources for Email</td> </tr> <tr> <td rowspan="2">Company Code Data</td> <td>MAG Germany</td> <td>1 unused</td> </tr> <tr> <td colspan="2">3 unused for Company Code Data</td> </tr> </table>	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		
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																																	

Cancel Unmerge Select Surviving Values

For the complete unmerge process, see the **Match and Merge Clerical Review - Unmerge** topic.

For details on unmerge in Web UI, see the **Configuring and Using Match and Merge Unmerge in Web UI** topic.

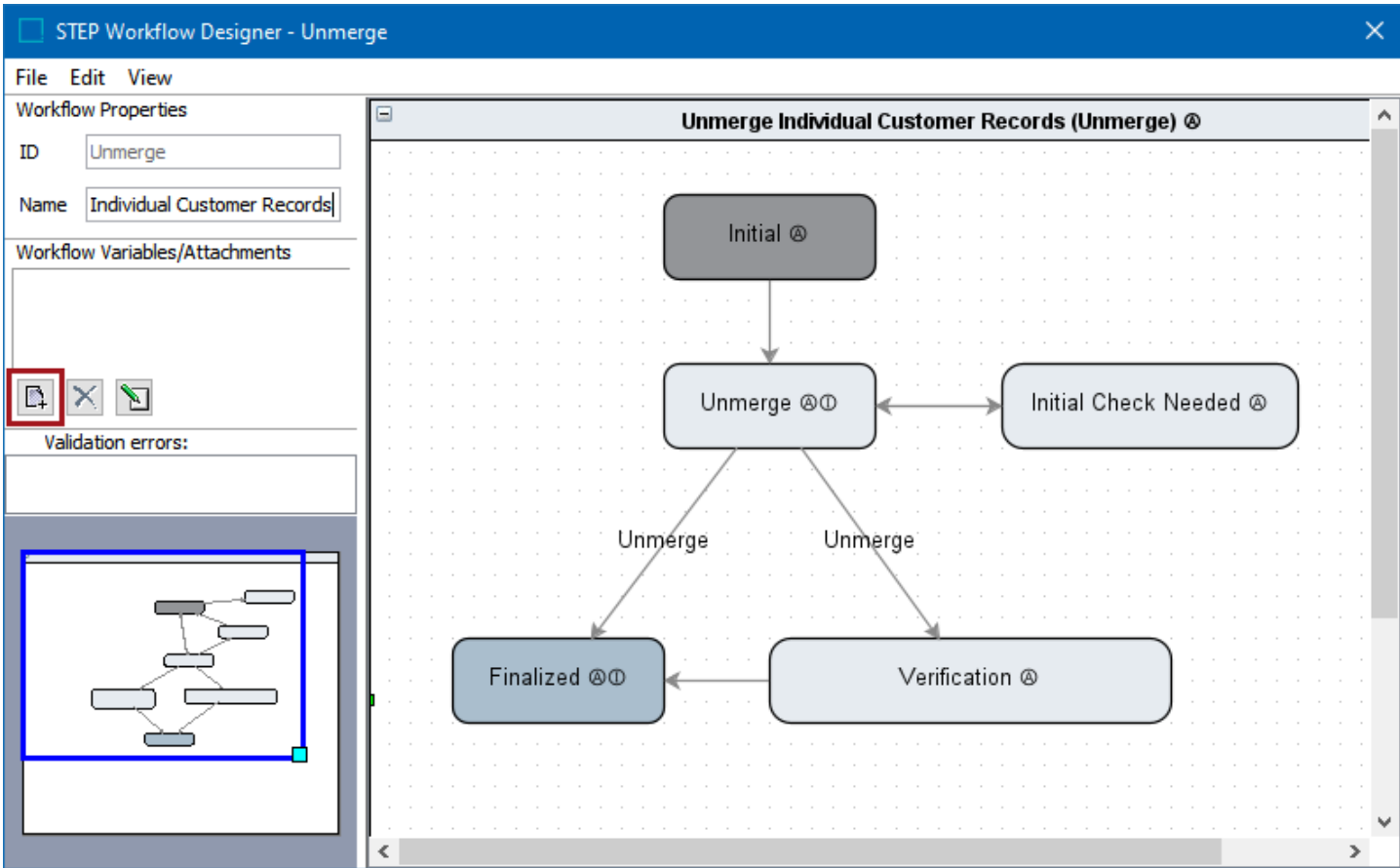
Configuration

Use these steps to configure an unmerge process via workflow in a Match and Merge solution.

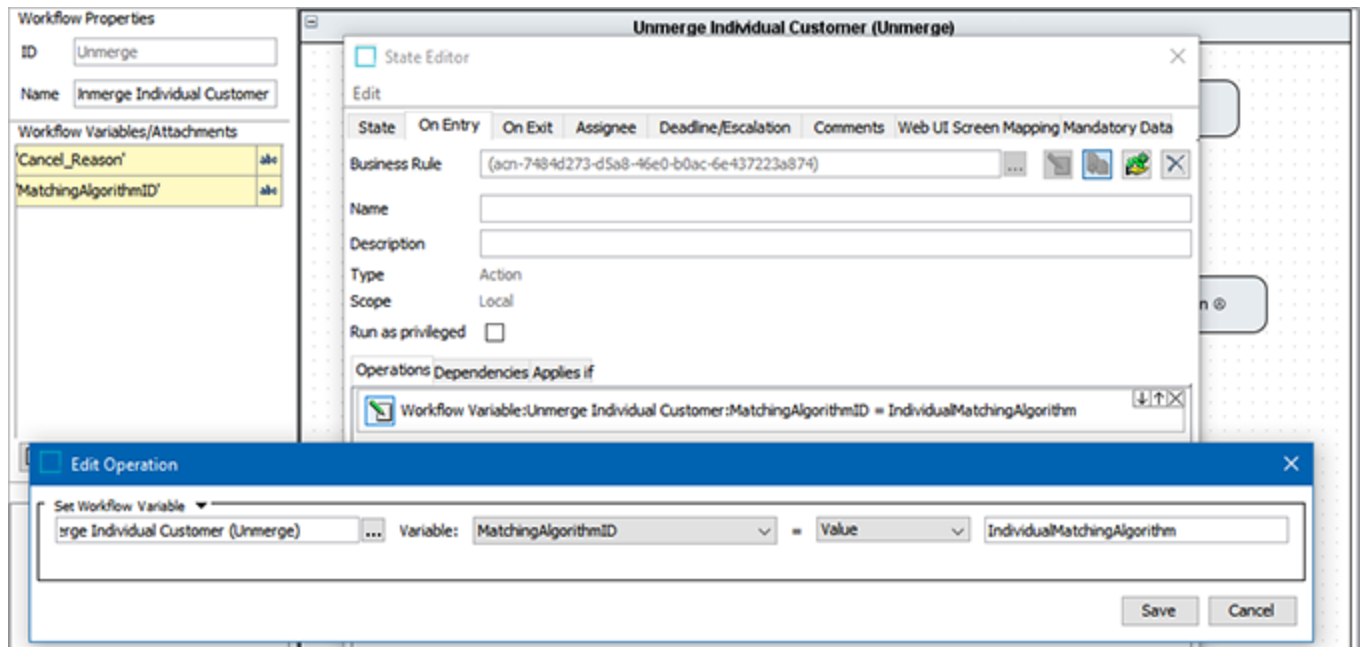
Note: The unmerge workflow below is an example of a complex unmerge workflow. The only requirements for an unmerge workflow are the **matching algorithm ID** and the **object type validity**.

1. On System Setup, create a new workflow for unmerge. For information on setting up a new workflow, see 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**.

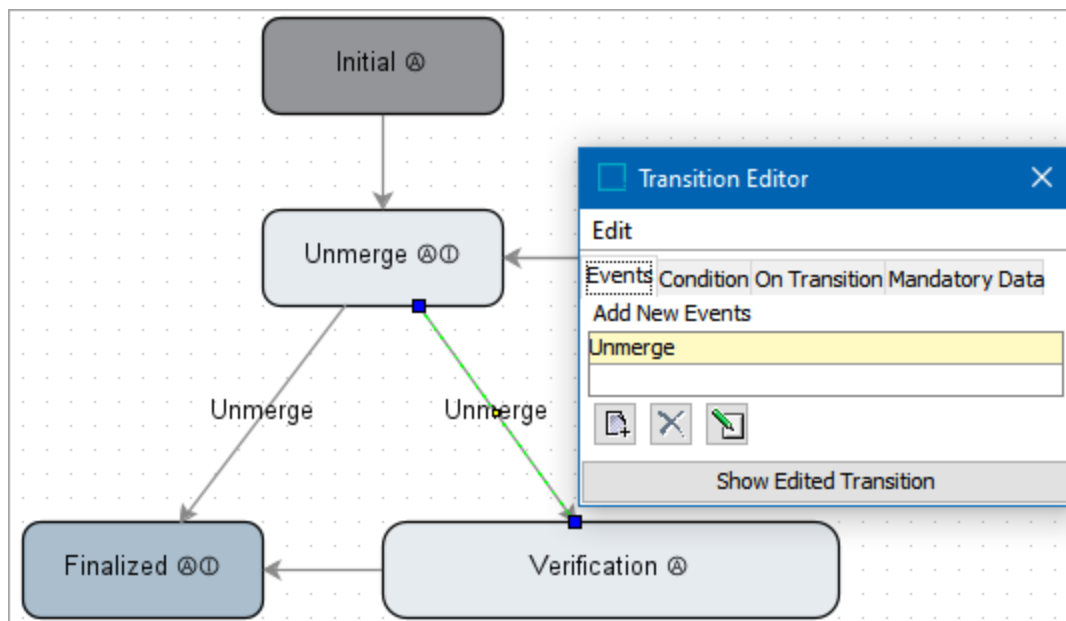


4. Right-click the Unmerge state in the workflow, click **Edit State**, and make the following updates:
 - On the OnEntry tab click the **Add new Business Action** link.
 - Click the **Edit Operation** button and select **Set Workflow Variable** from the dropdown.
 - In the parameters, select the current workflow, the **MatchingAlgorithmID** variable, 'Value' from the dropdown, and the ID of the matching algorithm. For more information, see the **Workflow Variables** topic in the **Workflows** documentation.

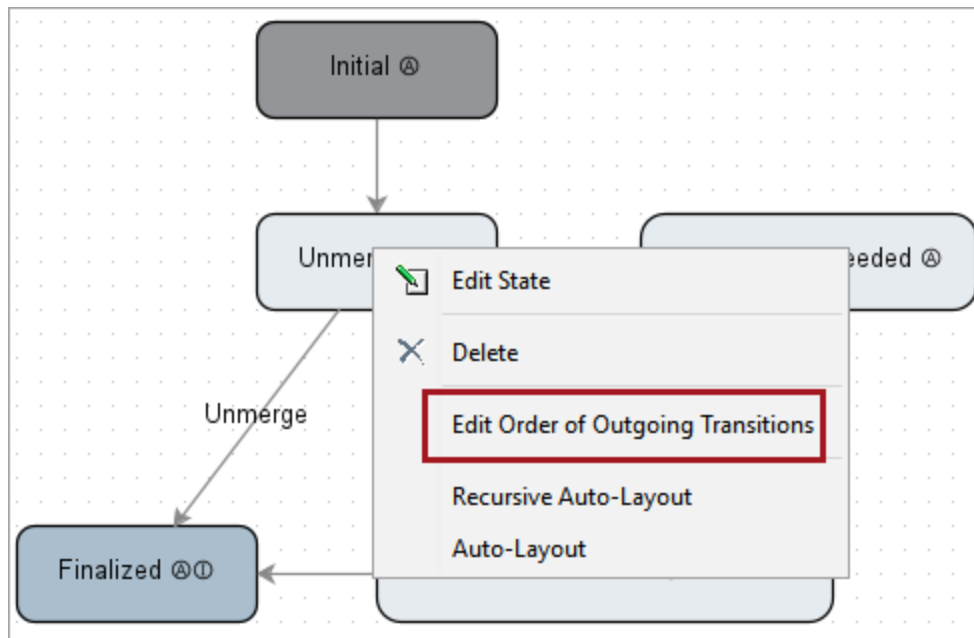


5. If a transition out of the Unmerge state exists but is not valid for a completed unmerge operation, edit the valid transitions to add events named 'Unmerge' to ensure the expected data flow.

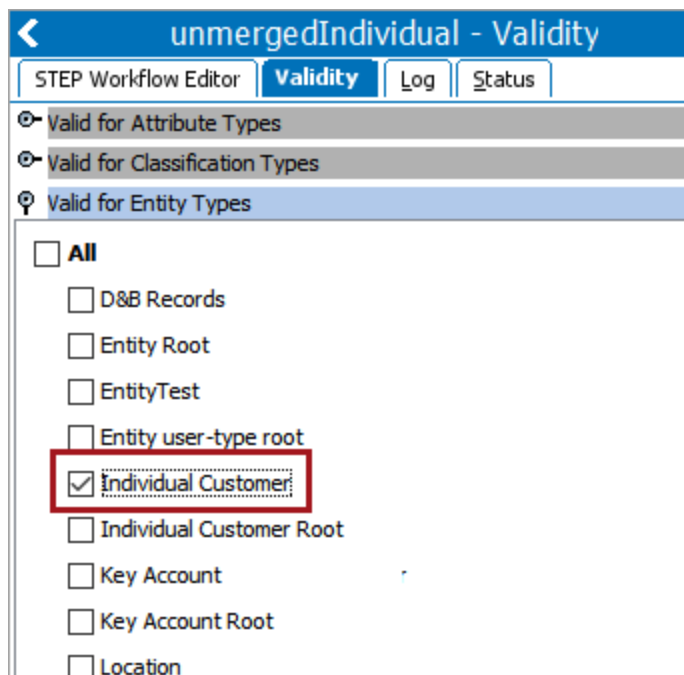
Note: Only transitions (one or more) with events named 'Unmerge' are used when completing the unmerge operation.



- 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.



- On the Workflow, click the Validity tab, and select the unmerge object type.



Creating Merge Golden Record Match Action Handlers

Working with a golden record setup often requires specific actions when a golden record is created or merged. The matching algorithm can be configured to call a business rule via a handler to allow for more granular processing of events. For example, when two existing golden records are merged, actions may need to occur in addition to the survivorship rules.

Note: These handlers are optional and may not be needed in all solutions.

The screenshot displays the configuration interface for the 'Individual Customer Matching Algorithm - Matching Algorithm'. The 'Match Action Configuration' dialog is open, showing settings for the 'Merge Golden Record' action. The 'Match Action Configuration' checkbox is highlighted with a red circle '2'. The 'Select Action' dialog is also open, showing a list of actions. A red circle '3' highlights the 'Select Action' dialog, and a red arrow points to the 'Details' button. A red circle '1' is located at the bottom left of the interface.

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.

The following handlers are available for the match and merge solution:

- **Create Handler** - Create a business action to run on the golden record after it has been created but before survivorship rules run.
 - The newly created golden record is bound to the 'Current Object' parameter. See the **Current Object Bind** topic in the online help **Resource Materials** documentation.
- **Merge Handler** - Create 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.
 - The newly created golden record is bound to the 'Current Object' parameter. See 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. See the **Secondary Object Bind** topic in the online help **Resource Materials** documentation.
- **Merge Keep First Handler** - For both 'match and merge' and match and link' solutions, see the **Creating a Merge Keep First Handler** topic for details.

Configuring Merge Golden Record Match Action

This functionality is used by a Match and Merge solution. For more information, see 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

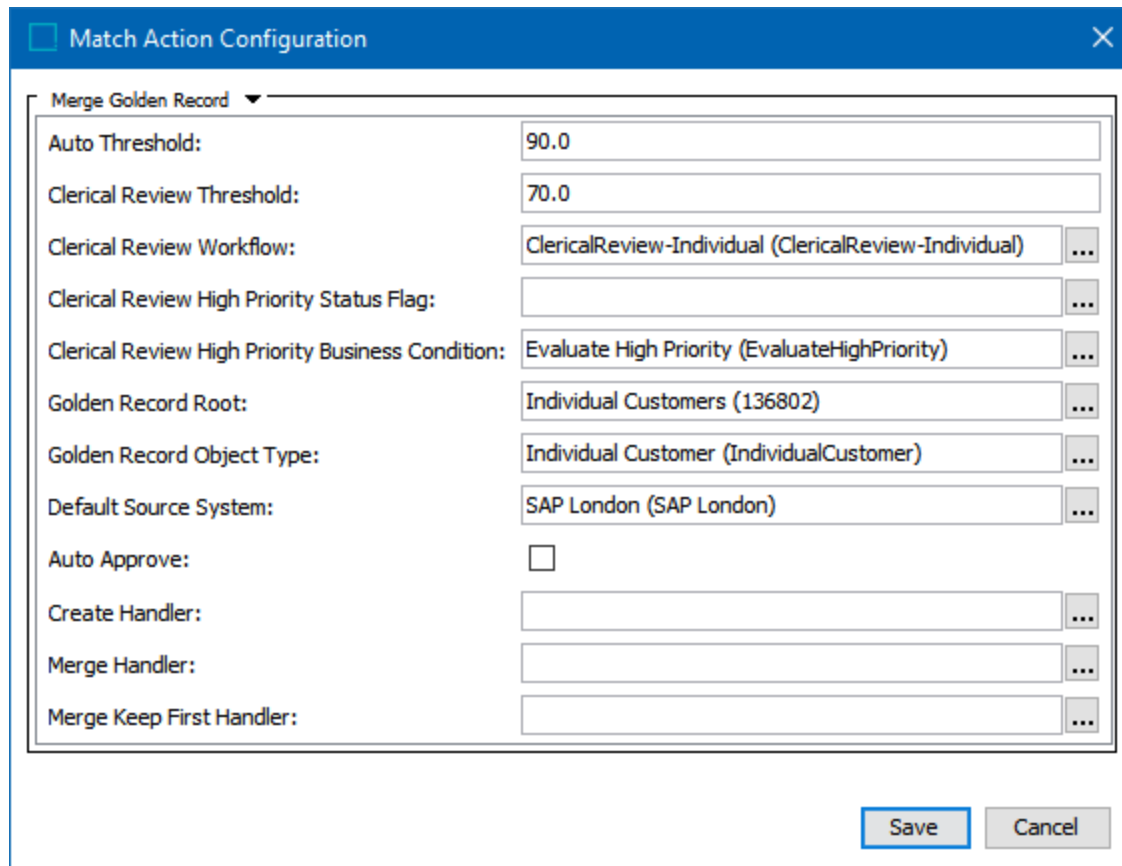
Create or identify the following objects:

- 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:



For information on a parameter, hover over the parameter label to display help text.

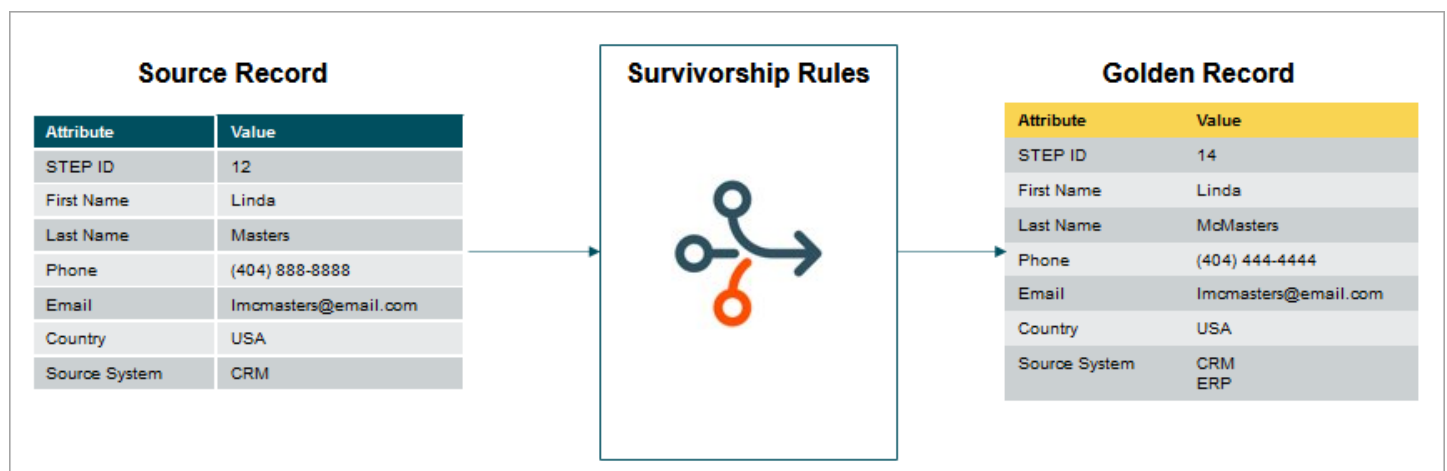
- **Auto Threshold** - add a match score (percentage) to indicate how equal two objects must be to automatically merge them. For more information, see the **Match and Merge** topic.
- **Clerical Review Threshold** - add a match score (percentage) lower than the Auto Threshold to indicate how equal two objects must be to enter the Clerical Review Workflow (where potential duplicates are manually addressed). For more information, see the **Match and Merge** topic.
- **Clerical Review Workflow** - click the ellipsis button (...) and select the relevant clerical review workflow. For more information, see 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, see 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, see 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, see 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, see 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, see 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.



For more information, see 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.

This functionality is used by a Match and Merge solution. For more information, see the **Match and Merge** topic and the **Configuring Match and Merge** topic.

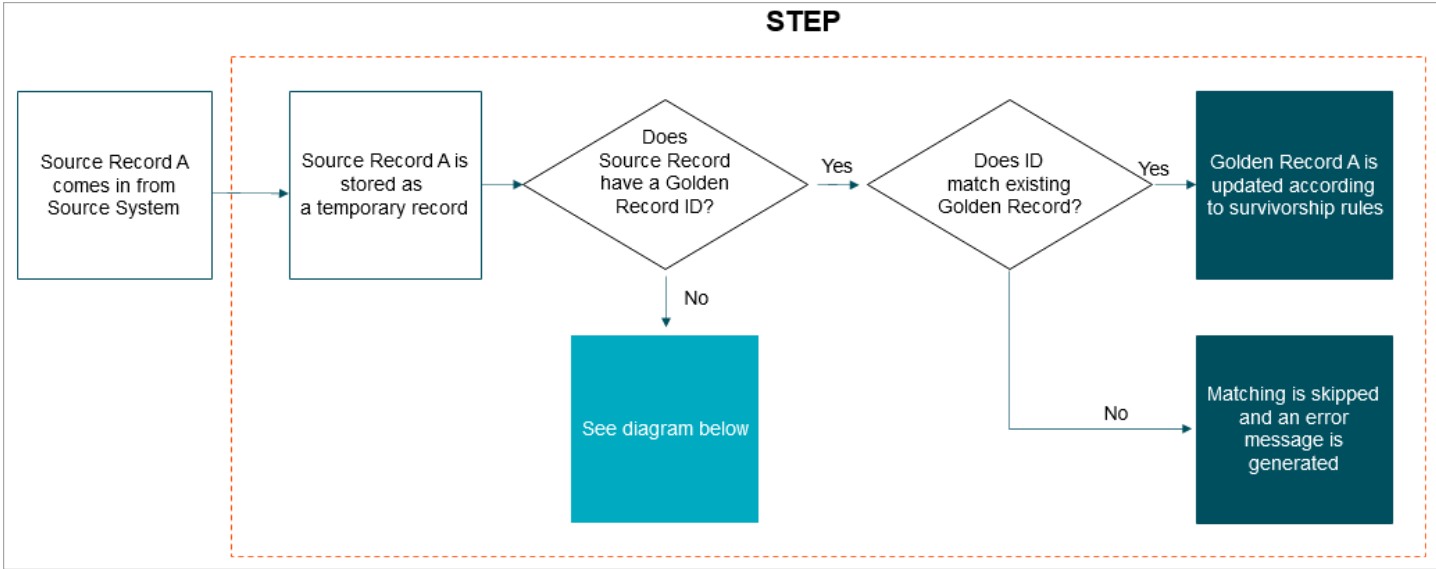
Asynchronous Merge Inbound Integration Endpoint

When match and merge is set up as a data hub, the data can flow into STEP via an asynchronous inbound integration endpoint (IIEP).

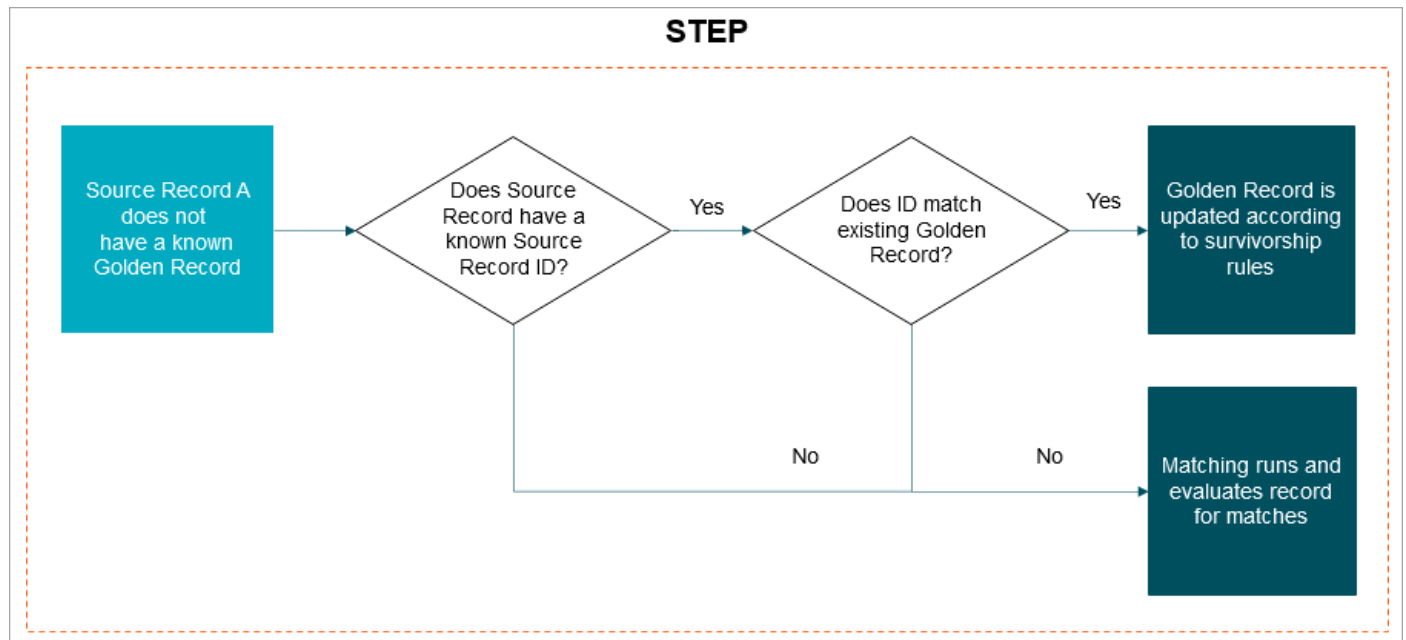
During an import, source records listed in the imported file are created and treated as temporary STEP objects. Despite their temporary nature, business rules can act on them, and they can be matched via a matching algorithm. When being matched, survivorship rules determine which (if any) source values get promoted to existing golden records. After the import process is complete, these temporary objects are discarded.

Imported data is processed in the following order:

1. The IIEP checks the temporary source record to see if it has a known golden record ID. If it does have an ID, then the corresponding golden record in STEP is updated based on the survivorship rules configured in the matching algorithm. If no existing golden record shares the same ID, then the source record is ignored and is not imported to a golden record. A relevant error message is provided.



2. The IIEP checks the temporary source record to see if it has a known source record ID. If an existing active (not deactivated) golden record is found that has the same source ID, survivorship rules are applied to promote source values to the golden record.



3. If neither a golden record ID nor source record ID is found, the configured matching algorithm is invoked and matches the temporary source records to existing golden records in STEP. If a source record matches one or more golden records above the auto threshold, then survivorship rules are applied to promote source values to the golden record that is selected by the match criteria match rules. If multiple golden records match above the auto threshold, those existing golden records are not merged in this process. Temporary source records that are matched below the auto threshold become their own golden records.

Note: Because the imported records are temporary STEP objects, all elements of matching must be prepared to run before references to other objects are created. You cannot query for 'references to' or 'references from' on temporary STEP objects. Furthermore, the match codes can only depend on values on the current object and survivorship rules only update values the current object.

For configuration details, see 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.

Note: The exact behavior of a match and merge web service endpoint depends on the endpoint configuration.

The following topics include more information on:

- Web service endpoints - see the **Web Service Endpoint** topic in the **Data Exchange** documentation.
- Web service merging configuration details - see the **Web Service Endpoint - Match and Merge** topic in the **Data Exchange** documentation.

Configuring the Merge Event Processor

This functionality is used by a Match and Merge solution. For more information, see 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 / Super User Guide** 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 / Super User Guide** documentation.

For more information, see the **Maintaining an Event Processor** topic of the **System Setup / Super User Guide** documentation.

Configuring and Using Match and Merge in Web UI

This functionality is used by a Match and Merge solution. For more information, see the **Match and Merge** topic and the **Configuring Match and Merge** topic.

In Web UI, the elements available for merging and unmerging are defined in the topics below:

- **Clerical Review Task List** - see the **Golden Record Clerical Review Task List** topic.

Golden Record Clerical Review Task List							
<input type="button" value="Clear all"/> <input type="button" value="Reject"/> <input type="button" value="Reassign"/> <input type="button" value="Submit"/> <input type="button" value="Merge"/> <input type="button" value="Advanced merge"/>							
ID	Name	Source Information	First	Last	Email	Phone Number	
CustomerGR229245	Customer003	SAP SAP_003	Theresa	Lebowitz	therlbo@email.com	(615)497-5547	
CustomerGR229247	Customer001	SAP SAP_001	Theresa	Lebowitz	tlebo@email.com	(615)497-8898	
<input checked="" type="checkbox"/>	CustomerGR229243	Customer004	SAP SAP_004	Theresa	Lebowitz	tlebowitz@email.com	(615)497-3333
CustomerGR229244	Customer002	SAP SAP_002	Theresa	Lebowitz	tleebu@email.com	(615)497-4138	
CustomerGR229246	Customer005	SAP SAP_005	Theresa	Lebowitz	theresalebo@email.com	(615)497-0121	
<input type="checkbox"/>	MergeGR32407	(MergeGR32407) SAP	GARED	FULLEN HACKETT	maureen.elzt.2015@gm...	(418)687-8954	
	MergeGR32410	(MergeGR32410) SAP	GARED	FULLEN HACKETT	breanta.alsip.ctl84@gma...	(229)490-7378	

- **Advanced Merge Dialog** - see the **Golden Record Advanced Merge Dialog** topic.

Advanced merge ✕

Exclude Record
 Include Record
 Set as Survivor

	<input type="checkbox"/>	<input type="checkbox"/>	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** - see the **Golden Record Source Traceability Screen** topic.

Olive Johnson INDIVIDUAL CUSTOMER • ID: 248854

[Overview](#)
 [Source Traceability](#)
 [History](#)
 [Household](#)
 [Confirmed Non Matches](#)
 [Household Deduplication](#)

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** - see the **Golden Record Source Information** section of the **Match and Merge Traceability** topic.

Aarone Kirk INDIVIDUAL CUSTOMER • ID: 651262

Overview Source Traceability History Household Confirmed Non Matches Household Deduplication

Quality 100%

Personal Details

First Name

Middle name

Last Name

Main Address

Email

[Add](#)

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** - see the **Configuring the Unmerge Wizard** topic and the **Using the Unmerge Wizard** topic.

Unmerge: Jeff Collins ID: 35005 **1** Distribute Source Records **1** **2** Select Surviving Values

Reset all → Move to → Reactivate Golden Record

	Original Golden Record 35005	Reactivated Golden Record 63003
Sources	<input type="checkbox"/> SAP London - 8518 <input type="checkbox"/> SAP US - 2462	<input type="checkbox"/> 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 - see the **Global Navigation Panel** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.
- via home page link or widget to search for golden records via golden record ID or source record ID - see the **Homepage Widgets** topic in the **Web User Interfaces / Web UI Setup and User Guide** 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

Golden Record Clerical Review Task List

This functionality is used by a Match and Merge solution. For more information, see 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.

Golden Record Clerical Review Task List							
ID	Name	Source Information	First	Last	Email	Phone Number	
<input type="checkbox"/>	CustomerGR229245	Customer003	SAP SAP_003	Theresa	Lebowitz	therlbo@email.com	(615)497-5547
<input type="checkbox"/>	CustomerGR229247	Customer001	SAP SAP_001	Theresa	Lebowitz	tlebo@email.com	(615)497-8898
<input checked="" type="checkbox"/>	CustomerGR229243	Customer004	SAP SAP_004	Theresa	Lebowitz	tlebowitz@email.com	(615)497-3333
<input type="checkbox"/>	CustomerGR229244	Customer002	SAP SAP_002	Theresa	Lebowitz	tleebu@email.com	(615)497-4138
<input type="checkbox"/>	CustomerGR229246	Customer005	SAP SAP_005	Theresa	Lebowitz	theresalebo@email.com	(615)497-0121
<input type="checkbox"/>	MergeGR32407	(MergeGR32407)	SAP	GARED	FULLEN HACKETT	maureen.elzt.2015@gm...	(418)687-8954
<input type="checkbox"/>	MergeGR32410	(MergeGR32410)	SAP	GARED	FULLEN HACKETT	breanta.alsip.ctl84@gma...	(229)490-7378

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 / Web UI Setup and User Guide** documentation.

A separate user group is recommended to be used for this task.

This topic includes details on **Configuration** and **Using Action Buttons**.

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**.

Properties

Configuration Web UI style

Tasklist-ClericalRevi Save Close New... Delete Rename Save as...

Golden Record Clerical Review Task List Properties

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 Head
- Golden Record Attribute Value Header (C
- Golden Record Attribute Value Header (Ir
- Golden Record Data Container Attribute \
- Golden Record Data Container Attribute \

Add... Edit... Remove Up Down

Include Labels

- 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.
 - 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.

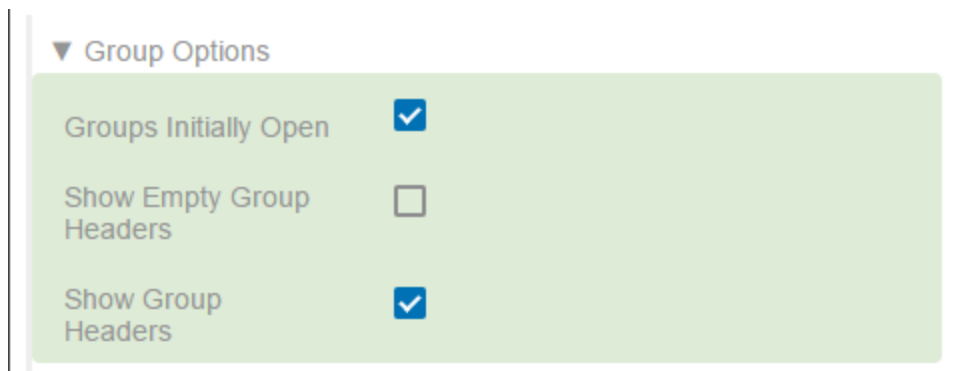
- Included Nested Groups - When checked, attributes from nested parameter groups should be included.
- Label - Specify a label for the header.
- Read Only - When checked, the values listed under this header cannot be edited.
- Show LOV IDs - When checked, relevant LOVs display their IDs next to the corresponding 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.
- 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.

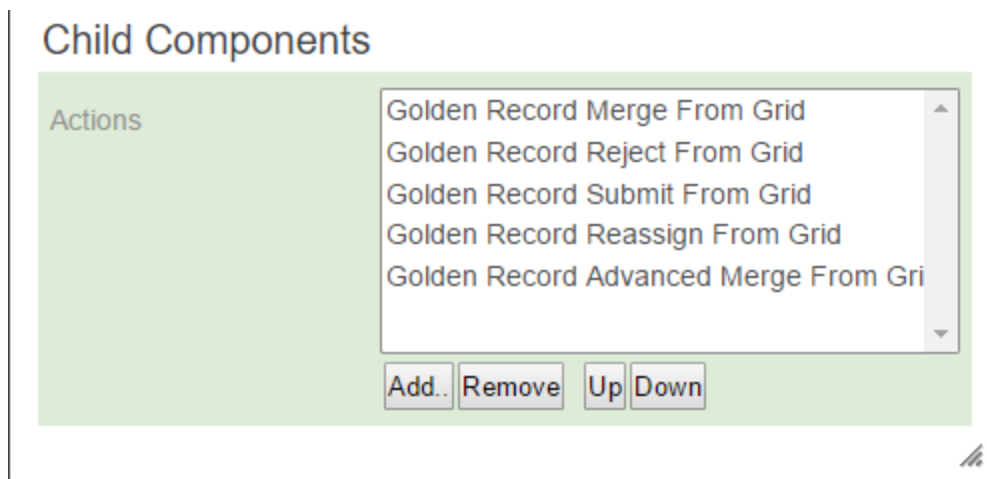
- Golden Record ID Header - Displays the IDs of the golden records in the table. Once selected, double click the component and specify a Label and Preferred Column Width in the configuration dialog.
- Golden Record Matching Agent Recommendation Header - Displays the merge and reject recommendations of the Machine Learning Matching Agent. Once selected, specify a Label and Preferred Column Width in the configuration dialog.
- 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.

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, see the **Dynamic Table Layout** topic of the **Web User Interfaces / Web UI Setup and User Guide** 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** parameter, set the following options:
 - Groups Initially Open - When checked, the attribute groups are opened by default.
 - Show Empty Group Headers - When checked the header are visible if the group is empty.
 - Show Group Headers - When checked, a header displays for attribute groups.



6. For the **Child Components** parameter, 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.



Note: Once an action is added, double click it to complete the required configuration.

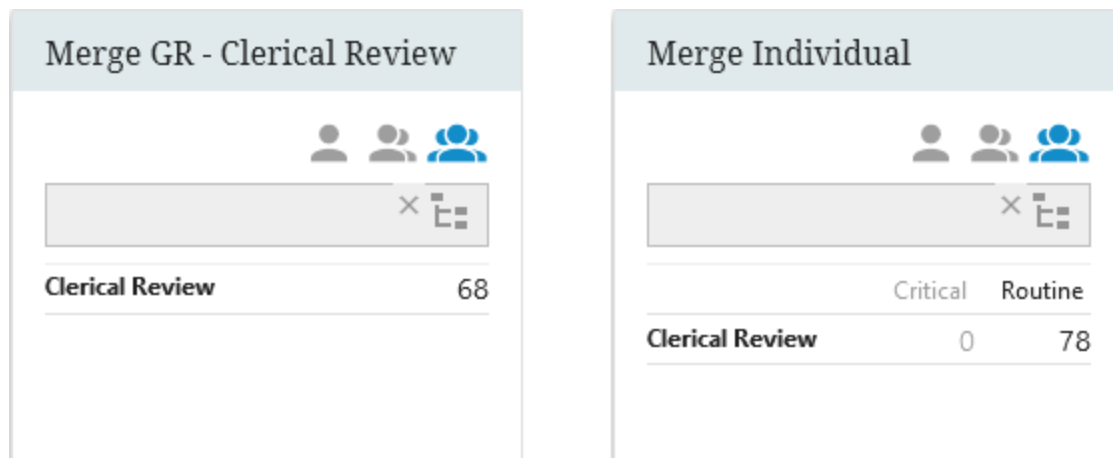
- Click **Save**.
- Assign the Golden Record Clerical Review Task List to a Status Selector Homepage widget or a workflow Status Selector on the Stack Panel Item. For more information on configuring and using this widget, see the **Status Selector Homepage Widget** topic of the **Web User Interfaces / Web UI Setup and User Guide** documentation.

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.

Follow these steps to manage the task list.

- 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.



- Select the desired tasks using one of the following

Golden Record Clerical Review Task List

Clear all
 Reject
 Reassign
 Submit
 Merge
 Advanced merge

ID	Name	Source Information
<input checked="" type="checkbox"/>	CustomerGR225438 CustomerGR225449 CustomerGR225336 CustomerGR225458	CustomerA0001 CustomerA0004 CustomerA0002 CustomerA0003
		SAP 1258; SAP 2651; SAP 8457 SAP 9875; SAP 9888 SAP Y SAP 2245; SAP 8745
<input checked="" type="checkbox"/>	MergeGR224870 MergeGR224943	Customer002 Customer001
		SAP 002 SAP 001

- Select an individual task via the checkbox. If multiple tasks are selected, all actions except the **Advanced merge** action are available.
- Click **Clear all** (if configured) to remove all selections on the screen.
- If configured, filter the list to show only High Priority tasks in the workflow state currently selected via filter at the top of the screen. Filtering is driven by the High Priority Status Flag configured in the match criteria of the relevant matching algorithm. The Match and Merge Clerical Review high priority filter does not respect 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, see the **High Priority Status Flag Global Representation Type** section of the **Main Properties** topic of the **Web User Interface / Web UI Setup and User Guide** documentation.

Golden Record Clerical Review Task List

Show Details
 Merge
 Reject
 Advanced Merge
 Reassign
 High priority : 1

Name	ID	Source	Main Address	Org SICCode
<input type="checkbox"/> Thread Bear	1090007	SFDC	23601 NE 206th St Battle Ground WA 9...	5199
<input type="checkbox"/> Threade Bear	1137071	SFDC	1500 State St Eau Claire WI 54701-407...	5131

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, see 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, see the **Configuring the Matching - Merge Golden Record Component Model** topic.
 - **Reassign** - assigns to another user 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 in the workflow.

- **Merge** - for each task selected, merges 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, see the **Golden Records Survivorship Rules** topic.

For information on deactivation, see 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.

For more information on configuring and using the advanced merge dialog, see the **Golden Record Advanced Merge Dialog** topic.

Golden Record Advanced Merge Dialog

This functionality is used by a Match and Merge solution. For more information, see 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, see the **Configuring the Golden Record Clerical Review Task List** topic.

This topic includes details on **Configuration**, **Header Configuration**, and **Using the View and Buttons**.

Advanced merge
✕

✕ Exclude Record ⊕ Include Record ☰ Set as Survivor

	<input type="checkbox"/>	<input type="checkbox"/>	Merge Preview
ID	CustomerGR229244 (Survivor)	CustomerGR229245	CustomerGR229247
Name	Jack Brown	Jack Brown	Jack Brown
Source Information	SAP SAP_002	SAP SAP_003	SAP SAP_001
Score	70 i	-	70 i
- Details			
First Name	Jack	Jack	Jack
Middle Name	Peter		Peter
Last Name	Brown	Brown	Brown
Email	jackb@email.com	jb@email.com	jbrown@email.com
PhoneNo	(615)497-2222	(615)497-1111	(615)497-3333
Weight	41 kg	74 kg	75 kg
Customer Reference	>Customer005>Customer0001	>Customer002	>CustomerA0003
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-...

➤ Merge
✕ Reject
👤 Reassign
➡ Submit
Cancel

Note: The Reject, Reassign, and Submit 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 / Web UI Setup and User Guide** 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, see 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)

Configuration Web UI style

Golden Record Cleri ▾ Save Close New... Delete Rename Save as

Golden Record Advanced Merge Dialog Properties [go to parent](#)

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 Informatio
Rank Score Header

Add... Edit... Remove Up Down

▼ 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, see 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

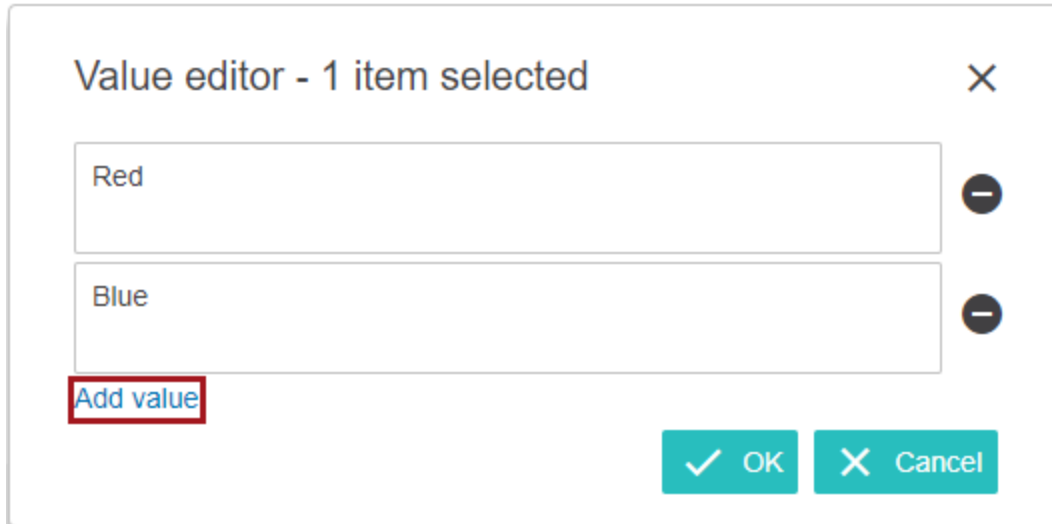
The 'Rank Score Header' displays the 'match score' of golden records as they compare to the surviving golden record. Configuration options include:

- 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, see the **Main Properties** topic in the **Web User Interfaces / Web UI Setup and User Guide** 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, see the **Main Properties** topic in the **Web User Interfaces / Web UI Setup and User Guide** 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 ...

Cancel
Add

To configure which attribute values / references display in this component and under what conditions they are displayed, see the **Global Data Container Representations** topic of the **Web User Interfaces / Web UI Setup and User Guide** 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, see the **Data Containers in Web UI** topic of the **Web User Interfaces / Web UI Setup and User Guide** documentation.

Main Address		
39 Ridge Line Ct Oroville C...	(InputCity)	<input type="text" value="Oroville"/>
	(InputCountry)	<input type="text" value="US"/>
	(InputState)	<input type="text" value="CA"/>
	(InputStreet)	<input type="text" value="39 Ridge Line Court"/>
	(InputZip)	<input type="text" value="95966"/>
	(CalcFormattedA...)	39 Ridge Line Ct Oroville CA 95966-9479 United States
	(StandardizedFor...)	39 Ridge Line Ct Oroville CA 95966-9479

Continue

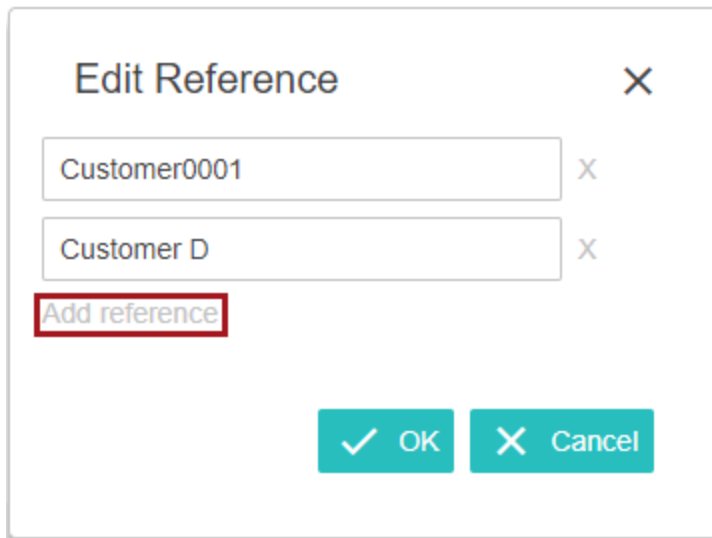
Advanced Merge Reference Header

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.



Deprecated: Advanced Merge Data Container Header

The 'Advanced Merge Data Container Header' has been withdrawn as of STEP 10.0. In its place, users are recommended to use the replacement component, Advanced Merge Globally Configured Data Container Header, which is described previously in this topic.

The 'Advanced Merge Data Container Header' displays data container values. Configuration options include:

- Data Container Type - Click the ellipsis button (...) and browse or search for the relevant data container type to display attributes from.
 - Display Attribute - Click the ellipsis button (...) and browse or search for the relevant calculated attribute to display data containers listed in the editor. This calculated attribute should convey the most important details for differentiating the individual data containers. For information on calculated attributes, see the **Calculated Attributes** topic of the **System Setup / Super User Guide** documentation.
 - Editable Attributes - Click the ellipsis button (...) and browse or search for the attribute group that contains attributes whose values are editable.
 - Standardized Attributes - Click the ellipsis button (...) and browse or search for a relevant attribute group that displays standardized values for data containers listed in the editor.
 - Metadata Attributes - Click the ellipsis button (...) and browse or search for a relevant attribute group that displays meta values for data containers listed in the editor.
 - Label - Specify a label for the header.
 - Context Help - Enter help text for the component to display.
 - Display Context help - Check the box to display context help text for attributes.
- 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. For more information on creating / maintaining data containers using the Data Container Attribute View Editor, see the **Data Containers in Web UI** topic of the **Web User Interfaces / Web UI Setup and User Guide** documentation.

Edit Contact
✕

<div style="background-color: #0070c0; color: white; padding: 2px;">Larry Toombs, LarryToombs@email.com, ... ✕</div> <div style="padding: 2px;">Chuck Woods, CW77@email.com, 777-44...</div> <div style="text-align: center; padding: 5px 0;">Add Contact...</div>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-right: 1px solid #ccc; padding: 2px;">First Name</td> <td style="padding: 2px;"><input type="text" value="Larry"/></td> </tr> <tr> <td style="border-right: 1px solid #ccc; padding: 2px;">Middle Name</td> <td style="padding: 2px;"><input type="text"/></td> </tr> <tr> <td style="border-right: 1px solid #ccc; padding: 2px;">Last Name</td> <td style="padding: 2px;"><input type="text" value="Toombs"/></td> </tr> <tr> <td style="border-right: 1px solid #ccc; padding: 2px;">Email</td> <td style="padding: 2px;"><input type="text" value="LarryToombs@email.com"/></td> </tr> <tr> <td style="border-right: 1px solid #ccc; padding: 2px;">PhoneNo</td> <td style="padding: 2px;"><input type="text" value="888-222-1111"/></td> </tr> </table>	First Name	<input type="text" value="Larry"/>	Middle Name	<input type="text"/>	Last Name	<input type="text" value="Toombs"/>	Email	<input type="text" value="LarryToombs@email.com"/>	PhoneNo	<input type="text" value="888-222-1111"/>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-right: 1px solid #ccc; padding: 2px;">Std Email</td> <td style="padding: 2px;">LToombs@email.com</td> </tr> <tr> <td style="border-right: 1px solid #ccc; padding: 2px;">Std PhoneNo</td> <td style="padding: 2px;">+1 888-222-1111</td> </tr> <tr> <td style="border-right: 1px solid #ccc; padding: 2px;">Updated</td> <td style="padding: 2px;">7/17/2017 10:30:00 PM</td> </tr> </table>	Std Email	LToombs@email.com	Std PhoneNo	+1 888-222-1111	Updated	7/17/2017 10:30:00 PM
First Name	<input type="text" value="Larry"/>																	
Middle Name	<input type="text"/>																	
Last Name	<input type="text" value="Toombs"/>																	
Email	<input type="text" value="LarryToombs@email.com"/>																	
PhoneNo	<input type="text" value="888-222-1111"/>																	
Std Email	LToombs@email.com																	
Std PhoneNo	+1 888-222-1111																	
Updated	7/17/2017 10:30:00 PM																	

✓ OK
✕ Cancel

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.

Advanced merge ✕

Exclude Record
 Include Record
 Set as Survivor

	<input type="checkbox"/>	<input type="checkbox"/>	Merge Preview
ID	CustomerGR229244 (Survivor)	CustomerGR229245	CustomerGR229247
Name	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
Middle Name	Peter		Peter
Last Name	Brown	Brown	Brown
Email	jackb@email.com	jb@email.com	jbrown@email.com
PhoneNo	(615)497-2222	(615)497-1111	(615)497-3333
Weight	41 kg	74 kg	75 kg
Customer Reference	>Customer005>Customer0001	>Customer002	>CustomerA0003
Contacts	Larry Toombs, LarryToombs@ent	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 take actions on the selected records.

- Click the **Exclude Record** 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 Record** 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.

	<input type="checkbox"/>	<input type="checkbox"/>
ID	CustomerGR225336 (Survivor)	CustomerGR225438

2. The golden record view in the center of the dialog lists all mapped headers.

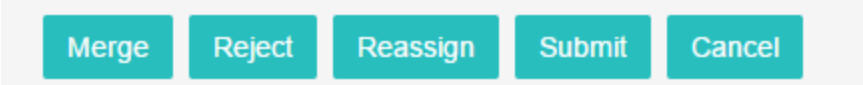
- Individual values selected for promotion to the surviving record are highlighted in blue.
- Values promoted based on survivorship rules appear in bold.

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:
- **Merge** the records into one surviving record. All records are automatically included in the merge unless they have been manually excluded via the **Exclude Record** action button.
 - **Reject** the records as the duplicates and close the task. The Exclude / Include buttons can be used to reject matches on a per-record basis.
 - **Reassign** the task to another user.
 - **Submit** the task to the next state in the workflow.
 - **Cancel** the dialog.



Golden Record Source Traceability Screen

This functionality is used by a Match and Merge solution. For more information, see 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 see when the changes were made.

Olive Johnson INDIVIDUAL CUSTOMER • ID: 248854

Overview Source Traceability History Household Confirmed Non Matches Household Deduplication

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 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 name	Value history Source Revision History <input type="checkbox"/>	
Last Name	Value	Source
Last Edited	Olive	SAP US - 38244430-7946
Source System	Olivia	SAP London - 16840504-2501
		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 name	Value history Source Revision History <input checked="" type="checkbox"/>	
Last Name	Value	Source
Last Edited	Olive	SAP US - 38244430-7946
Source System	Olivia	SAP London - 16840504-2501
		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
Deactivated	Olivia	SAP London - 16840504-2501
		Unused
		2020-05-12 15:27:34 CEST

For more information, see 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 / Web UI Setup and User Guide** 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**.

Properties

Configuration Web UI style

MGR-Customer-Sou ▼ Save Close New... Delete Rename Save as...

Golden Record Source Traceability Properties

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 Group

Data Container Attribute Value

Reference Type

Add.. Remove Up Down

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 / Web UI Setup and User Guide** documentation.

Configuring the Unmerge Wizard

This functionality is used by a Match and Merge solution. For more information, see 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, see 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 / Web UI Setup and User Guide** 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 / Web UI Setup and User Guide** documentation.

Add Screen

Screen ID

unmergeWizard

- Power BI Screen
- Power Search
- Print On Demand
- Product Editor
- Product Summary
- Recycle Bin Screen
- Search Statistics
- Task List
- Unmerge Wizard Screen
- User Details
- User List
- Workflow Profile Screen

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

Cancel
Add

2. On the Unmerge Wizard Screen configure the following parameters.

Properties

Configuration Web UI style

unmergeWizard Save Close New... Delete Rename Save as...

Unmerge Wizard Screen Properties

The Unmerge Wizard Screen allows users to separate source records and Deactivated Golden Records that were incorrectly merged into an existing Golden Record.

Component Description 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)

MainAddressDataContainer (datacontainertype)

EmailDataContainer (datacontainertype)

PhoneDataContainer (datacontainertype)

CustomerToDivision (entityreferencetype)

Add... Remove Up Down

- 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 / Web UI Setup and User Guide** 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

Add...
Edit...
Remove
Up
Down

* Screen

unmergeWizard
▼
Add

Cancel Add

5. Click **Save**.
6. If data stewards use an unmerge workflow, add an unmerge button on the entity details screen(s). For more information, see 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.

Properties

Configuration Web UI style

Details-Individual Save Close New... Delete Rename Save as...

Start Workflow Action Properties [go to parent](#)

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

▼ Advanced

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, see 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 / Web UI Setup and User Guide** documentation.

Properties (edited)

Configuration Web UI style

Unmerge Task List Save Close New... Delete Rename Save as...

Task List Properties

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

▶ Advanced

Child Components

Node List Node List [go to component](#)

- 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)

Configuration Web UI style

Unmerge Task List Save Close New... Delete Rename Save as...

Node List Properties [go to parent](#)

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

Include Labels

Lookup Screen Type For Navigation

Page Size

Use Details Overlay

Child Components

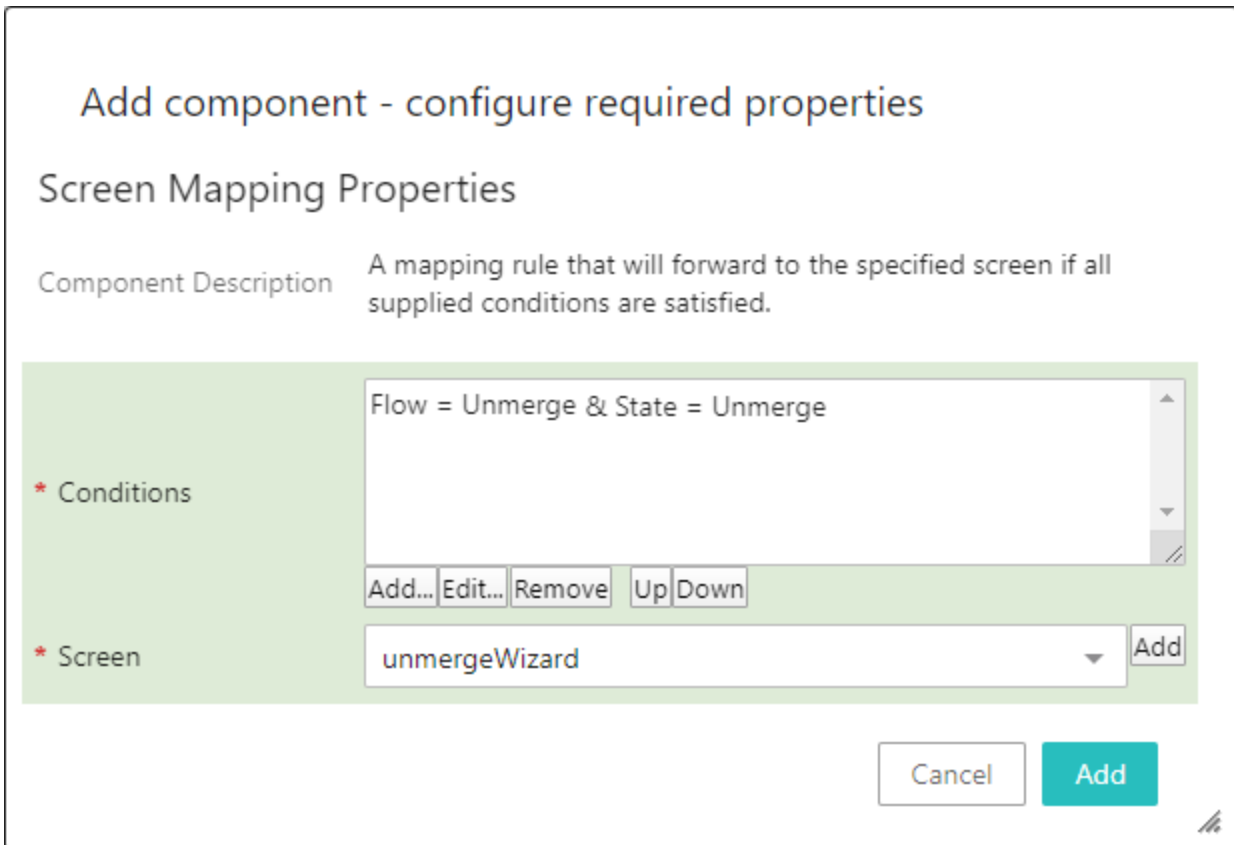
Display Modes

Add.. Remove Up Down

Actions

Add.. Remove Up Down

9. Click **Save**.
10. Add the screen mapping as defined in the **Mappings** topic in the **Web User Interfaces / Web UI Setup and User Guide** 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.



Add component - configure required properties

Screen Mapping Properties

Component Description A mapping rule that will forward to the specified screen if all supplied conditions are satisfied.

*** Conditions**

Flow = Unmerge & State = Unmerge

Add...
Edit...
Remove
Up
Down

*** Screen**

unmergeWizard

Add

Cancel Add

11. Click **Save**.
12. If data stewards use an unmerge workflow, add the unmerge workflow to homepage widget. For more information, see 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 / Web UI Setup and User Guide** 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.

Auto Refresh Interval	<input type="text" value="0"/>
Component Title	<input type="text" value="Unmerge"/>
Initiate Label	<input type="text" value="Initiate"/>
Initiate Screens	<div style="border: 1px solid #ccc; height: 60px; width: 100%;"></div> <div style="border: 1px solid #ccc; padding: 2px;">main</div> <div style="display: flex; gap: 5px;"> <input type="button" value="Add"/> <input type="button" value="Remove"/> <input type="button" value="Up"/> <input type="button" value="Down"/> </div>
* Result Screen	<div style="border: 1px solid #ccc; padding: 2px;">tasklist-unmerge-individualcus</div> <div style="display: flex; gap: 5px;"> <input type="button" value="..."/> <input type="button" value="Add"/> </div>
Show Collection Filter	<input type="checkbox"/>
Collection Top Nodes	<div style="border: 1px solid #ccc; height: 60px; width: 100%;"></div> <div style="display: flex; gap: 5px;"> <input type="button" value="Add..."/> <input type="button" value="Remove"/> <input type="button" value="Up"/> <input type="button" value="Down"/> </div>
Show Initiate	<input type="checkbox"/>
Status Flags Enabled	<input type="checkbox"/>
Show Status Flag Headers	<input type="checkbox"/>
Show Total	<input type="checkbox"/>
* States	<div style="border: 1px solid #ccc; padding: 2px;"> Before_unmerge Unmerge After_unmerge Completed2 </div> <div style="border: 1px solid #ccc; padding: 2px;">AttributeMaintenance End</div> <div style="display: flex; gap: 5px;"> <input type="button" value="Add"/> <input type="button" value="Remove"/> <input type="button" value="Up"/> <input type="button" value="Down"/> </div>
Total Label	<input type="text" value="Total"/>
* Workflow	<input type="text" value="Unmerge"/>

13. Click **Save** and **Close**.

Using the Unmerge Wizard

This functionality is used by a Match and Merge solution. For more information, see 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.

Unmerge: Jeff Collins ID: 35005 1 Distribute Source Records 2 Select Surviving Values 1

Reset all Move to Reactivate Golden Record 2

	Original Golden Record 35005	New Golden Record																																
Sources 3	<input type="checkbox"/> SAP London - 8518 <input type="checkbox"/> SAP US - 2462 <input type="checkbox"/> Deactivated Golden Record 63003 <input type="checkbox"/> Dynamics Europe - 4323	Select a source record to move this new golden record.																																
Surviving Values 4	<table border="1"> <tr> <td>Name</td> <td>Jeff Collins</td> <td>2 unused</td> </tr> <tr> <td>First Name</td> <td>J.</td> <td>2 unused</td> </tr> <tr> <td>Last Name</td> <td>Collins</td> <td></td> </tr> <tr> <td>Credibility Score</td> <td>6</td> <td>2 unused</td> </tr> <tr> <td>Main Address</td> <td>305th Ave Hadley, Massachusetts, 01035 USA</td> <td>7 unused</td> </tr> <tr> <td rowspan="3">Phone</td> <td>Business: 555-6412</td> <td>3 unused</td> </tr> <tr> <td>Private: 514-7258</td> <td>4 unused</td> </tr> <tr> <td>Other: 514-5416</td> <td></td> </tr> <tr> <td rowspan="2">Email</td> <td>jeff.collins@yahoo.com</td> <td></td> </tr> <tr> <td colspan="2">4 sources for Email</td> </tr> <tr> <td rowspan="2">Company Code Data</td> <td>MAG Germany</td> <td>1 unused</td> </tr> <tr> <td colspan="2">3 unused for Company Code Data</td> </tr> </table>		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	
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																																	

Cancel Unmerge Select Surviving Values

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. See 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. See 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, see 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, see 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 **1** **2** Select Surviving Values

Reset all → Move to **2** Reactivate Golden Record

3 Reactivated Golden Record 63003 Record 35005 Reactivated Golden Record 63003

Sources **3** Create new Golden Record

1 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 4 unused	305th Ave Hadley, Massachusetts, 01035 USA 1 unused

Unmerge displays the results for all golden records.

Unmerge: Jeff Collins ID: 35005 **1** Distribute Source Records **1** — **2** Select Surviving Values

Reset all → Move to ▾ Reactivate Golden Record

	Original Golden Record 35005	Reactivated Golden Record 63003	New Golden Record 1
Sources ^			
	<input type="checkbox"/> SAP London - 8518 ▾	<input type="checkbox"/> Dynamics Europe - 4323 ▾	<input type="checkbox"/> 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 <small>1 unused</small>	7
Main Address	305th Ave Phoenix, Arizona (AZ), 85027 US ▾	305th Ave Hadley, Massachusetts, 01035 USA ▾	305th Ave Hadley, Massachusetts (MA), 01035 US ▾

- Click the **Select Surviving Values** button to continue with the unmerge or click **Cancel Unmerge** to close the dialog without making changes.
- 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.

- On the Distribute Source Records step, select a deactivated source record and click the **Reactivate Golden Record** toolbar action button.

Reset all → Move to ▾ Reactivate Golden Record **2**

Original Golden Record 35005

Sources ^

- SAP London - 8518 ▾
- SAP US - 2462 ▾
- 1** Deactivated Golden Record 63003 ▾
 - Dynamics Europe - 4323 ▾

- 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 **1** **2** Select Surviving Values

Reset all → Move to ▾ Reactivate Golden Record

	Original Golden Record 35005	Reactivated Golden Record 63003
Sources	<input type="checkbox"/> SAP London - 8518 <input type="checkbox"/> SAP US - 2462	<input type="checkbox"/> 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

Cancel Unmerge Select Surviving Values

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 ^	
	<ul style="list-style-type: none"> • SAP London - 18840504-2501 • Deactivated Golden Record 53005 <ul style="list-style-type: none"> • Dynamics Europe - 129610-4248
Surviving Values ^	
Name	Jen Collins
First Name	Jen Collins Dynamics Europe - 129610-4248 10/27/2020, 9:51:49 AM
Last Name	Jennifer Collins SAP London - 18840504-2501 10/27/2020, 9:46:03 AM
Main Address	(None) 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.

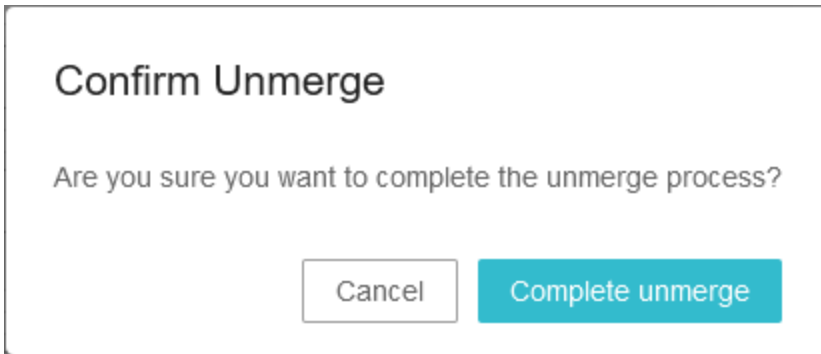
	Original Golden Record 25005
Sources ^	
	<ul style="list-style-type: none"> • SAP London - 18840504-2501 • Deactivated Golden Record 53005 <ul style="list-style-type: none"> • Dynamics Europe - 129610-4248
Surviving Values ^	
Name	Jen Collins
First Name	Jennifer
Last Name	Collins
Main Address	<div style="border: 1px solid #ccc; padding: 5px;"> <p>Collins</p> <p>Dynamics Europe - 129610-4248 10/27/2020, 9:51:49 AM</p> <p>SAP London - 18840504-2501 10/27/2020, 9:46:03 AM</p> <p>(None)</p> </div>

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.



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.



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.

The screenshot shows a 'Match Action Configuration' dialog box with a blue header and a close button (X). The main content area is titled 'List Processing Deduplicate Records' with a dropdown arrow. It contains two input fields: 'Auto Threshold' with the value '90.0' and 'Merge Keep First Handler' with the value 'LP_MergeKeepFirstHandler_Dedup (LP_MKFH)'. A small icon is visible to the right of the second field. At the bottom right, there are 'Save' and 'Cancel' buttons.

For more information, see 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.

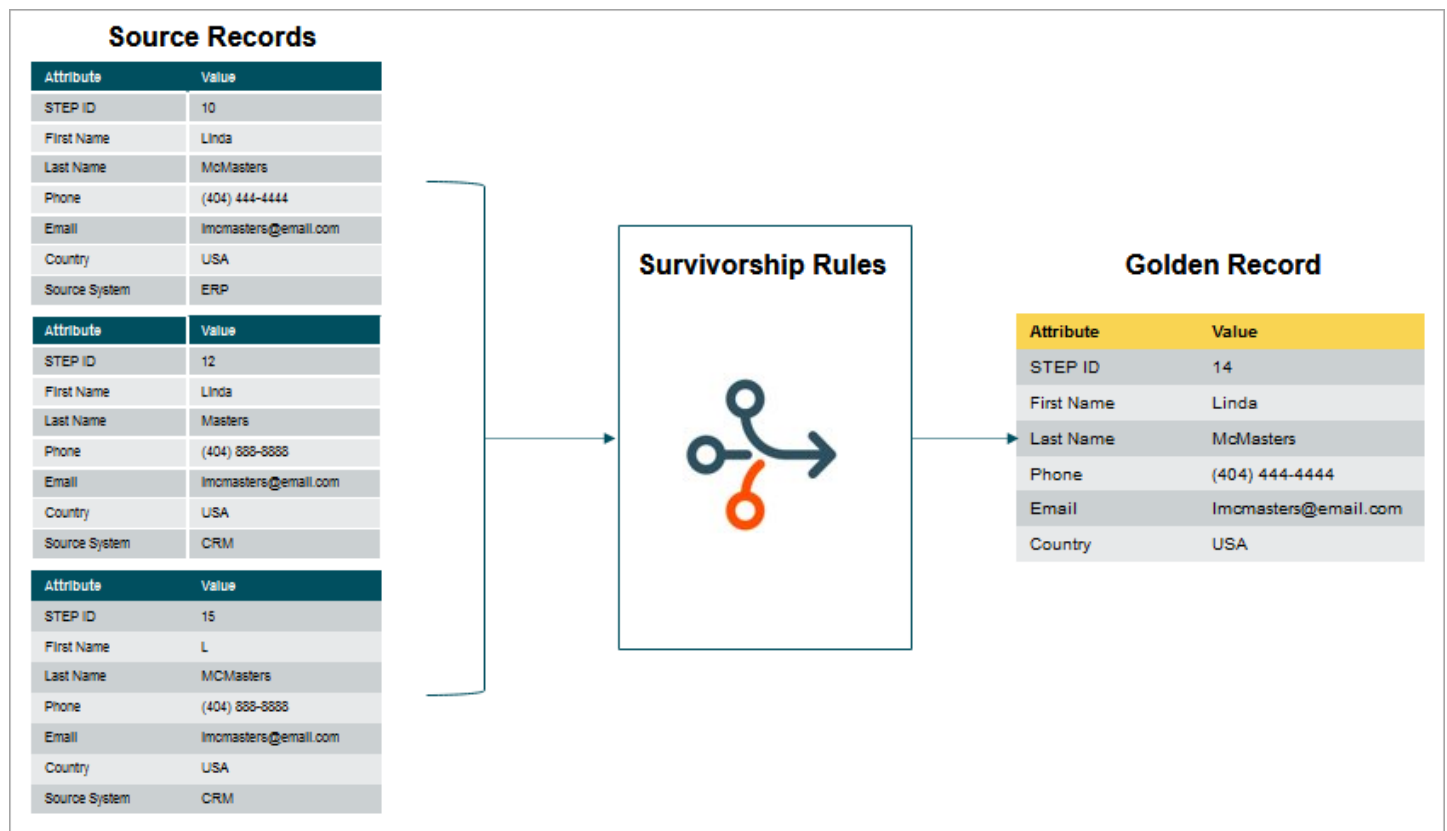
For more information, see the following topics:

- **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.



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, see 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, see 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, see 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.

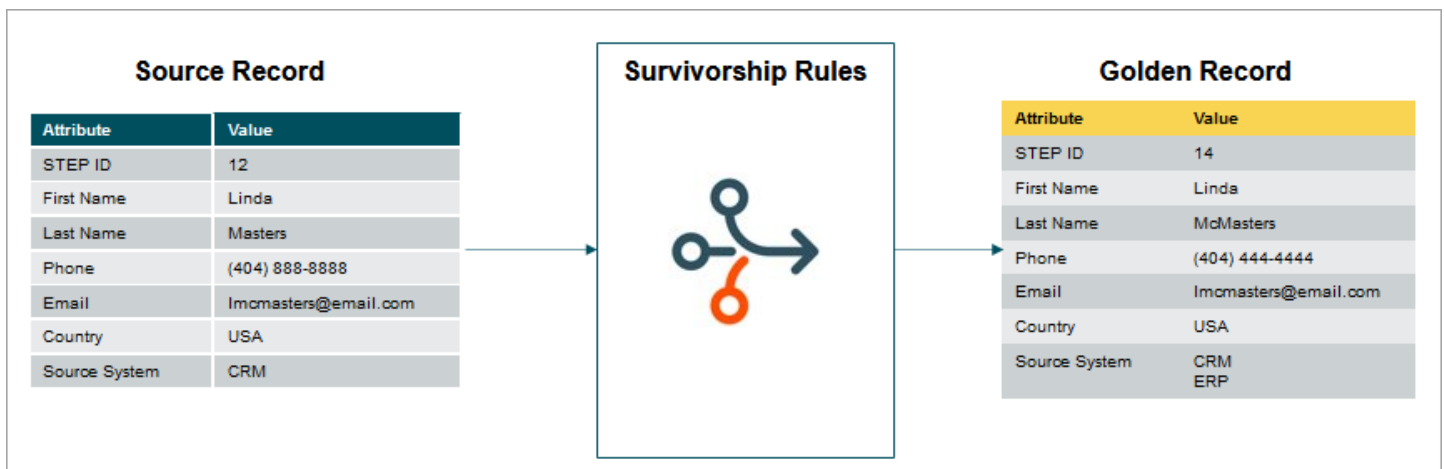
Note: A survivorship rule should never update values outside the golden record.

For more information, see 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.



For more information, see 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.

Edit Operation
✕

Execute JavaScript ▾

Binds: ✕

Variable name	Binds to
targetNode	Current Object
manager	STEP Manager
log	Logger
sourceNodes	Survivorship Rule Source Objects

Messages: ✕

Variable name	Message	Translations

JavaScript:

```

1 // The data container types that this survivorship javascript will handle.
2 var survivorDataContainerTypeIDsInput = ["SAPCustomerCompanyCodeData"];
3
4 // This description of your data model describes which attributes and references
5 var survivorshipDataModelInput = {
6   "SAPCustomerCompanyCodeData": {
7     "CombinedUniqueAttributeIDs" : [
8     ],
9     "CombinedUniqueDctReferenceTypes" : [
10    "SAPCustomerCompanyCodeDataCompanyCode"
11  ],
12  "SurvivingAttributes" : [
13    "SAP-XAUSZ",
14    "Bank Statement Comment",
15    "SAP-NODEL",

```

Edit externally

Save
Test JavaScript
Cancel

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, see 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, see **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 from 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

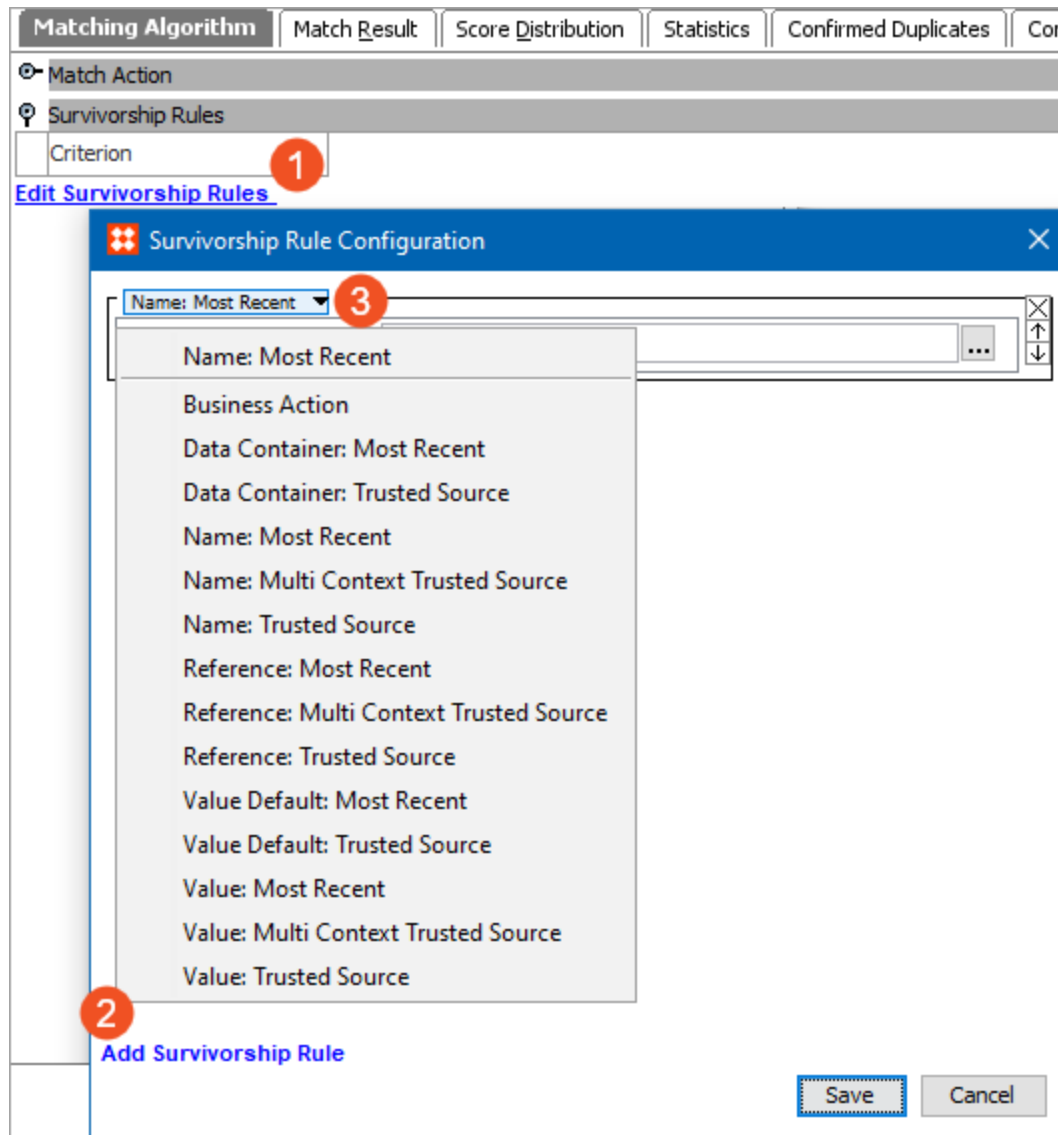
Survivorship rules are stored on a matching algorithm and define which source data is added to the golden record for the selected merging or linking solution. For more information, see the **Golden Records Survivorship Rules** topic.

For details on how to 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.



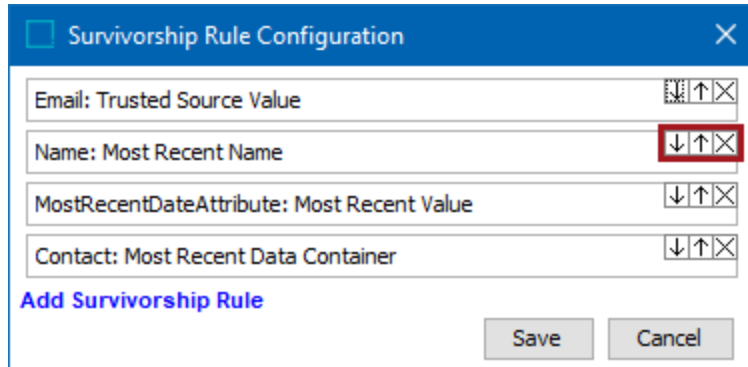
3. Provide the required parameter information for the selected rule. Parameter details are included in the following topics:

- 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.



For more information on matching algorithms, see 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.

- **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, see 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

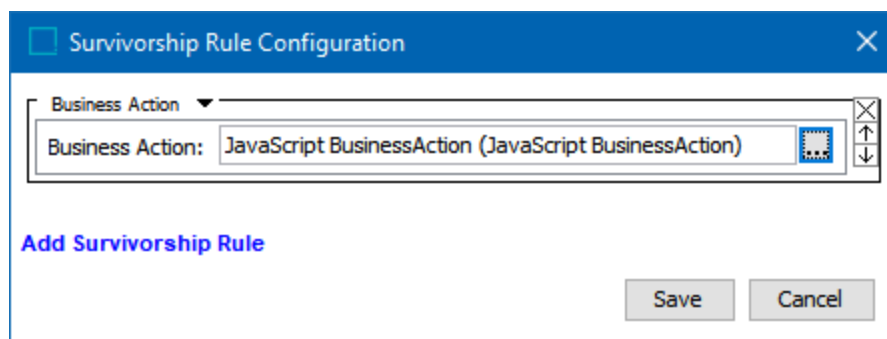
On a matching algorithm, the following rule is available to define promotion to a golden record via a business action.

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 - 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, see 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.

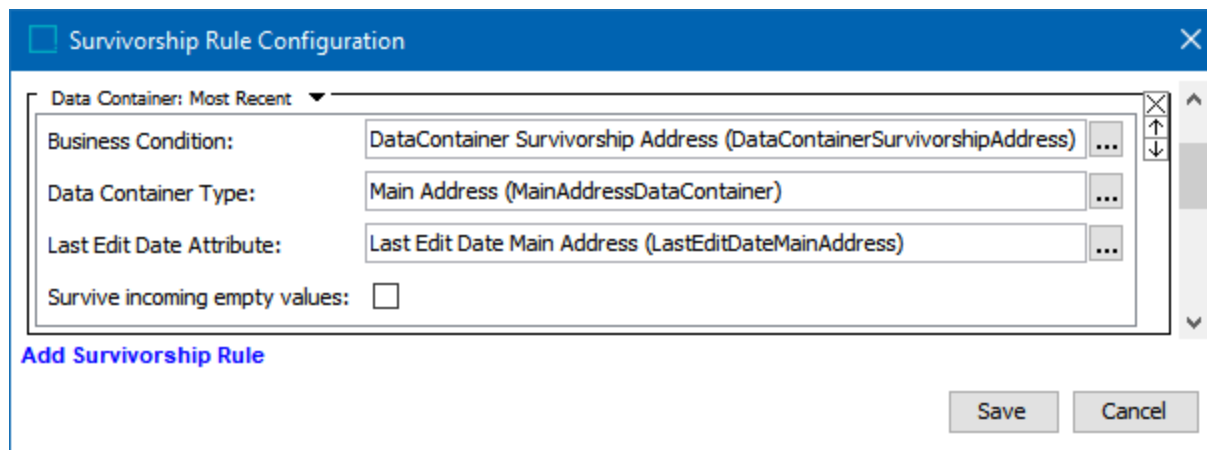
Survivorship Data Container Rules

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** - 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, see 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, see the **Data Container Keys** topic in the **System Setup / Super User Guide** 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>
```

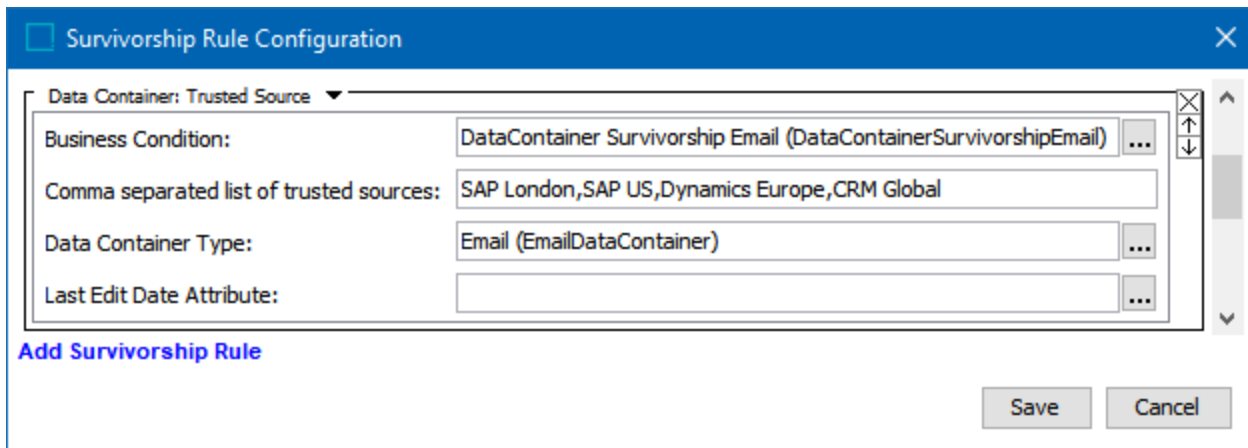
results in the following outcome based on the checkbox setting:

- 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.



- **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, see 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, see the **Data Container Keys** topic in the **System Setup / Super User Guide** 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. For information on the Source System ID Attribute setting, see 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, see the **Data Container Keys** topic in the **System Setup / Super User Guide** documentation.

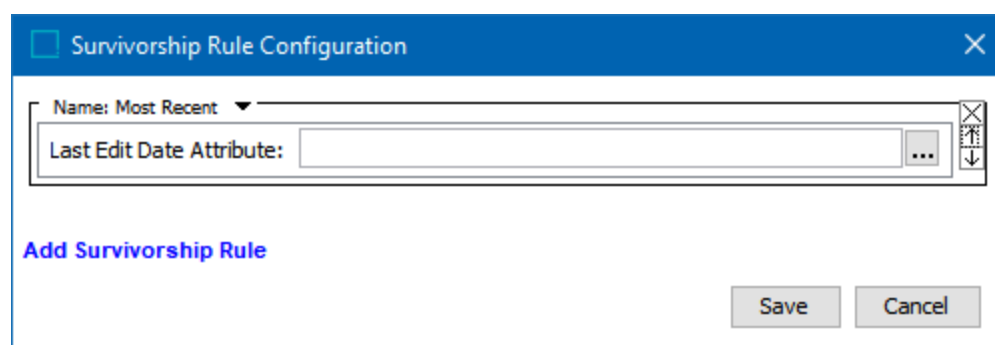
Survivorship Name Rules

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.



- **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.

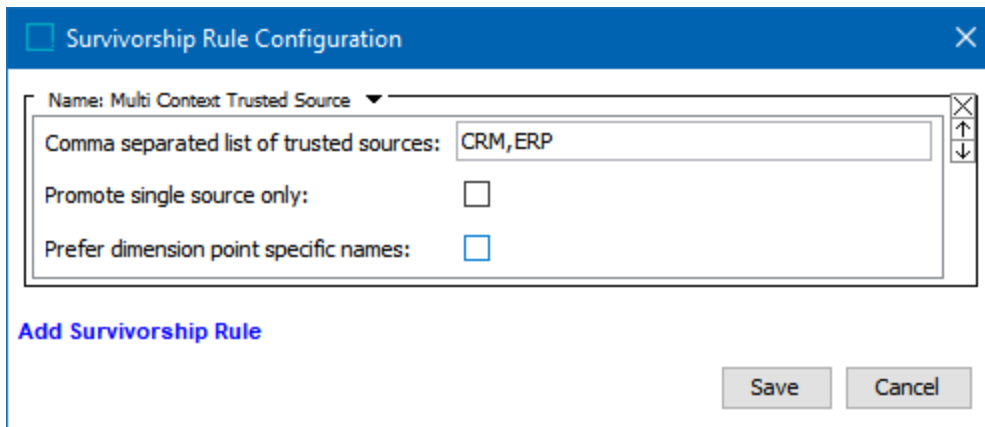
Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

- 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.



- **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. For information on the Source System ID Attribute setting, see the **Configuring the Matching - Merge Golden Record Component Model** topic.

- **Promote single source only** - When checked, the name from the most trusted source is used for all contexts / qualifiers, which prevents an empty name value in the golden record as long as one of the trusted sources has a name. For example, when only the French language, France country context has a name value, that value would be written into all other contexts.

When not checked, each context / qualifier supplies its own name, including empty values, when found.

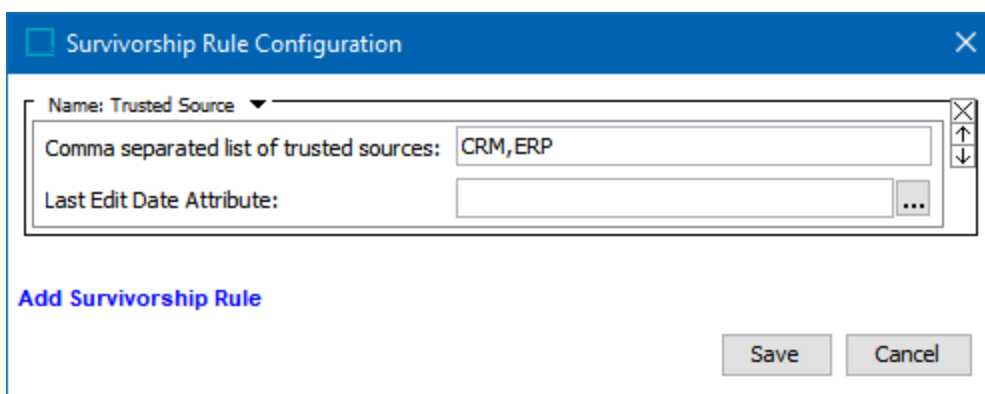
- **Prefer dimension point specific names** - When checked, only a local name is promoted.

When not checked, available inherited content is promoted if a local name does not exist.

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.



- **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. For information on the Source System ID Attribute setting, see 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.

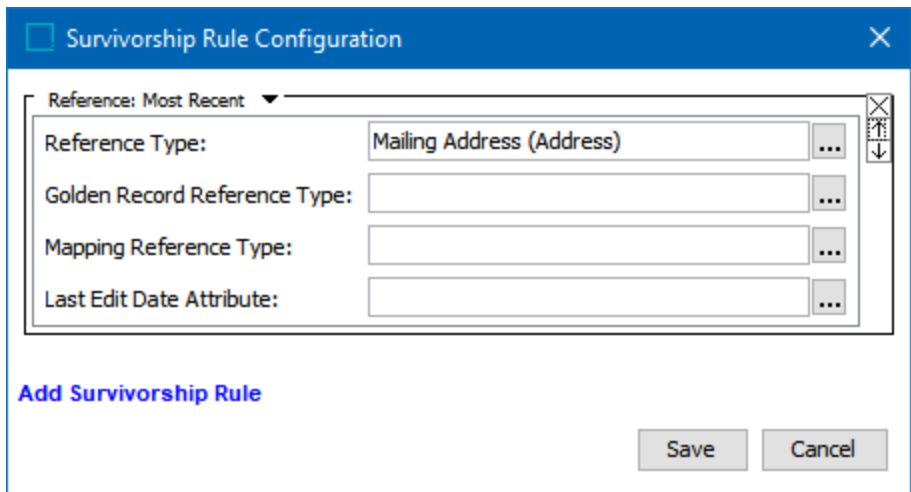
Survivorship Reference Rules

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** - Click the ellipsis button (...) to specify the valid reference / link type from the source objects you are handling.

This parameter is required. When this is the only reference type selected, a reference / link of the same type pointing to the same target is 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.

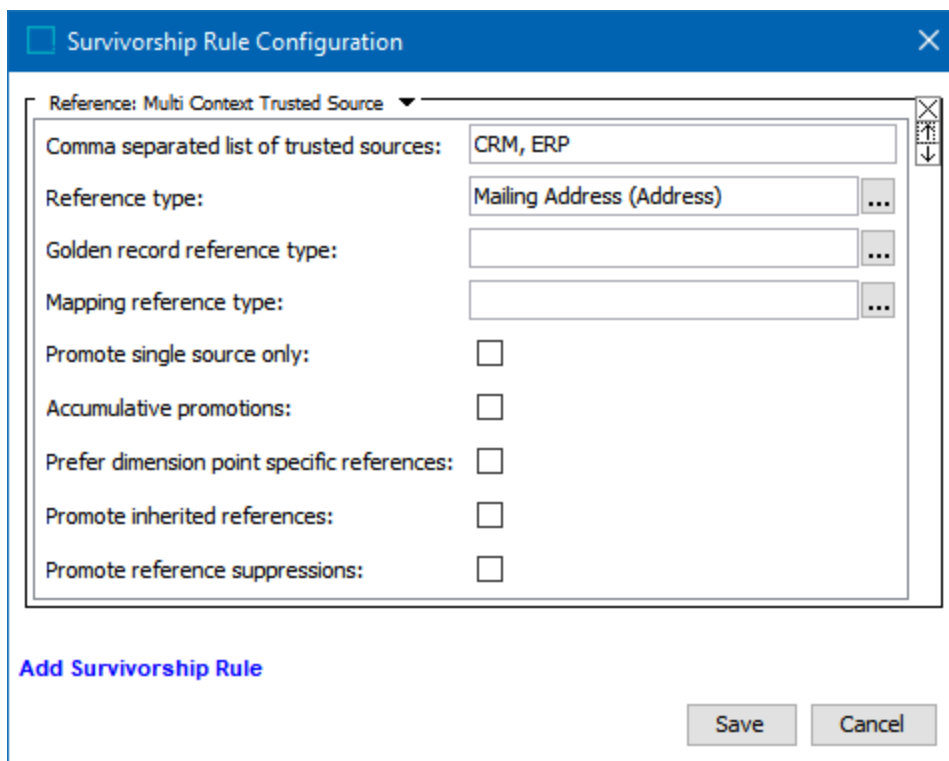
Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

- 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.



- **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. For information on the Source System ID Attribute setting, see the **Configuring the Matching - Merge Golden Record Component Model** topic.

- **Reference type** - Click the ellipsis button (...) to specify the valid reference / link type from the source objects you are handling.

This parameter is required. When this is the only reference type selected, a reference / link of the same type pointing to the same target is 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 multi-valued references / links and their metadata from multiple source records are written to the golden record.

When not checked, only all multi-valued references / links and their metadata from the most trusted source records are written to the golden record.

This option should not be used when a single-valued reference / link type is selected.

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 exist.

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.

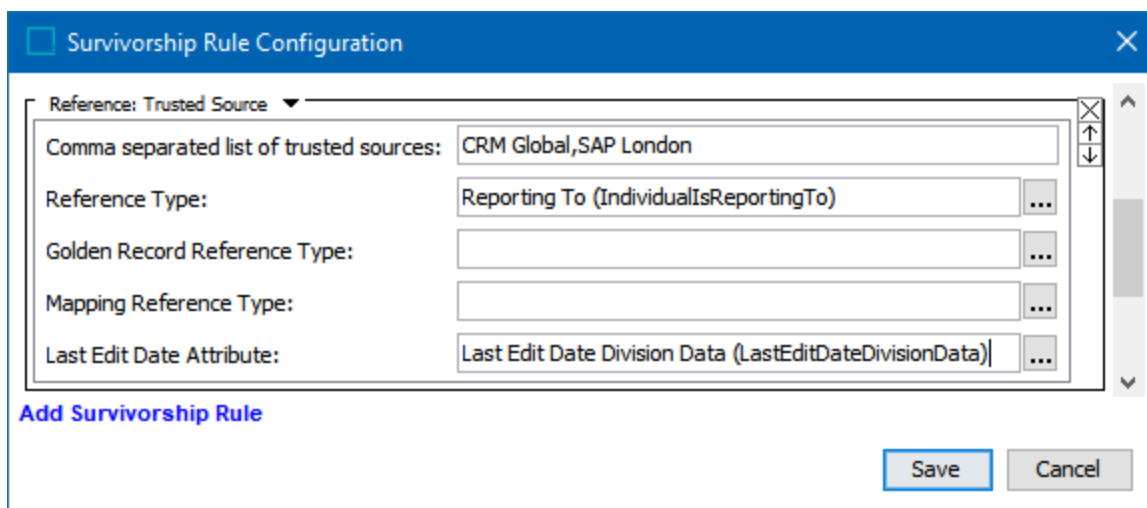
- **Promote reference suppressions** - When checked, suppressed references / links are written to the golden record.

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.



- **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. For information on the Source System ID Attribute setting, see the **Configuring the Matching - Merge Golden Record Component Model** topic.
- **Reference Type:** Click the ellipsis button (...) to specify the valid reference / link type from the source objects you are handling.
This parameter is required. When this is the only parameter populated, a reference / link of the same type pointing to the same target is 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.

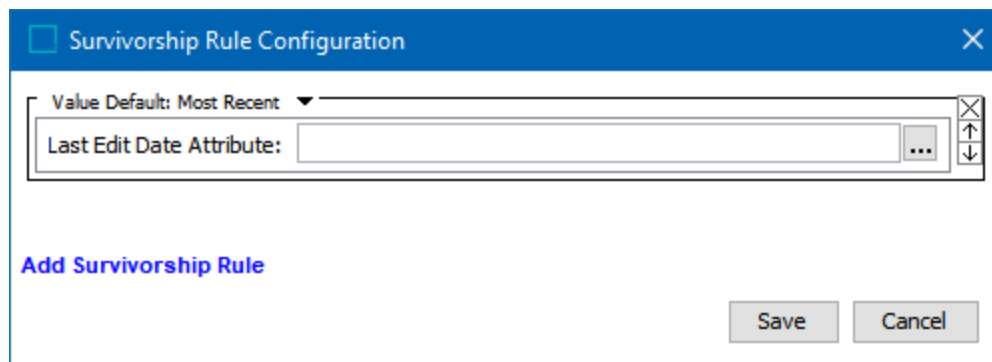
Survivorship Value Rules

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.



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.

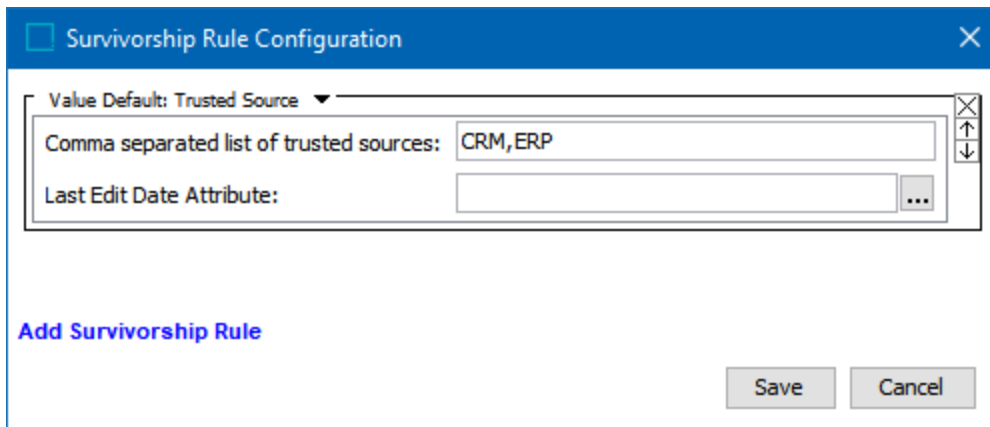
Optionally, click the ellipsis button (...) and select the attribute that holds the value to be used as the last edit date when determining the most recent source record to promote to the golden record.

- 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.

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.

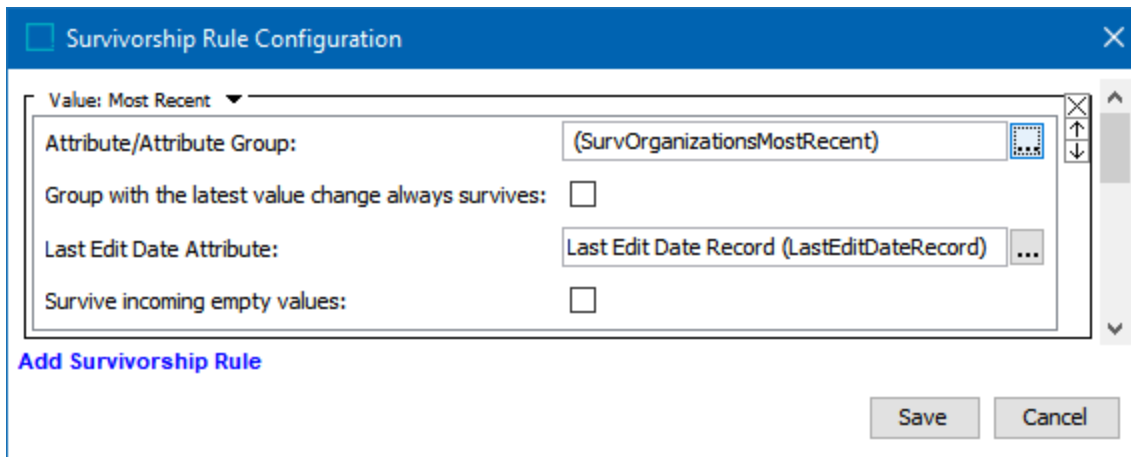


- **Comma separated list of trusted sources** - Enter a comma-separated list of the case-sensitive Source System ID for all trusted sources, starting with the most trusted source, then the next-most, and so on. Content is taken from the first trusted source with data. If content does not exist for any of the trusted sources, nothing is promoted to the golden record. For information on the Source System ID Attribute setting, see 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.

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** - 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>
```

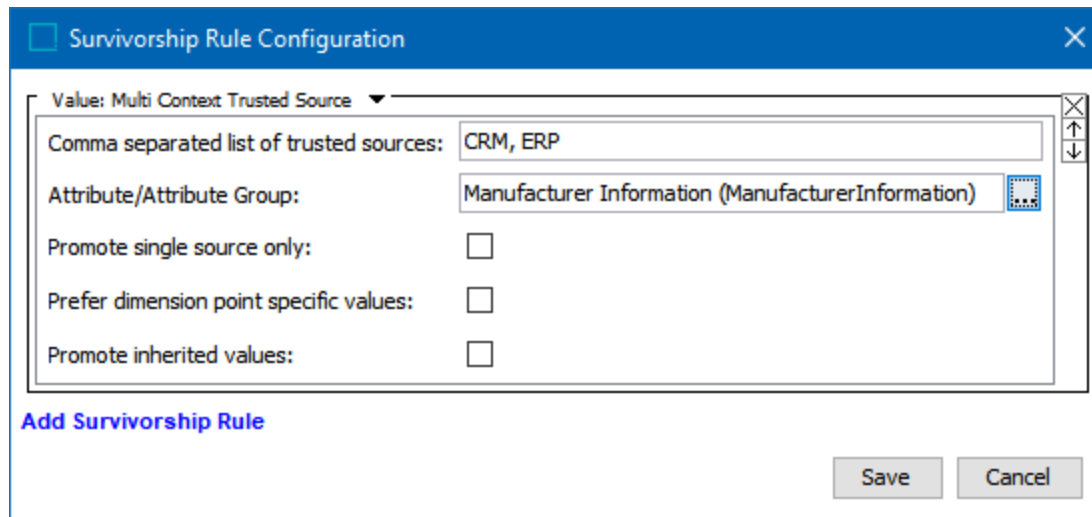
results in the following outcome based on the checkbox setting:

- 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.



- 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. For information on the Source System ID Attribute setting, see 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.

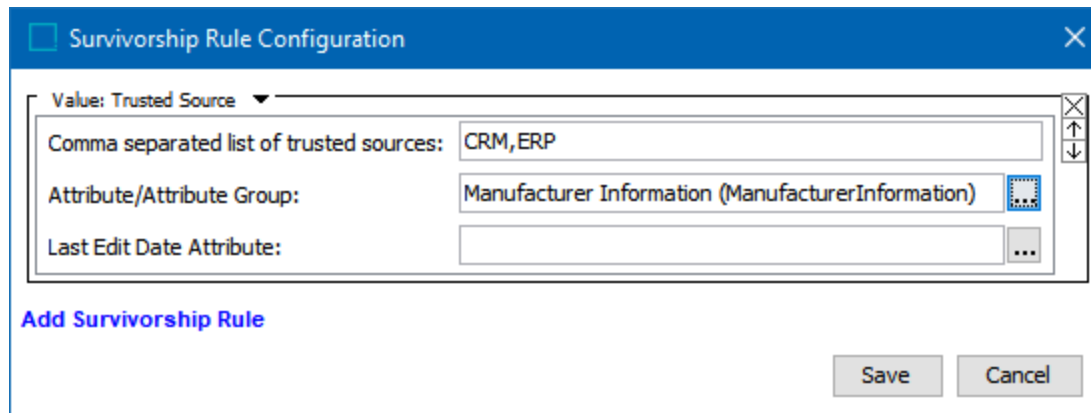
- **Promote inherited values** - When checked, inherited values are written to the golden record for the selected attribute / attribute group only if the golden record object type is valid.

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.



- **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. For information on the Source System ID Attribute setting, see 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**.

The screenshot displays the 'Individual Customer Matching Algorithm - Matching Algorithm' configuration interface. The 'Match Action Configuration' section is active, showing various thresholds and workflows. A red circle '1' highlights the 'Match Action Configuration' checkbox. A red circle '2' highlights the 'Merge Golden Record' dropdown menu. A red circle '3' highlights the 'Select Action' dialog box, which is open and shows a list of actions. The 'Merge Keep First Handler' is selected in the list. A red arrow points to the 'Merge Keep First Handler' in the list. The 'Merge Keep First Handler' is also highlighted with a red box in the main configuration window.

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. See 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. See the **Secondary Object Bind** topic in the online help **Resource Materials** documentation.

For information about other handlers, see 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

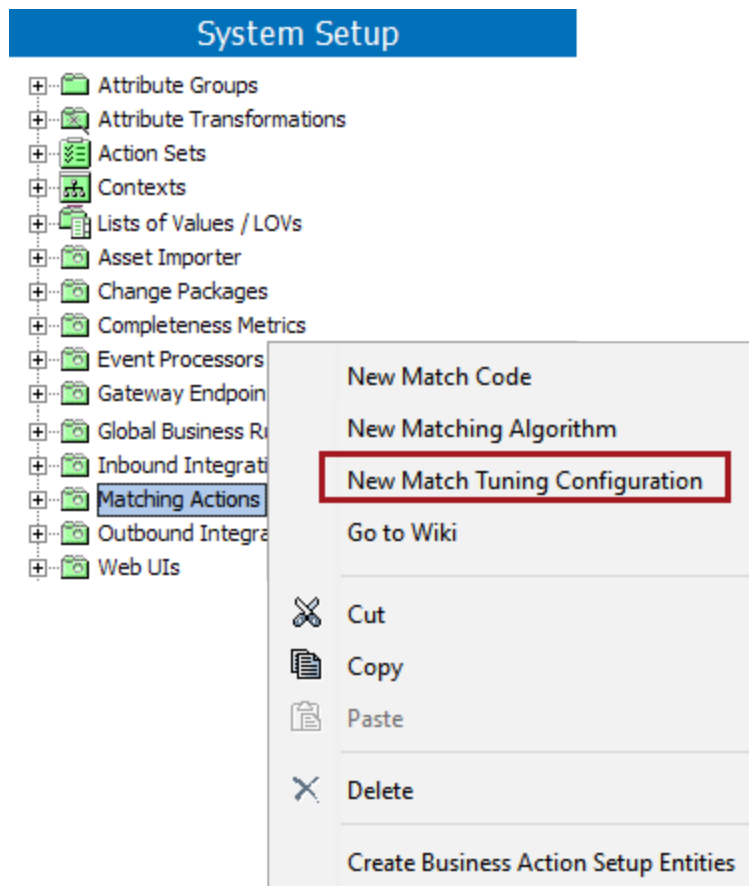
Review and improve a matching algorithm's effectiveness using the legacy functionality:

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.

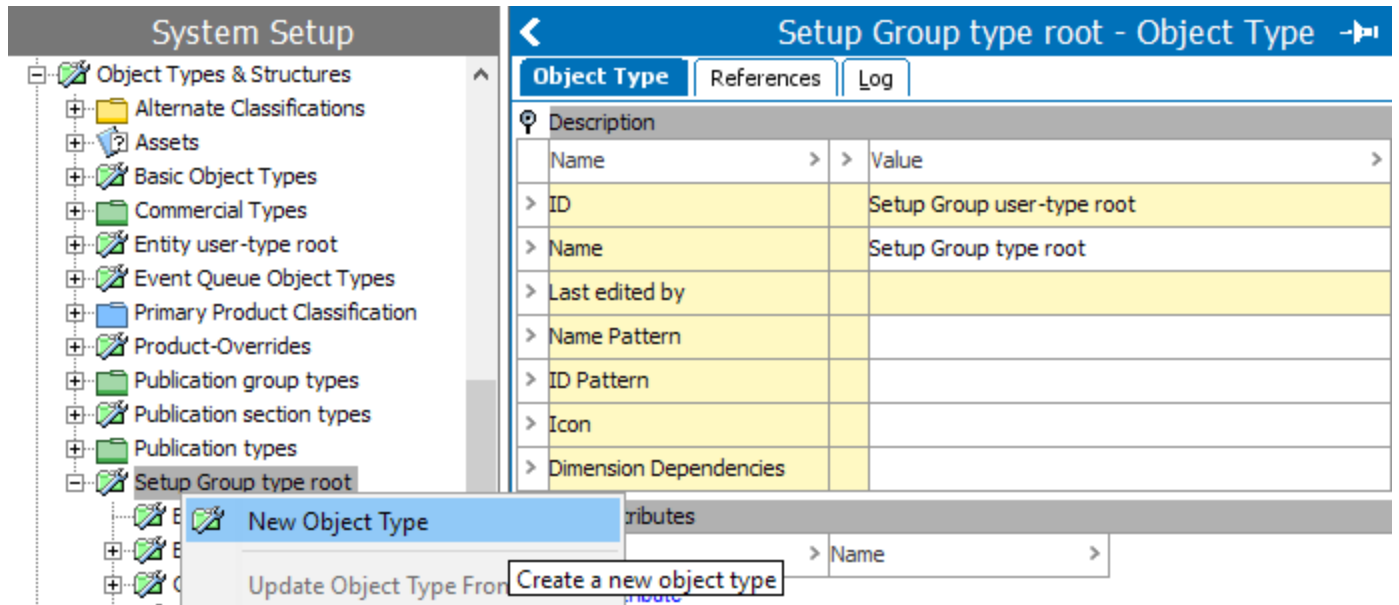
1. Create setup group type for match tuning.
2. Link match tuning object types to setup group types.
3. Create a match tuning setup group.

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.



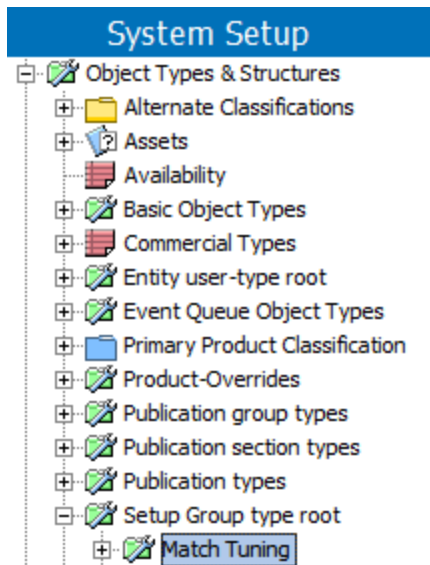
3. Click **New Object Type**, and the Create Object Type dialog will display.

The 'Create Object Type' dialog box is shown. It has the following fields and options:

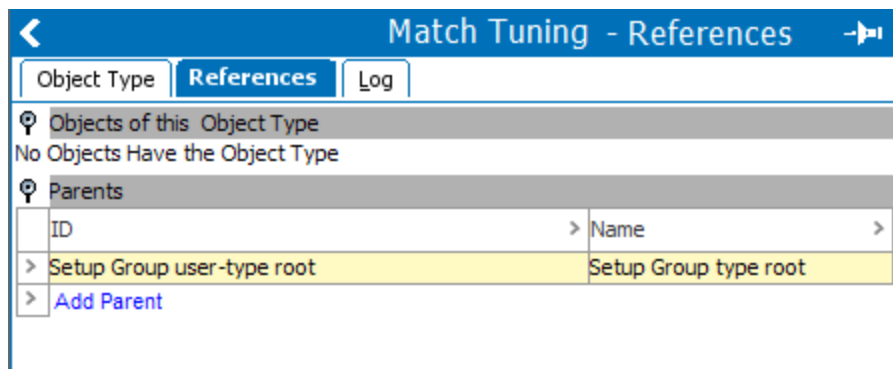
- ID: MatchTuning
- Name: Match Tuning
- Dimension Dependency:
 - Country
 - Language
- Buttons: Create, Cancel

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.



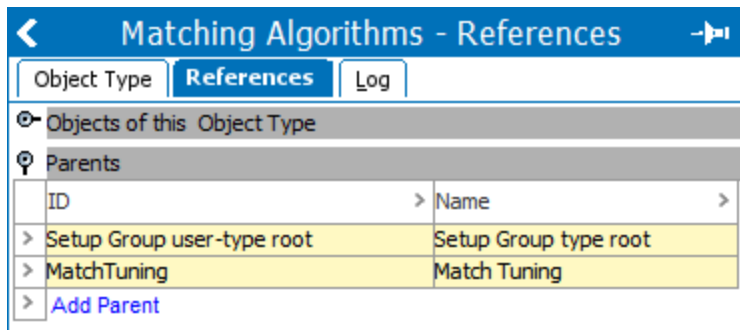
7. Select the newly added Setup Group type > References tab > open the Parents flipper.



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.

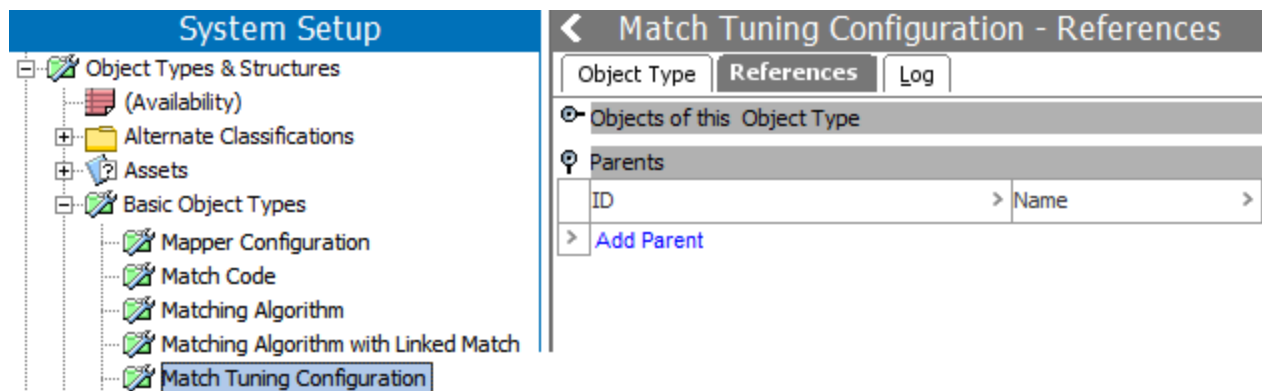
The dialog closes, and the newly created setup group type (i.e., Match Tuning) is listed as a parent along with the Setup group user-type root.



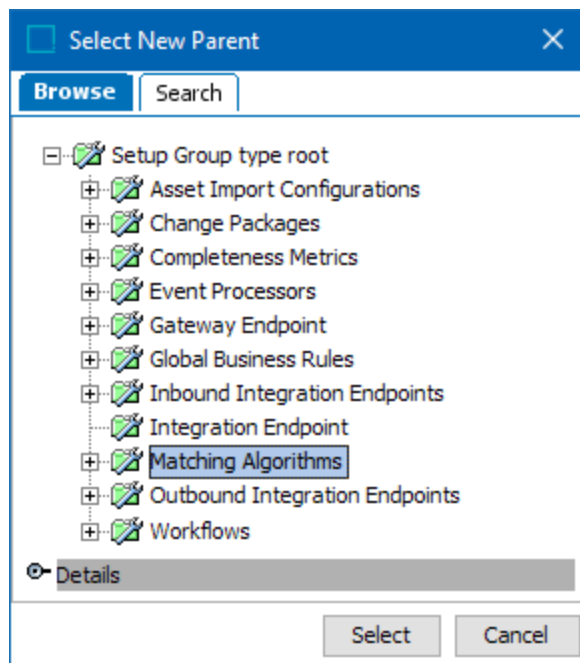
Link Match Tuning Object Types to Setup Group Types

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.



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

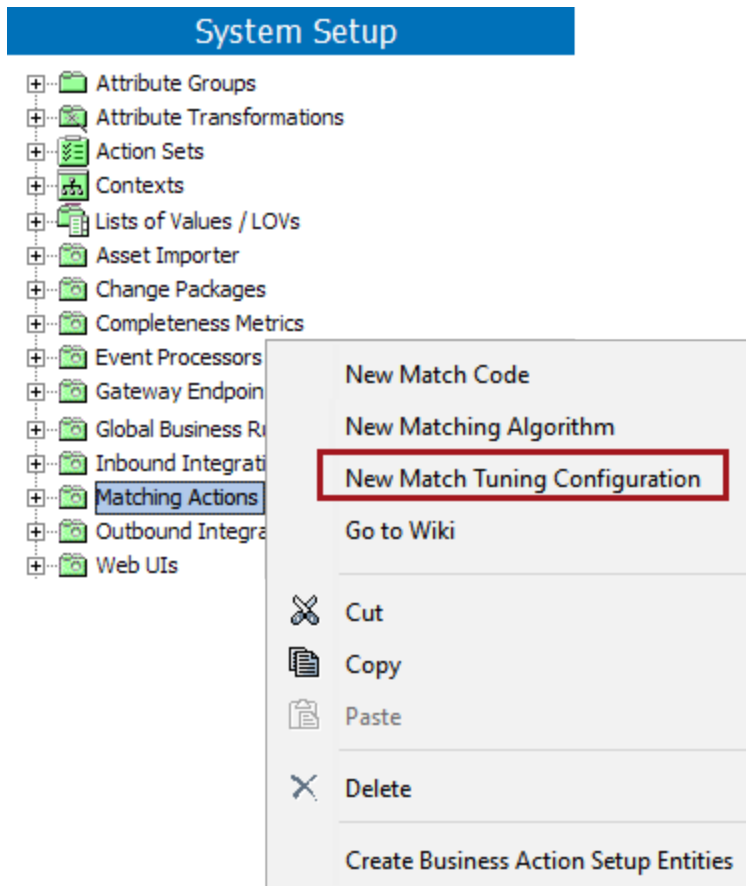
- Change Packages
- Completeness Metrics
- Event Processors
- Gateway Endpoint
- Global Business Rules
- Inbound Integration Endpoints
- MatchingActions
- Outbound Integration Endpoints
- RemoveAttributesFromGroup
- Status Flags
- Web UIs
- Workflow Profiles
- Workflows

ID:

Name:

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, see the **Match Tuning** topic.

Prerequisites

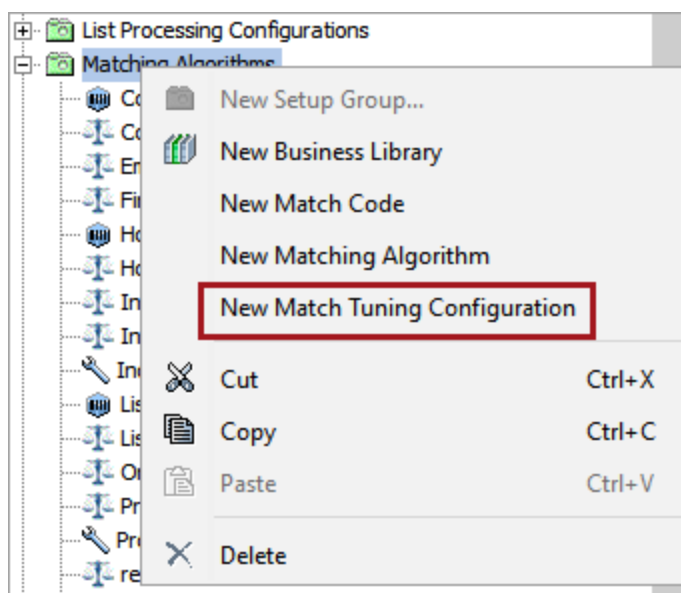
1. Ensure that the initial setup is complete, as defined in the **Initial Setup for Match Tuning** topic.
2. Open the Matching component model and view the 'Match Tuning Asset Object Types' parameter to identify the asset object types used to store the profile data for match tuning. If none are selected, see 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, see the **Reference Target Lock Policy on Object Types** topic of the **System Setup / Super User Guide** 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, see the **Data Governance** topic in the **Customer MDM Solution Enablement** documentation.

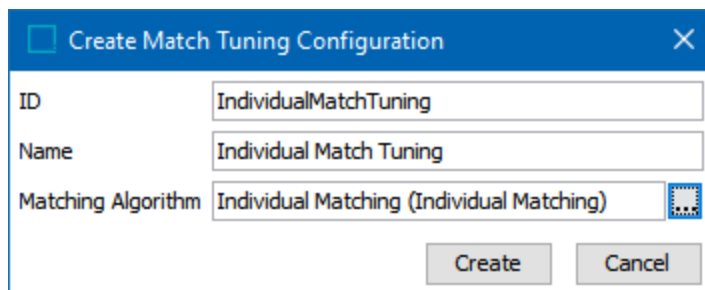
Configuration

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.



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

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

Upload Tuning Data | Generate/Update Data Profile | Evaluate Matching Algorithm

Configuration Validation Status

Specified Data

Data file(s)	00 SampleIndividuals
Data file root	Sample Import Data
Pre-processor	Transformation by Import Configuration

[Edit Data Specification](#)

Specified Matching Information

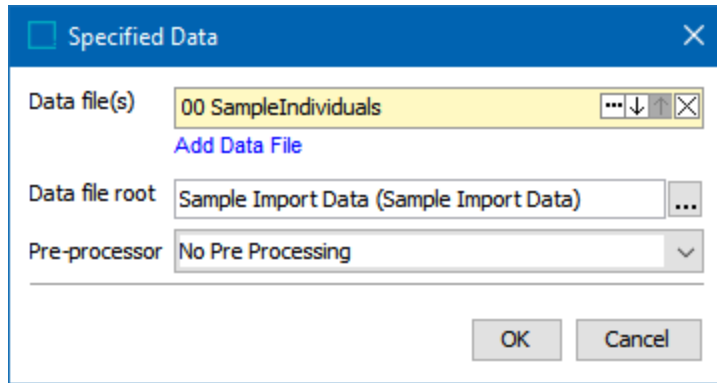
Queue for profiling	MTCProf
Number of threads used for profiling	2
Queue for matching algorithm evaluation	MTCMatch
Number of threads used for matching algorithm evaluation	2
Matching algorithm	Individual Matching
Minimum object count for match code groups	20
Maximum number of match code groups	100
Match interval to export	70 - 100 %
Pairs per percent	10
Attributes to export	
Export match details	true

[Edit Matching Information](#)

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 see the errors.

4. Open the 'Specified Data' flipper and click the **Edit Data Specification** link to display the 'Specified Data' dialog.

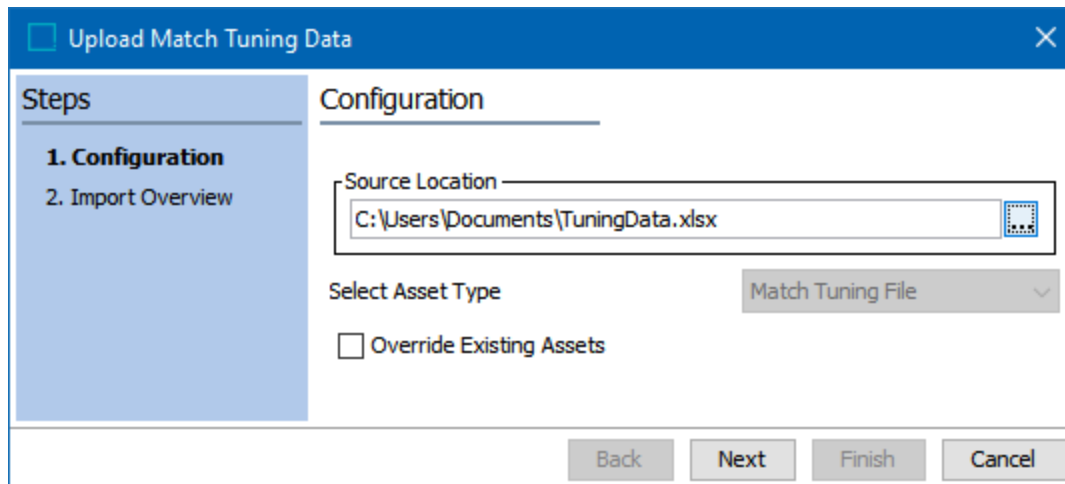


The 'Specified Data' dialog box contains the following fields:

- Data file(s):** A text field containing '00 SampleIndividuals' with an ellipsis button and a close button. Below it is a blue link labeled 'Add Data File'.
- Data file root:** A text field containing 'Sample Import Data (Sample Import Data)' with an ellipsis button.
- Pre-processor:** A dropdown menu currently set to 'No Pre Processing'.

Buttons at the bottom: OK, Cancel.

- 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, see 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.



The 'Upload Match Tuning Data' wizard has two main sections:

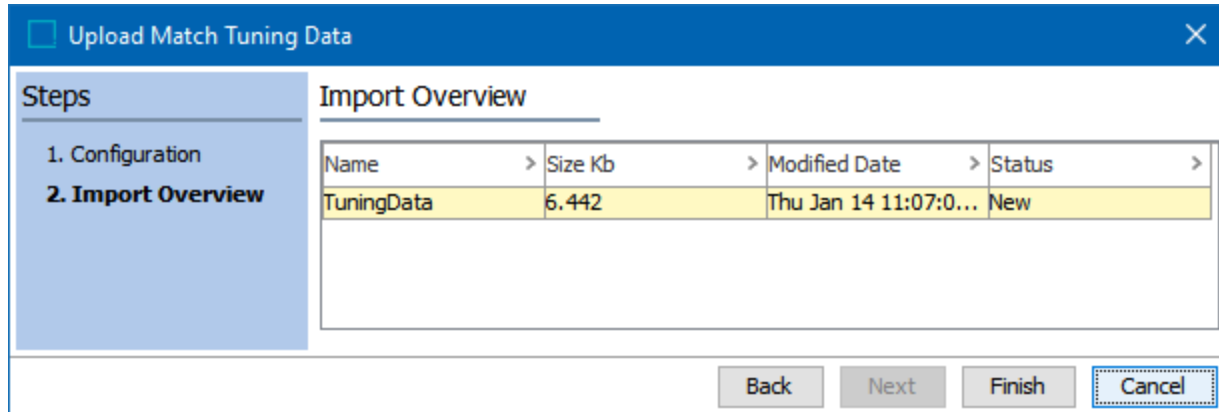
- Steps:** A list with '1. Configuration' (selected) and '2. Import Overview'.
- Configuration:**
 - Source Location:** A text field containing 'C:\Users\Documents\TuningData.xlsx' with an ellipsis button.
 - Select Asset Type:** A dropdown menu set to 'Match Tuning File'.
 - Override Existing Assets:** An unchecked checkbox.

Buttons at the bottom: Back, Next, Finish, Cancel.

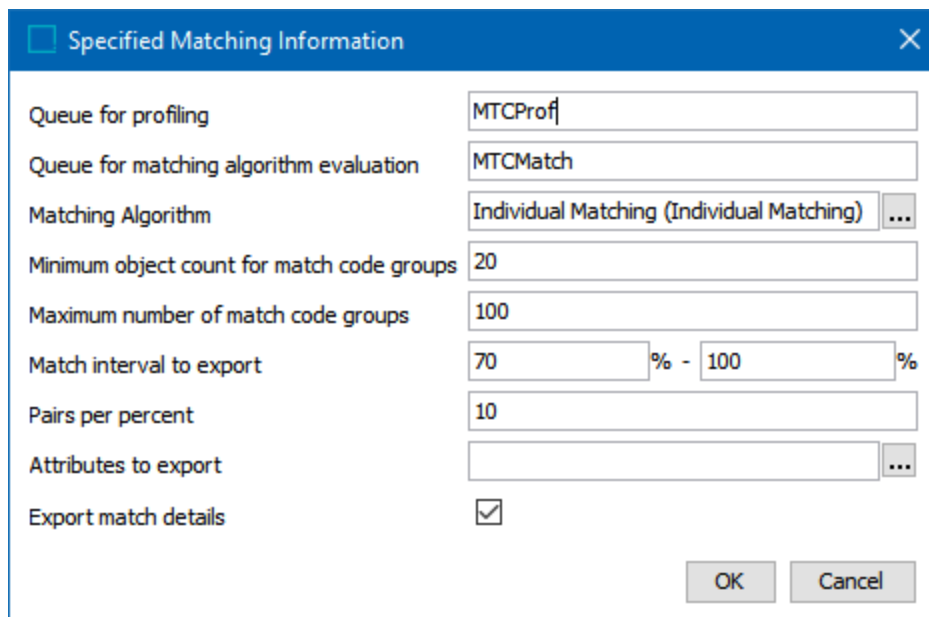
The sample data used for tuning is uploaded as follows:

- 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.



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 see 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:



- **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.

Individual Matching - Matching Algorithm

Matching Algorithm | Match Criteria | Match Code Values | **Match Result**

Pair Export
 Pair Export Confirmed
 Pair Import Confirmed

Showing page 1 Sort Ascending [Add Additional Matching Algorithm Column](#)

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 Yu	Cathy You	Tue Dec 08 07:08:35 EST 2020	50
> Catherine Yu	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, see 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:

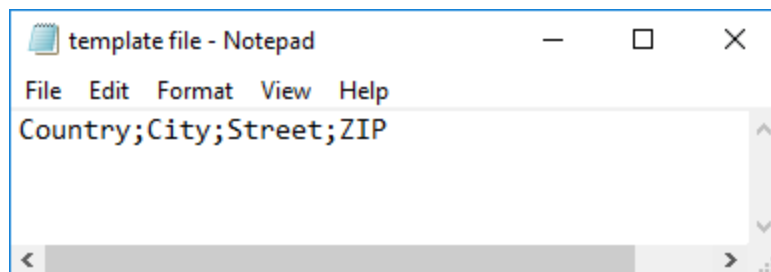
- **<Pair>** - One row per source object and the 'Pair' info is used to indicate which objects belong together. The first two rows have the value '1,' the next two rows have '2,' and so on.
- **<Match y n>** - Indicates whether pairs are matches or not. A value is only required for the first object in 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:

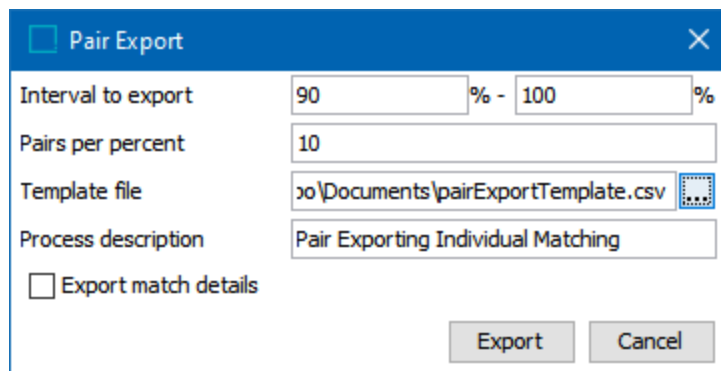
- Attribute IDs separated by semicolons (;)
- Save as CSV format



Configuration

Use the following steps to perform Pair Export.

1. Click the **Pair Export** button.
2. In the Pair Export dialog, specify the following:



- **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.

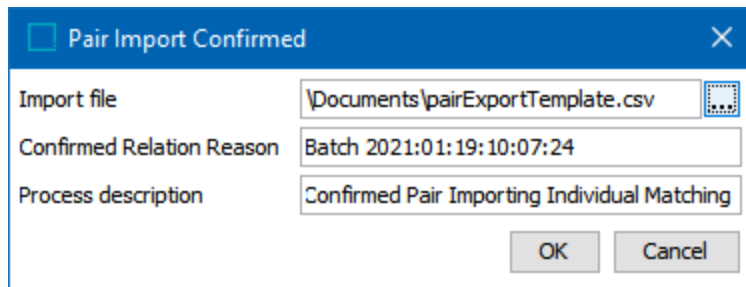
Pair Import Confirmed

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:



- **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.

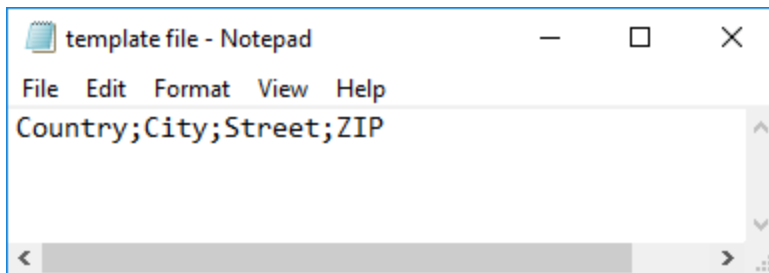
Pair Export Confirmed

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 (;)

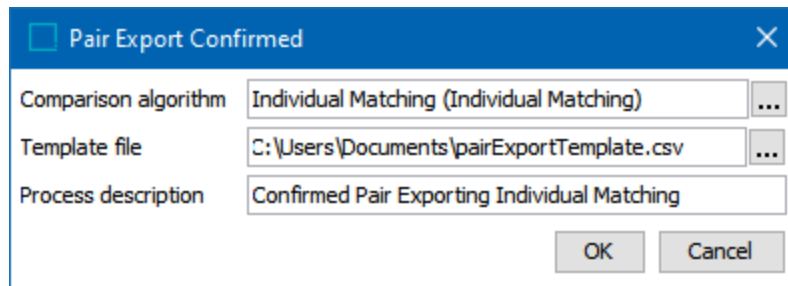
- Save as CSV format



Configuration

Use the following steps to perform Pair Export Confirmed.

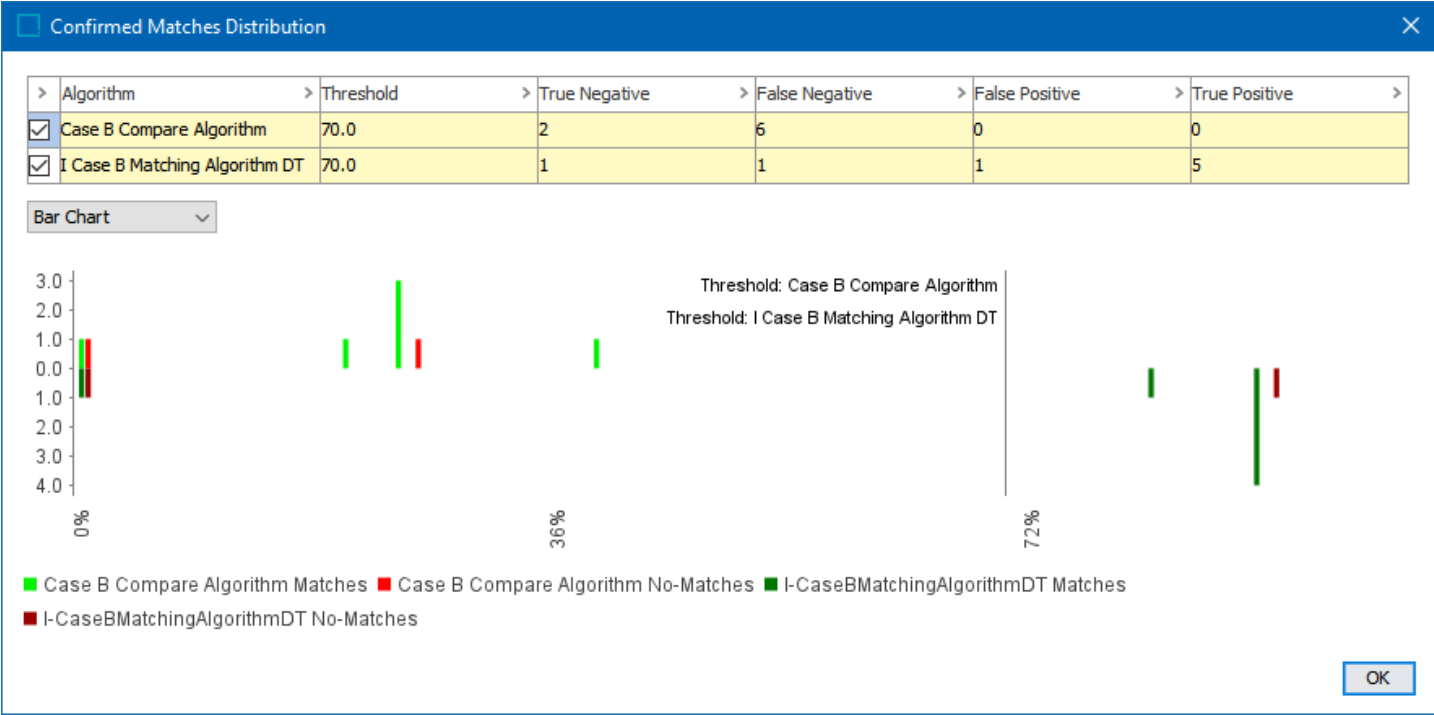
1. Click the **Pair Export Confirmed** button.
2. In the 'Pair Export Confirmed' dialog, specify the following:



- **Comparison Algorithm:** 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.



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.



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 hits, 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

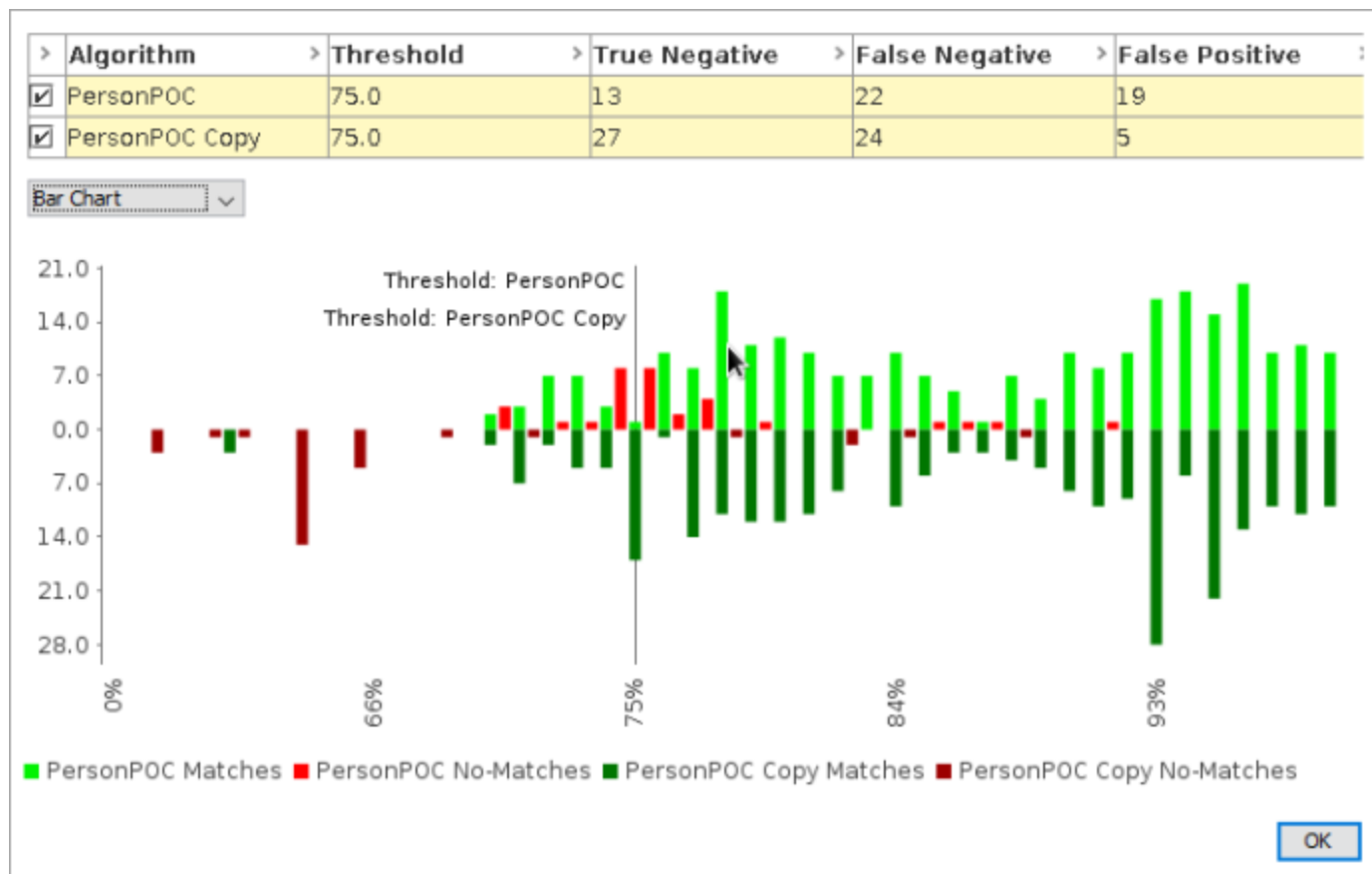
The bars in the chart show the frequency of the scores of the selected algorithm. The bar chart can either show a single algorithm or two algorithms in a special compare mode that enables a detailed comparison of the two algorithms.

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.

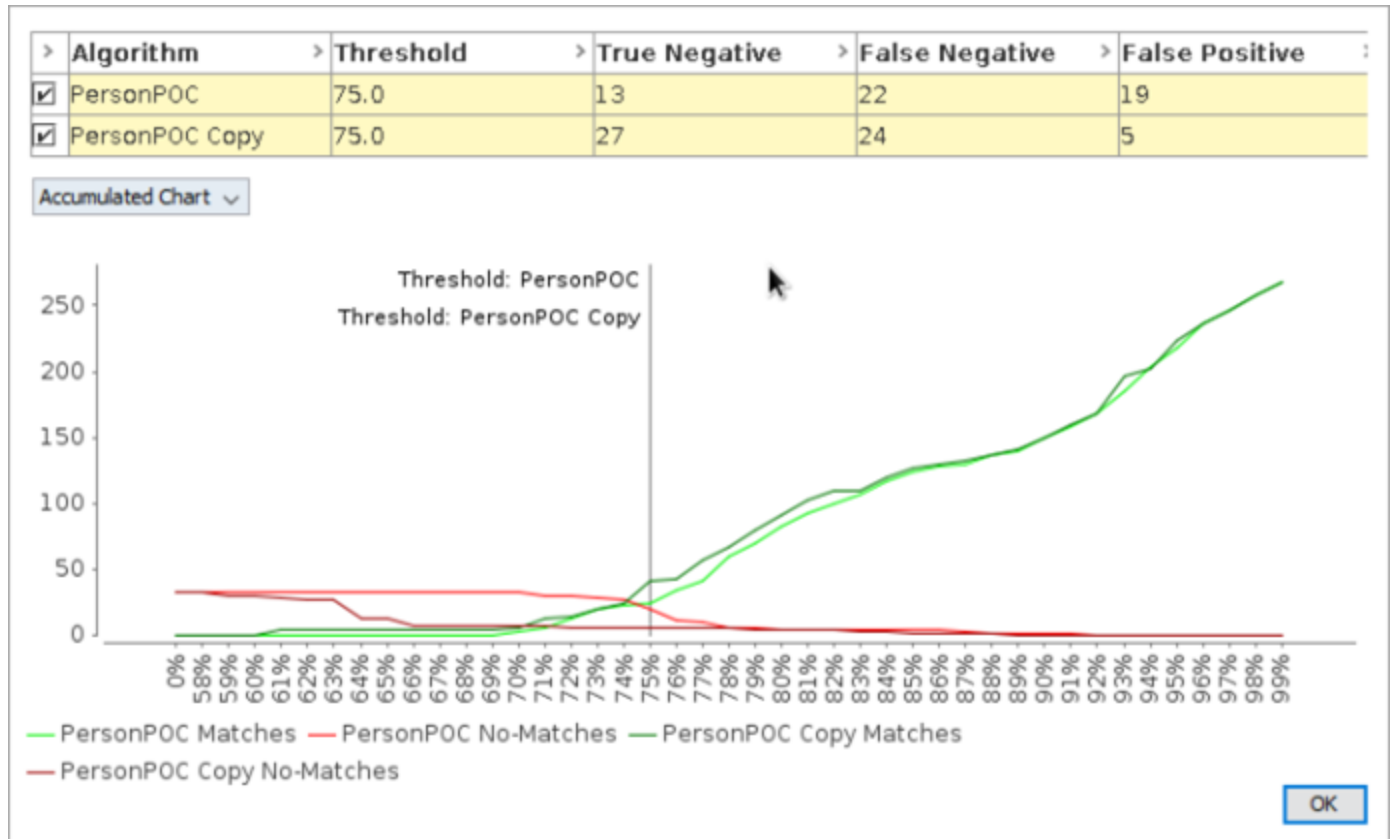
For exact numbers of false positives and false negatives, see the table. Because the bars have a resolution of 1 percent point, the exact number of false positives and false negatives are not available in the graph.

- 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.



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.



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, see 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.

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 Matching Agent

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 matching agent (MLMA) eases the workload by providing recommendations for merging or rejecting based on the data steward's decisions.

Golden Record Clerical Review Task List

Advanced Merge Merge Reassign Reject

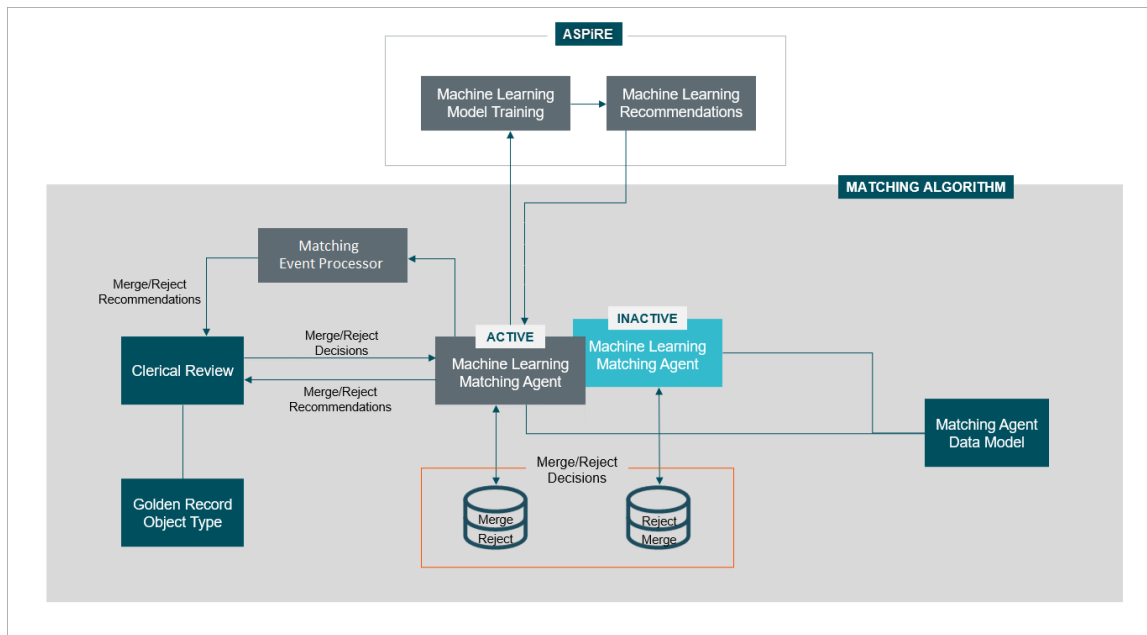
⚠ Not all potential duplicates are shown for all tasks.

Task	Golden Record	Main Address	Source Informati...	First Name	Last Name	Email	Phone
<input type="checkbox"/> Assignee: Stibo Users Created: 3/1/22 Merge	ID: 822363 • Match Score: -- Created: 2/11/22 • Updated: 2/11/22	1932 Iantana dr...	CRM Global 102...	Michael	Pierce	36mjp64@yaho...	3163975260; 5...
	ID: 824421 • Match Score: 67.5 Created: 2/11/22 • Updated: 2/11/22	1932 Ibatana dt...	Dynamics Europ...	Mixshel	Pirece	37mjp64@yaho...	3751059728; 3...
<input type="checkbox"/> Assignee: Stibo Users Created: 3/1/22 Reject	ID: 819936 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	19 Overlook Rid...	SAP London 16...	Cathy	Miller	GustavoBarrosC...	(593) 145-3181;...
	ID: 820273 • Match Score: 83.84 Created: 2/2/22 • Updated: 2/9/22	19 Overlook Rd...	Dynamics Europ...	Kathy	Miller	eros.nec@Morbi...	(604) 658-0190;...
<input type="checkbox"/> Assignee: Stibo Users Created: 3/1/22 Merge	ID: 820088 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	36 Garden St Sh...	SAP US 161705...	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...	SAP US 165901...	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...	Dynamics Europ...	Sheley	Fullgum	non.ante@seddi...	(216) 435-0544;...
<input type="checkbox"/> Assignee: Stibo Users Created: 3/1/22 Reject	ID: 823371 • Match Score: -- Created: 2/11/22 • Updated: 2/11/22	4916elizabether...	Dynamics Europ...	Timothy	Price	TimothyPrice91...	9198780765; 8...
	ID: 825328 • Match Score: 50 Created: 2/11/22 • Updated: 2/11/22	4916elizaberhdi...	SAP London 10...	Tjmtohy	Priec	4chldmroperti...	3011505334; 2...

Total Number of Tasks: 63; Selected items: 0

The machine learning engine is part of Stibo Systems' ASPIRE cloud environment. The MLMA analyzes and learns from the data steward's decisions within the clerical review task list. After the data steward meets the threshold of decisions, the matching agent provides merge and reject recommendations within the clerical review task list, which the data steward can either heed or disregard.

Solution Overview



The matching event processor updates new tasks 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 MLMA functionality is in the ramp-up phase and is only accessible through an early adopter program. As part of this program, Stibo Systems will provide a client ID and passphrase, which are needed to configure the authentication REST gateway. To learn more about the ramp-up phase / status, see the **License and Component Lifecycle** topic in the **System Release and Patch Notes** section of online help. To participate in the early adopter program, send an email request to SYSCLericalReviewMatchingAgent@StiboSystems.com.

The following topics outline the setup and function of the MLMA:

1. Configuring MLMA
2. Maintaining the MLMA Data Model
3. Matching Agents
4. Adding MLMA Recommendations to a Clerical Review Task

ASPiRE License

ASPiRE uses the Dedupe library with an **MIT** license.

Configuring MLMA

In order to utilize the machine learning matching agent (MLMA), you must first configure several elements. The MLMA uses REST gateways to communicate between a matching algorithm and the cloud on which the MLMA works.

Prerequisite

Implement a match and merge solution as defined in the **Configuring Match and Merge** topic.

Configuration

The following topics outline the configuration of the MLMA:

1. Configuring the MLMA REST Gateways
2. Configuring the Matching Agent Object Type
3. Configuring the Clerical Review Workflow for MLMA
4. Adding MLMA Recommendations to a Clerical Review Task

Configuring the MLMA REST Gateways

A REST authentication gateway and a service gateway are necessary for the machine learning matching agent (MLMA) to function. Configuration includes:

- Stibo Systems Cloud Authentication Gateway and Authentication Business Function
To access the cloud, MLMA uses a token-based authentication on the authentication gateway.
- Stibo Systems Cloud Service REST Gateway
Access to the machine learning cloud goes through this gateway. The security tokens needed to authenticate are provided by a business function as described in the **Configure Authentication Business Function** section in this topic.

Configure Stibo Systems Cloud Authentication Gateway

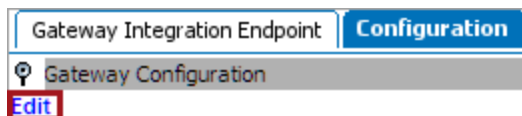
For general information on how to configure a REST gateway, see the **Configuring a Gateway Integration Endpoint - REST** topic in the **Data Exchange** documentation.

1. In the sharedconfig.properties file on the STEP application server, add the case-sensitive property with the URL shown below:

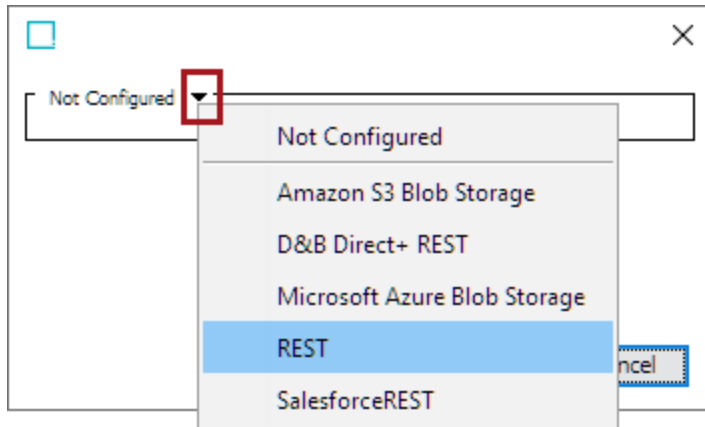
```
RESTGateway.ServerURL=StiboCloudURL=https://app.stibosystems.com
```

Changes to the properties file, outlined above, are implemented when the server is restarted.

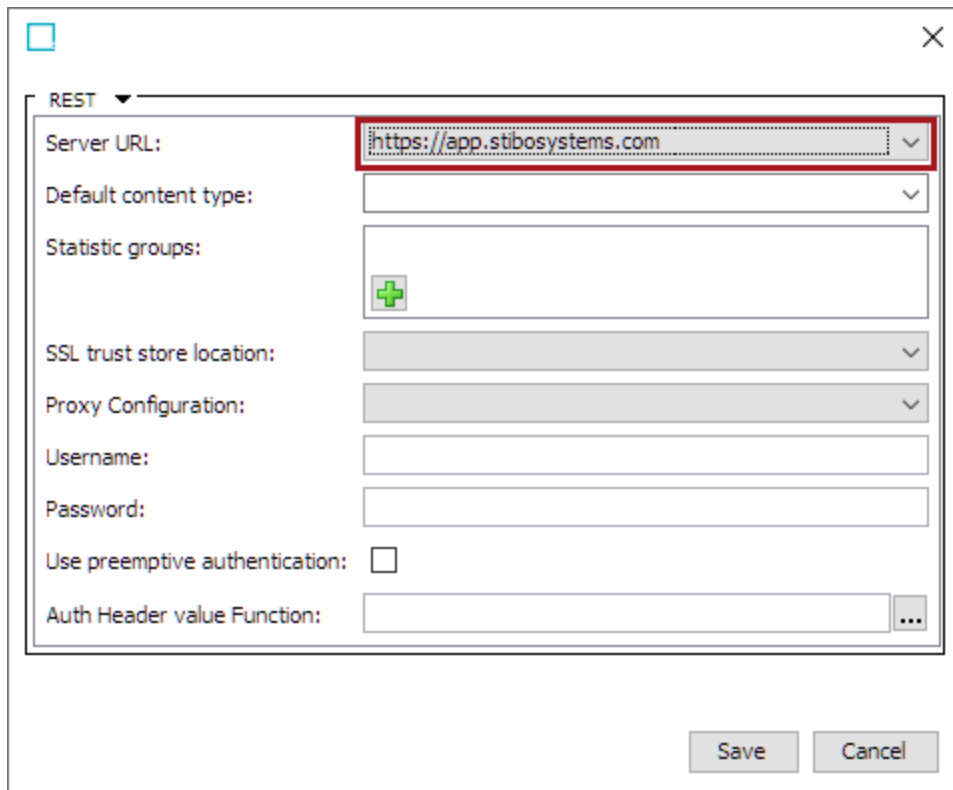
2. Create a gateway integration endpoint, as described in the **Gateway Integration Endpoints** topic in the **Data Exchange** documentation.
3. Under the **Configurations** tab of the gateway integration endpoint, navigate to **Gateway Configuration** and click the **Edit** link.



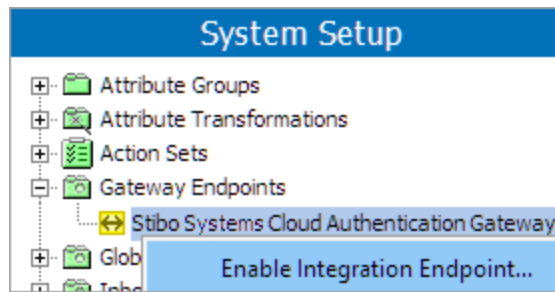
4. In the dialog, click the dropdown arrow and select 'REST.'



5. Under **Server URL**, select the URL you just created.



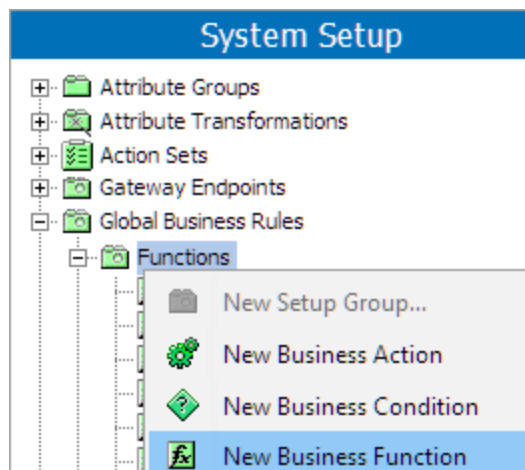
6. On this authentication gateway, leave the **Auth Header value Function** parameter empty.
7. Click **Save**.
8. Right-click the gateway and select 'Enable Integration Endpoint.'



Configure Authentication Business Function

For general information about business functions, see the **Business Functions** topic in the **Business Rules** documentation.

1. In System Setup, locate the node which houses business functions. Right-click and select 'New Business Function.'



2. Navigate to the bottom of the screen and click **Edit Business Function**.

Matching Agent Auth rev.0.1 - Business Function

Business Function		Usage	Log	Status
Name	>	Value		
ID	>	MatchingAgentAuth		
Name	>	Matching Agent Auth		
Revision	>	0.1 Last edited on Thu Sep 23 15:57:08 CEST 2021		
Description	>			
Type	>	Function		
Run as privileged	>	<input type="checkbox"/>		

Function Dependencies

JavaScript Function: Bindings, 0 messages, Parameter, String,

[Edit Business Function](#)

- On the Business Function Editor dialog, click the **Edit** button on the JavaScript function to open the Edit Operation dialog.

Business Function Editor - Matching Agent Authentication

ID

Name

Description

Type

Run as privileged

Function Depend

JavaScript

Edit Operation

JavaScript Function

Bindings:

Variable name	Binds to	Parameter
gateway	Gateway Integration Endpoint	Stibo Cloud Authentication Gateway
secret	Secret

Messages:

Parameters:

Return Type:

Return Type

Map<String,String>

JavaScript:

```

1 var body = {
2   grant_type: 'client_credentials',
3   client_id: '<client ID>',
4   client_secret: secret
5 }
6 var request = gateway.post().path('/base/account/v1/token').urlEncodedBody(body);
7 var response = request.invoke();
8 var token = JSON.parse(response + '').access_token;
9 return java.util.Collections.singletonMap('Authorization', 'Bearer ' + token);

```

[Edit externally](#)

Save Test JavaScript Cancel

- Add a gateway bind using the authentication gateway you just created. For more information on gateway binds, see the **JavaScript Binds** topic in the **Resource Materials** documentation.
- Add a secret bind and type in the passphrase received as part of the early adopter program. For more information on secret binds, see the **Other Binds** topic in the **Resource Materials** documentation.
- On the **Return Type** parameter, click the **Edit** button and set Return Type to: Map<String, String>
- Insert JavaScript below:

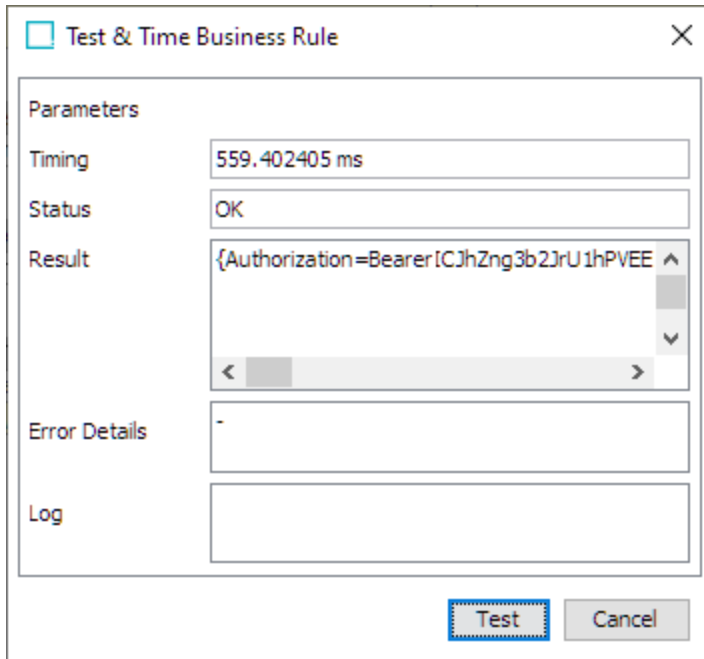
```

1      var body = {
2          grant_type: 'client_credentials',
3          client_id: '<client ID>',
4          client_secret: secret
5      }
6      var request = gateway.post().path
7      ('/base/account/v1/token').urlEncodedBody(body);
8      var response = request.invoke();
9      var token = JSON.parse(response + '').access_token;
      return java.util.Collections.singletonMap('Authorization',
      'Bearer ' + token);

```

Important: Ensure the correct ID is entered in the `client_id: '<client ID>'` line as received as part of the early adopter program

- Click the **Test JavaScript** button to open the Test & Time Business Rule dialog and click the **Test** button. If the test returns Status 'OK,' the connection and your credentials work.

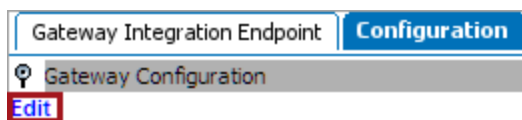


- Click **Save** on the Edit Operation dialog.

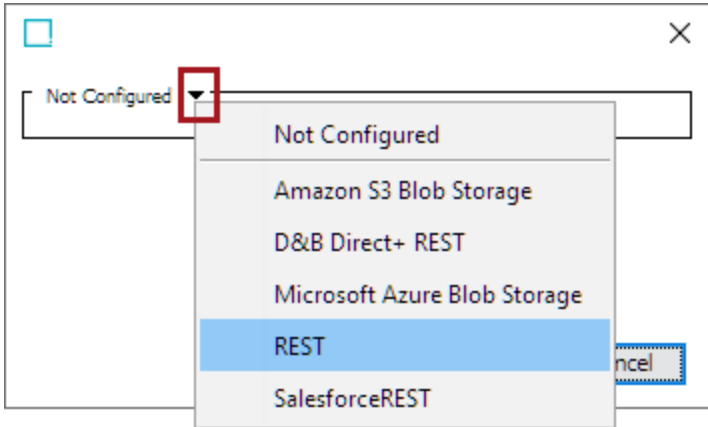
Configure Stibo Systems Cloud Service REST Gateway

For general information on how to configure a REST gateway, see the **Configuring a Gateway Integration Endpoint - REST** topic in the **Data Exchange** documentation.

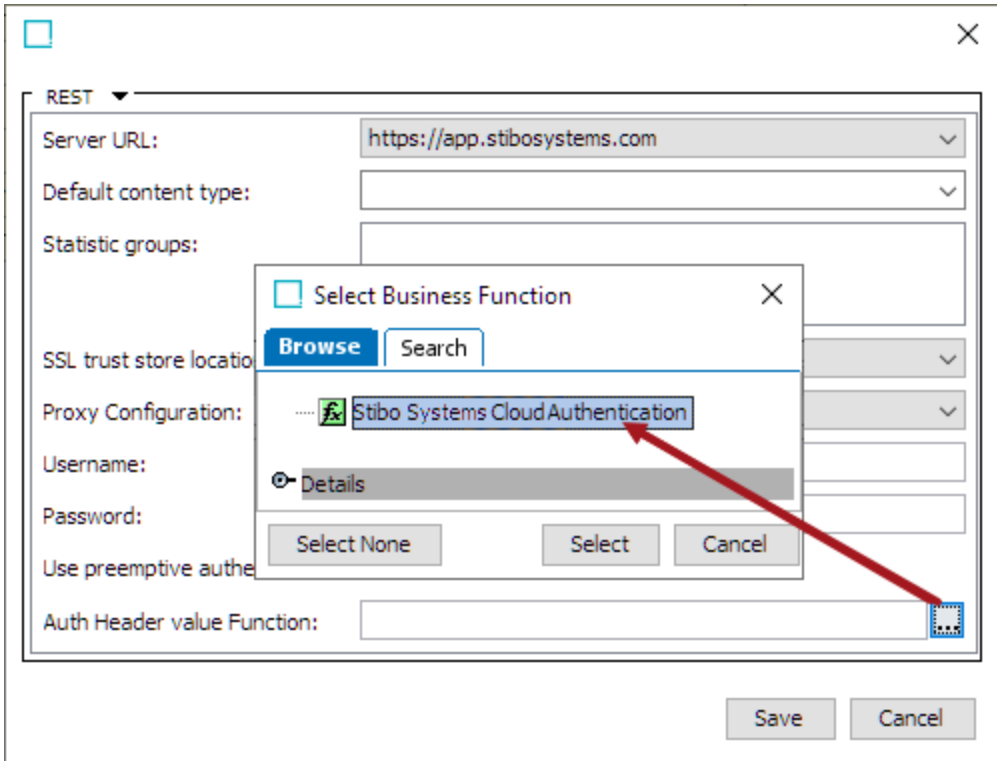
1. Create a gateway integration endpoint, as described in the **Gateway Integration Endpoints** topic in the **Data Exchange** documentation.
2. Under the **Configurations** tab of the gateway integration endpoint, navigate to **Gateway Configuration** and click the **Edit** link.



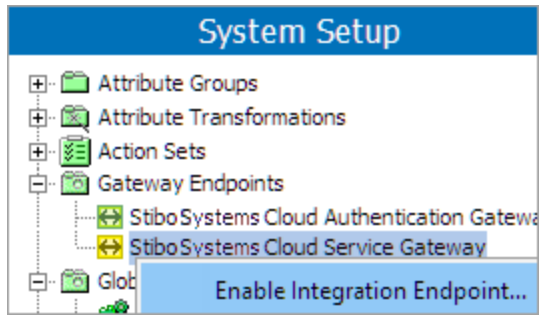
3. In the new dialog, click the dropdown arrow and select 'REST.'



- 4. Under **Server URL**, select the URL created in the first step of the **Create Authentication Gateway** section.
- 5. In the same dialog, click the ellipsis button (...) of **Auth Header value Function** and select the business function created in the **Configure Authentication Business Function** section.



- 6. Click **Save**.
- 7. Right-click the gateway and select 'Enable Integration Endpoint.'

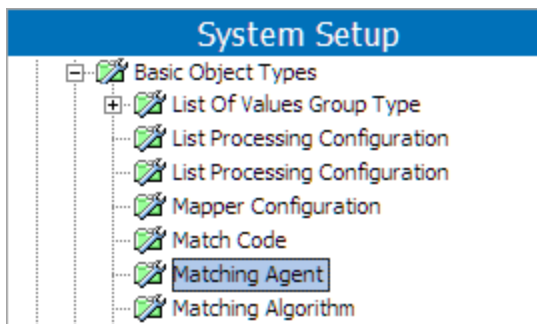


Configuring the Matching Agent Object Type

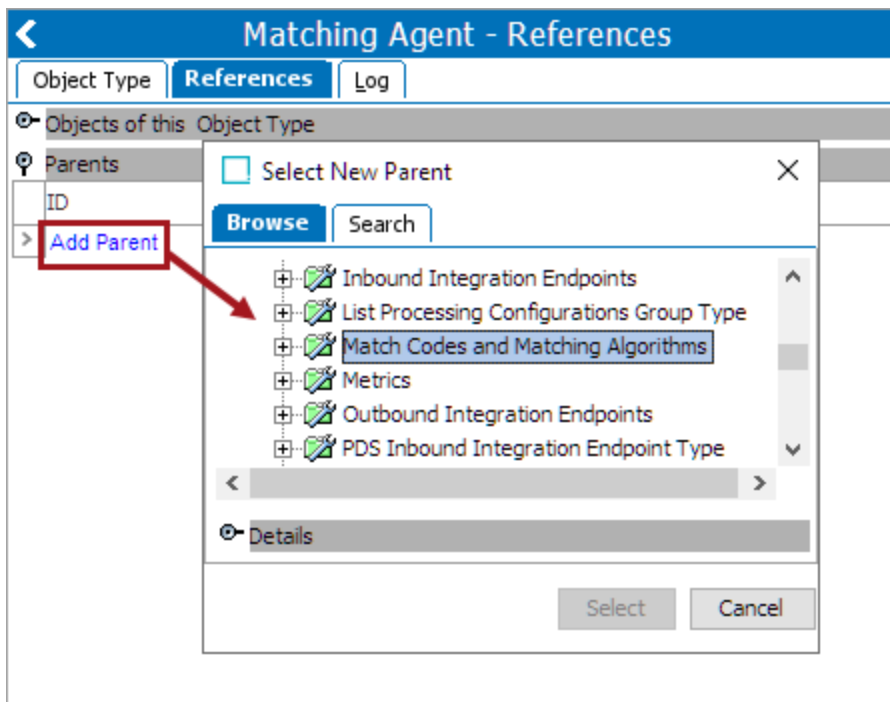
The matching agent stores the data steward's clerical review decisions. The machine learning matching agent (MLMA) then incorporates these decisions into its training and uses them to provide merge and reject recommendations for potential duplicate records.

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.



Configuring the Clerical Review Workflow for MLMA

The golden record clerical review workflow used by the machine learning matching agent (MLMA) determines the recommended action for the data steward based on a workflow variable.

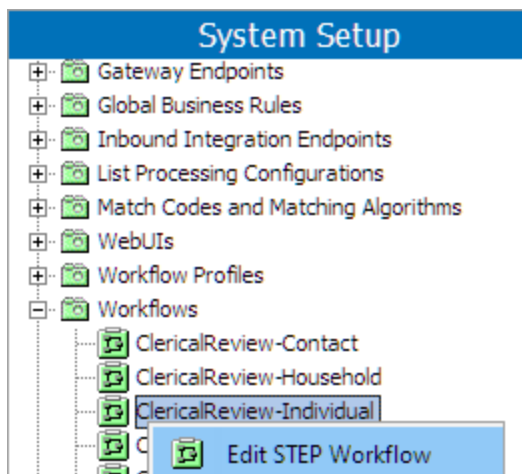
Prerequisites

Before the MLMA workflow variable can be configured, the golden record clerical review workflow must be configured or identified. For more information, see the **Creating a Merge Golden Record Clerical Review Workflow** topic.

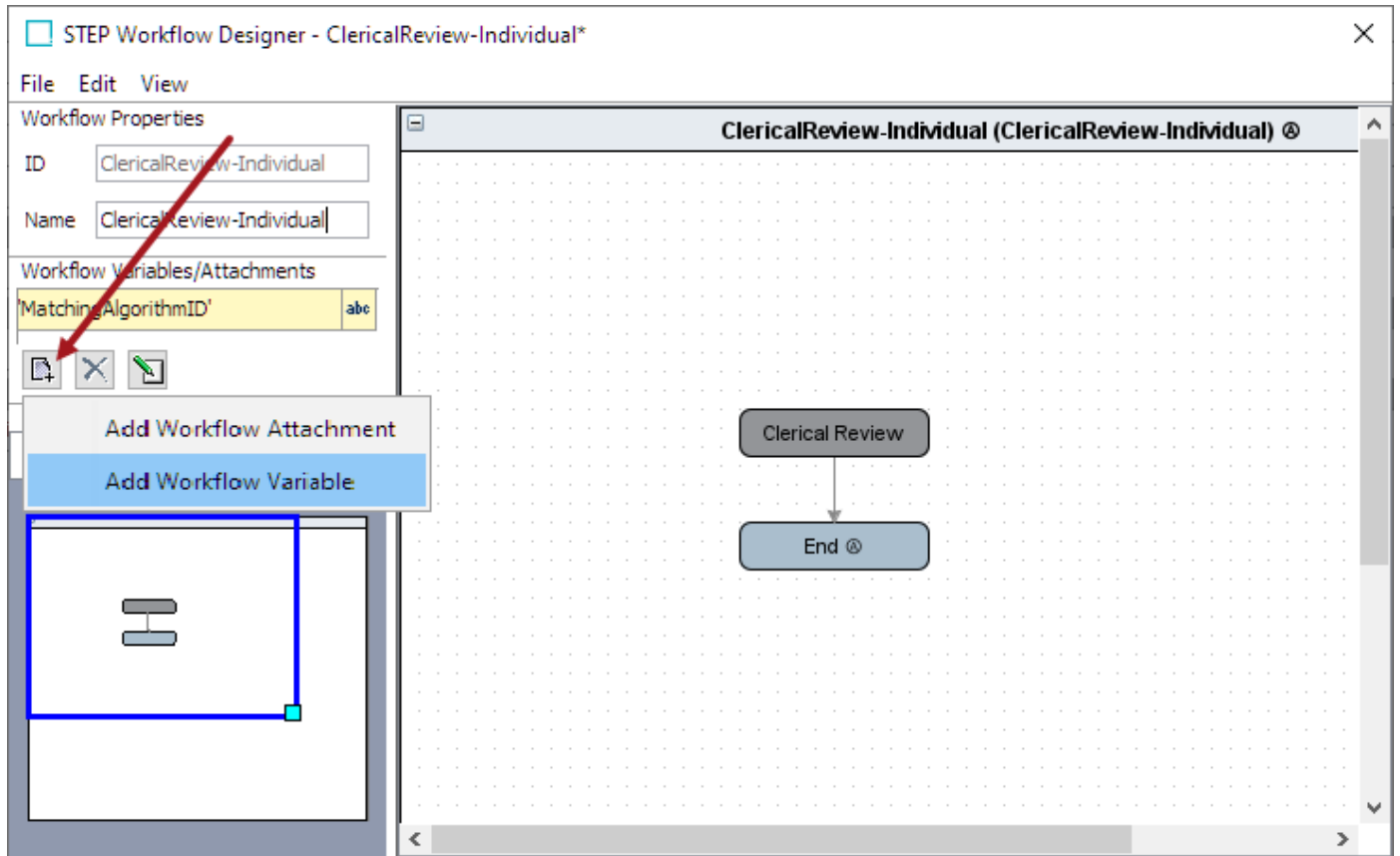
Configuring the Workflow

The following steps show how to modify the clerical review workflow for the MLMA:

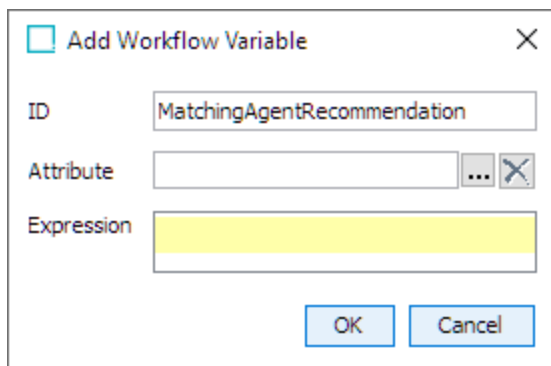
1. On the System Setup tab in workbench, navigate to Workflows, then right-click on the desired clerical review workflow and select 'Edit STEP Workflow' to display the STEP Workflow Designer.



2. Click the add icon and select 'Add Workflow Variable.'



3. In the Add Workflow Variable dialog, type in 'MatchingAgentRecommendation' for the ID. Do not add any attributes or expressions.



Click **OK**.

4. On the File menu, click **Save and exit** to close the STEP Workflow Designer dialog.

Adding MLMA Recommendations to a Clerical Review Task

The following steps show how to configure an existing clerical review task list for the machine learning matching agent. For more information on creating a clerical review task list for golden records, see the **Golden Record Clerical Review Task List** topic.

1. In design mode, navigate to the desired Golden Record Clerical Review Task List screen, and under 'Headers,' click **Add**.

Properties

Configuration Web UI style

Tasklist-Individual Save Close New... Delete Rename Save as...

Golden Record Clerical Review Task List Properties

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 (100 / ID)
- Golden Record Attribute Value Header (FirstName / First Name)
- Golden Record Attribute Value Header (LastName / Last Name)
- Golden Record Attribute Value Header (Gender)
- Golden Record Data Container Attribute Value Header (MainAddressDa

Add... Edit... Remove Up Down

Include Labels

▶ Group Options

- In the Add Component dialog, select the 'Golden Record Matching Agent Recommendation Header,' and click **Add**.

Add Component

Golden Record Attribute Value Group Header	Table header that shows the Matching Agent merge recommendation per task in the Clerical Review Task List
Golden Record Attribute Value Header	
Golden Record Data Container Attribute Value Header	
Golden Record ID Header	
Golden Record Matching Agent Recommendation Header	
Golden Record Name Header	
Golden Record Reference Type Header	
Golden Record Source Information Header	

Filter

Show deprecated components

Cancel
Add

- Use the **Up** and **Down** buttons to move the header to the desired location.

Properties

Configuration Web UI style

Tasklist-Individual Save Close New... Delete Rename Save as...

Golden Record Clerical Review Task List Properties

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 Matching Agent Recommendation Header
- Golden Record ID Header (100 / ID)
- Golden Record Attribute Value Header (FirstName / First Name)
- Golden Record Attribute Value Header (LastName / Last Name)
- Golden Record Attribute Value Header (Gender)

Add... Edit... Remove Up Down

Include Labels

▶ Group Options

4. Click **Save** and **Close**.

Maintaining the MLMA Data Model

In order to function, the machine learning matching agent (MLMA) must point to a matching algorithm. Within the matching algorithm, you must set up the data model, which the MLMA will use when making merge or reject recommendations.

Prerequisites

Configure a matching algorithm and the machine learning matching agent. For more information, see 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 Algorithm	Match Criteria	Match Code Values	Match Result	Agent Configuration	Score
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					

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 in order 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, see 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 must 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, see the **Matching Agents** topic.

Matching Agents

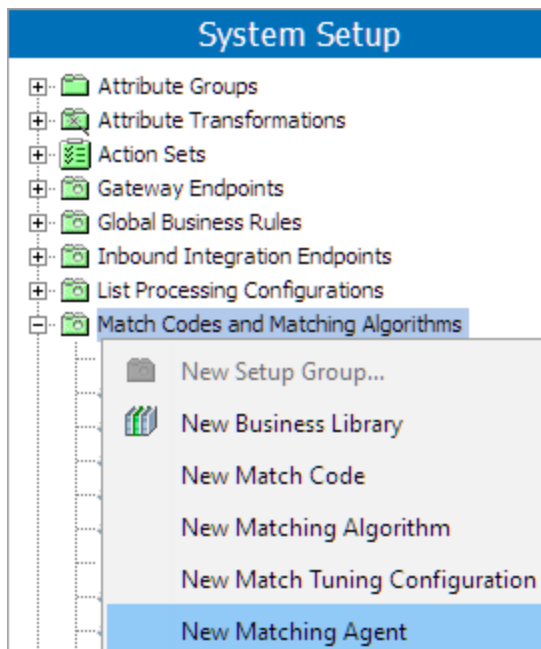
The matching agent stores the data steward's clerical review decisions. The machine learning matching agent (MLMA) then incorporates these decisions into its training and uses them to provide merge and reject recommendations for potential duplicate records.

Prerequisites

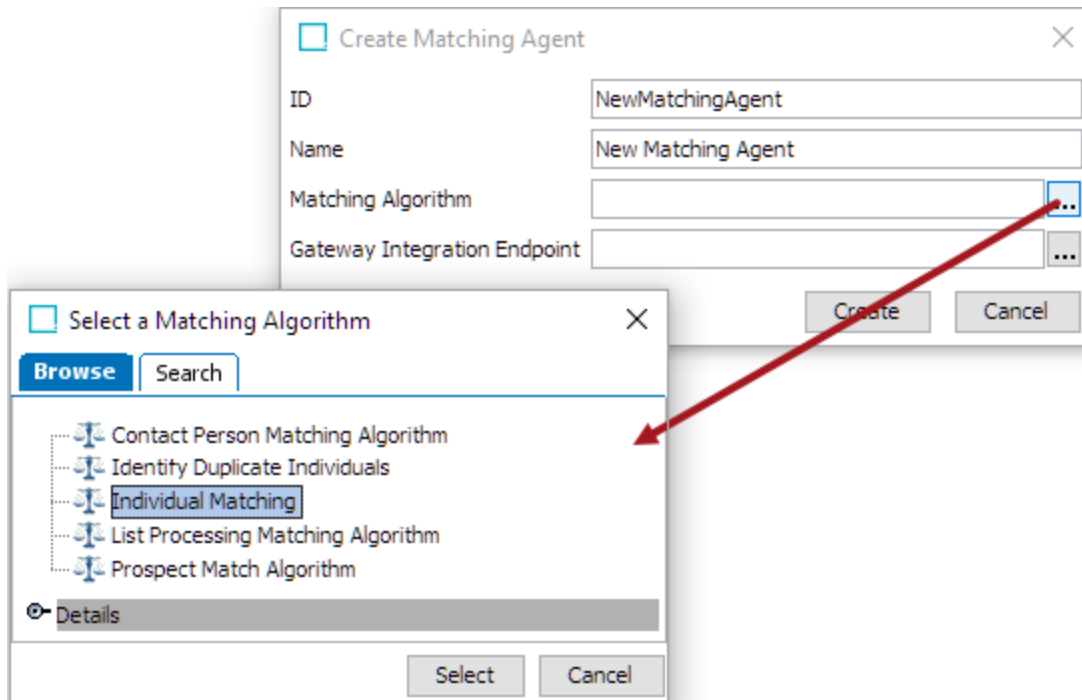
1. Identify or configure a matching algorithm on which the MLMA can function. For more information, see the **Configuring Matching Algorithms** topic.
2. Configure the REST gateways for the matching agent. For more information, see the **Configuring the MLMA REST Gateways** topic.
3. Configure the matching agent object type. For more information, see the **Configuring the Matching Agent Object Type** topic.

Create Matching Agents

1. Navigate to the setup group used for matching algorithms. Right-click the parent node and select 'New Matching Agent.'

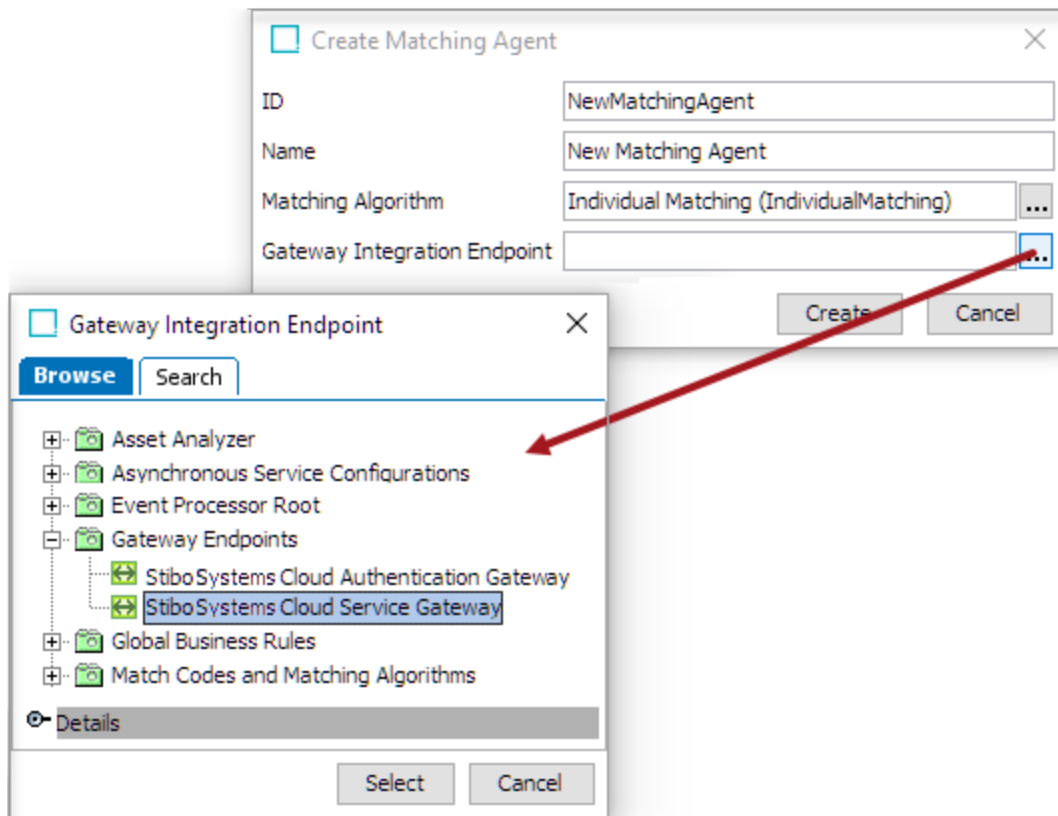


2. Set the name and ID, then click the ellipsis button (...) next to the **Matching Algorithm** parameter to select a matching algorithm.

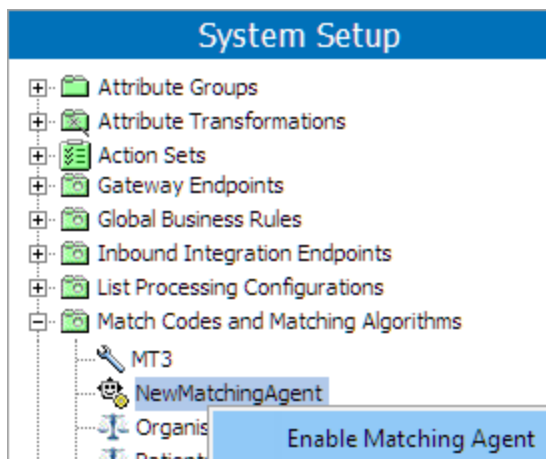


Note: Only matching algorithms using embedded match codes can be used.

3. For the **Gateway Integration Endpoint** parameter, click the ellipsis button (...) and select the service integration endpoint.



4. 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.

Do Training

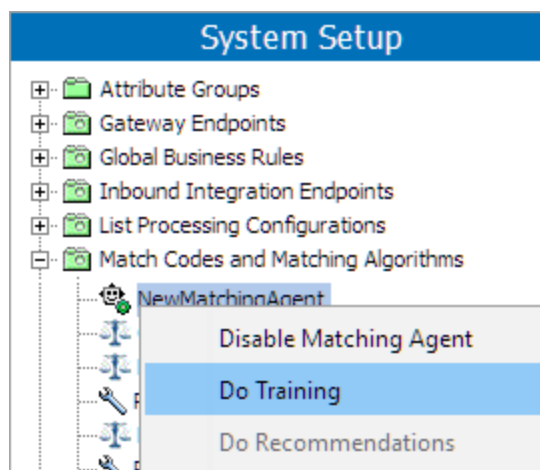
Once enabled, the machine learning matching agent trains automatically and continuously as the data steward makes decisions to merge and reject. Training works as a background process, gathering the data steward's decisions to base its recommendations on. After the data steward has met the threshold of required decisions, the matching agent will begin offering recommendations.

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 appear in the clerical review task list.

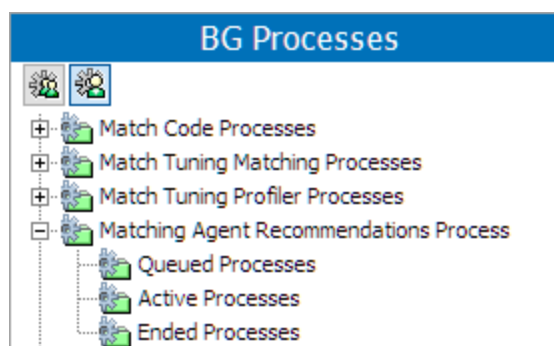
If you need to receive recommendations before the minimum threshold of decisions, you can manually perform the training at any time. However, Stibo Systems does not recommend manual training.

Important: Manual training before the minimum required decisions have been made could result in less accurate recommendations.

In order to manually perform the training and thereby 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 always automatically starts. Once this happens, 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, see the **Adding MLMA 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. This action can be used after importing new records to the system that have resulted in new tasks in Clerical Review.

Note: There is no need to manually run 'Do Recommendations' after the 'Do Training' action since it happens automatically.

Clerical Review Task List with Matching Agent Recommendations

The following assumes you have configured the machine learning matching agent (MLMA) for clerical review. For more information, see the **Configuring MLMA** topic.

Clerical Review Decisions and Match 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 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 algorithm performing the machine learning training excludes merge decisions for tasks with three or more records. This is because of the possible data differences in all individual pair combinations that could cause false positives in the machine learning engine. For this reason, the first recommendations may take some time before appearing if the task list contains a minor number of tasks with only two golden records.

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. In the early adopter program, these thresholds can be manually adjusted. Contact your Stibo Systems account manager or partner manager to initiate this activity.

Below is an example of how a golden record clerical review task list might appear with recommendations:

Golden Record Clerical Review Task List

Advanced Merge
 Merge
 Reassign
 Reject

⚠ Not all potential duplicates are shown for all tasks.

Task	Golden Record	Main Address	Source Informati...	First Name	Last Name	Email	Phone
<input type="checkbox"/> Assignee: Stibo Users Created: 3/1/22 Merge	ID: 822363 • Match Score: -- Created: 2/11/22 • Updated: 2/11/22	1932 Iantana dr...	CRM Global 102...	Michael	Pierce	36mjp64@yaho...	3163975260; 5...
	ID: 824421 • Match Score: 67.5 Created: 2/11/22 • Updated: 2/11/22	1932 Ibatana dt...	Dynamics Europ...	Mixshel	Pirece	37mjp64@yaho...	3751059728; 3...
<input type="checkbox"/> Assignee: Stibo Users Created: 3/1/22 Reject	ID: 819936 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	19 Overlook Rid...	SAP London 16...	Cathy	Miller	GustavoBarrosC...	(593) 145-3181;...
	ID: 820273 • Match Score: 83.84 Created: 2/2/22 • Updated: 2/9/22	19 Overlook Rd...	Dynamics Europ...	Kathy	Miller	eros.nec@Morbi...	(604) 658-0190;...
<input type="checkbox"/> Assignee: Stibo Users Created: 3/1/22 Merge	ID: 820088 • Match Score: -- Created: 2/2/22 • Updated: 2/9/22	36 Garden St Sh...	SAP US 161705...	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...	SAP US 165901...	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...	Dynamics Europ...	Sheley	Fullgum	non.ante@seddi...	(216) 435-0544;...
<input type="checkbox"/> Assignee: Stibo Users Created: 3/1/22 Reject	ID: 823371 • Match Score: -- Created: 2/11/22 • Updated: 2/11/22	4916elizabethdr...	Dynamics Europ...	Timothy	Price	TimothyPrice91...	9198780765; 8...
	ID: 825328 • Match Score: 50 Created: 2/11/22 • Updated: 2/11/22	4916elizaberhdi...	SAP London 10...	Tjmtohy	Priec	4chldrmorperti...	3011505334; 2...

Total Number of Tasks: 63; Selected items: 0

Find Similar

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 by clicking 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, click the 'Soap API documentation' link, and click the 'Core WebServices available in the STEP system' link to find details on: <ul style="list-style-type: none"> • getSimilarObjectsRequestType • getSimilarObjectsResponseType 	REST On STEP API Documentation, click 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: <ul style="list-style-type: none"> • /entities/find-similar
Interfaces with	Web UI via: <ul style="list-style-type: none"> • 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 (see the Object Super Types topic in the Getting Started / User	Entity

	getSimilarObjects	entities/find-similar
allowed	Guide documentation)	
Matching Algorithm	<p>Must include Node Binds for the input values. The same matching algorithm can be used for:</p> <ul style="list-style-type: none"> • '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	See the Find Similar - getSimilarObjects topic	See the Find Similar - entities/find-similar topic

Find Similar - getSimilarObjects

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, see the **Initiate Item Screen** topic in **Web User Interfaces / Web UI Setup and User Guide** documentation.
- The 'Find Similar' search on an 'Add Reference' action in Web UI. For more information, see the **Add Reference Action** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.
- 'getSimilarObjects' SOAP web service request. For more information, see the **Find Similar (getSimilarObjects)** topic in the **Customer MDM Solution Enablement** documentation.

For use case examples, see 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, see 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. See 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, see the **Find Similar Node Binds** topic.

Find Similar 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:

```

1         if ( node == null ) {
2             //generate matchcodes for getSimilarObject
3         }
4         else {
5             //generate matchcodes for existing objects
6         }

```

Note: When used for getSimilarObjects, it is safe to establish large match code groups without impacting performance.

For more information, see 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, see the **Configuring Matching Algorithms** topic of the documentation.

Find Similar getSimilarObjects Node Binds

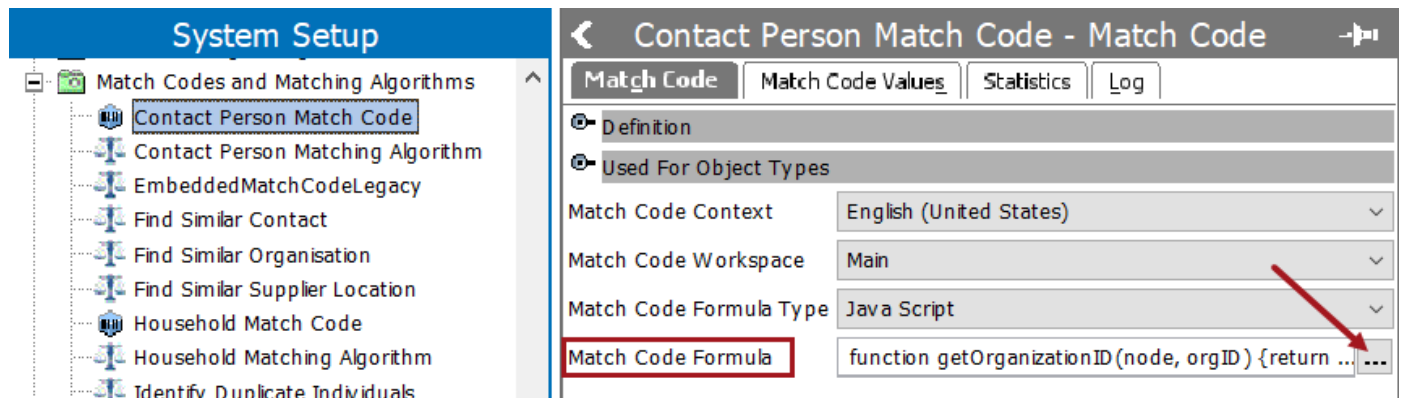
A node is a permanent STEP object. When a Find Similar getSimilarObjects call is made however, a node is not created in STEP. This non-permanent state means that the match code cannot obtain the 'Current Object,' and the matching algorithm cannot obtain the `first()` node via the 'Match Expression Context.' 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 (...):



- Open the Binds flipper and define all required binds. For information on JavaScript binds, see the **JavaScript Binds** topic in the online help **Resource Materials** documentation.

JavaScript		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.

```

1 function getOrganizationID(node, orgID) {
2     if(node) {
3         var iter = node.getReferences(node.getManager().getR
4             if(iter.hasNext()) {
5                 return iter.next().getTarget().getID();
6             } else {
7                 return null;
8             }
9     } else {
10        return orgID;
11    }
12 }
13
14 var matchCodeArr = new Array();
15
16 var input = {
17     "node" : node,
18     "FirstName" : FirstName,
19     "LastName" : LastName,
20     "Zip" : Zip,
21     "EmailAddress" : EmailAddress,
22     "PhoneNumber" : PhoneNumber
23 };
24 var organizationID = getOrganizationID(node, OrganizationID);
25 if(organizationID) {
26     matchCodeLib.appendEmailMatchCode(input, matchCodeArr, or
27     matchCodeLib.appendPhoneMatchCode(input, matchCodeArr, or
28     matchCodeLib.appendIndividualNameAndAddressMatchCode(inpu
29 }
30 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 getSimilarObjects on Initiate Item Screen

On Initiate Item screens in Web UI, Find Similar getSimilarObjects helps users identify similar objects prior to initiating new ones, limiting the number of duplicate objects in Web UI.

This topic include the following sections:

- Using Find Similar on Initiate Item Screen
- Configuring Find Similar on Initiate Item Screen


For other configuration and uses, see the **Find Similar** topic.

Using Find Similar on Initiate Item Screen

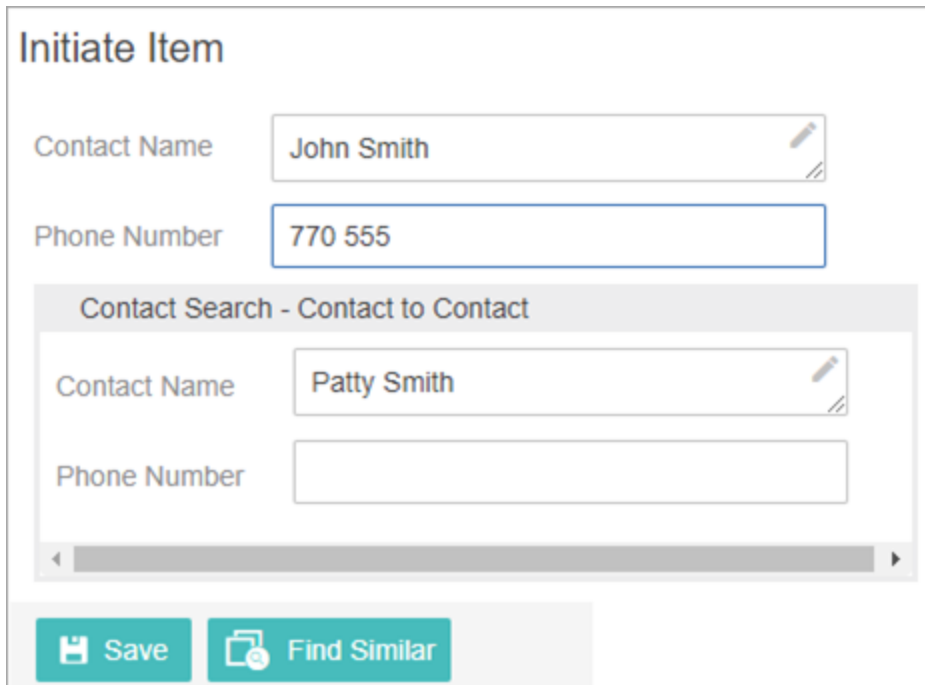
This example shows how Find Similar is used on an 'Initiate Item' screen, including a 'Store Single Referenced Target' component.

The 'Store Single Referenced Target' component is intended to find a similar record on 'Initiate Item' workflow state. The component can be used to create a single reference target with attributes and create a reference to it from the matched record.

For example, a user creates a Contact object by clicking 'Initiate Contact' on the 'Status Selector Homepage Widget.'

CONTACT WORKFLOW	
	
Initiate Contact	
State-1	1
State-2	0
State-3	0
End State	0

On the Initiate Screen, the user enters data into the direct object search fields and into the referenced object search fields and then clicks the 'Find Similar' button.



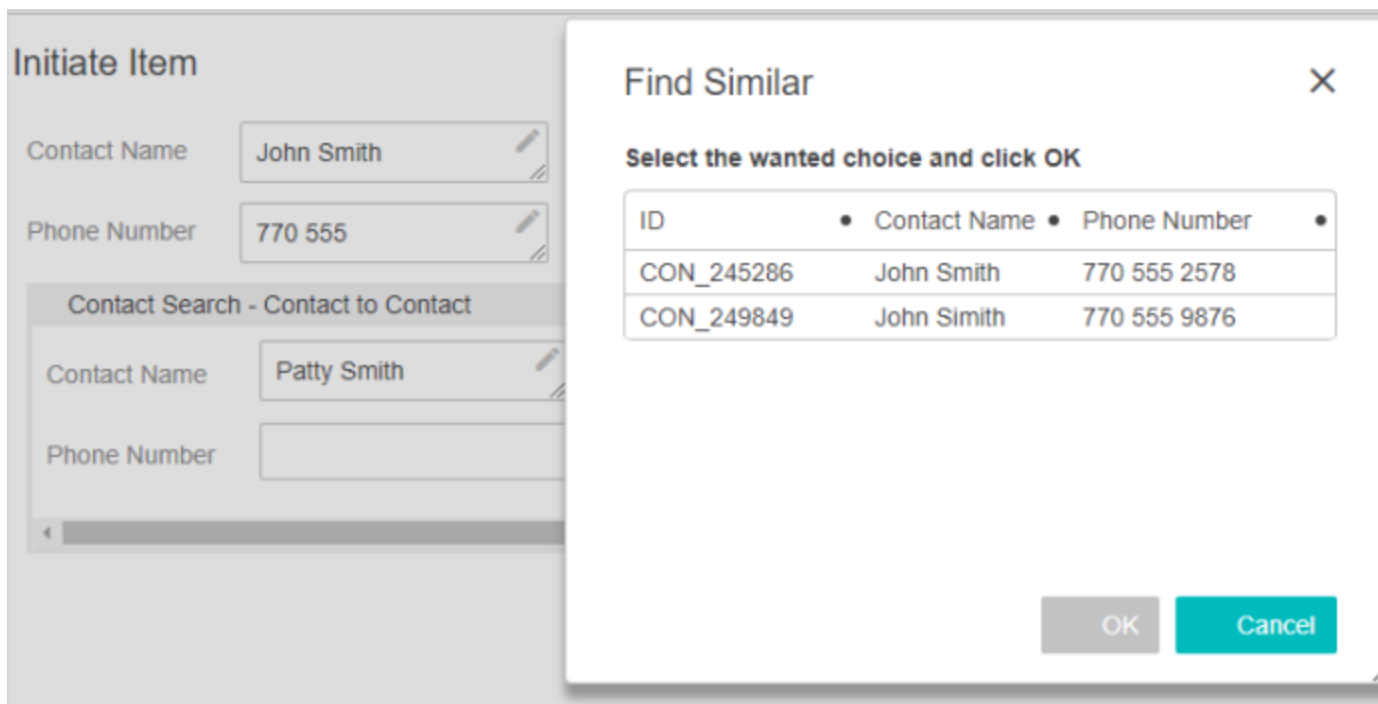
The screenshot shows a web form titled "Initiate Item". It contains two main sections for data entry:

- Direct Object Search:**
 - Contact Name: John Smith
 - Phone Number: 770 555
- Referenced Object Search (Contact Search - Contact to Contact):**
 - Contact Name: Patty Smith
 - Phone Number: (empty)

At the bottom of the form, there are two buttons: "Save" and "Find Similar".

The algorithm runs in the background, and the configured Dialog List Screen displays. Be aware that this is not a standard search, and the results are based on the matching algorithm running in the background.

Important: As stated at the beginning of this topic, the key to Find Similar functionality is the matching setup that the customer creates and uses for duplicate handling. This is important because the attributes being searched must be part of the list of attributes that match codes are generated for; if not, the search will not work as expected.



A maximum of fifty (50) objects are shown at one time on the results list. The user can select one of the objects shown in the results list. After choosing an item, select the OK button. Clicking OK takes the user directly to that object via the appropriate Web UI screens.

If the user wants to create a new object with the search data entered, the user clicks Cancel to exit the results list. On the Initiate Screen, the user will click Save and be taken directly to the new object. The data entered into the attribute value fields, which are configured in the Store Single ReferencedTarget Properties, is also used to generate the referenced object and the reference between these objects is also created.

Initiate Item

Contact Name

Phone Number

Contact Search - Contact to Contact

Contact Name

Phone Number

Save **Find Similar**

Tree

- Products
- Discontinued Products
- Packaging
- Assets
- Suppliers
- Merchandising Hierarchy
- Collections
- Entity Root
 - Address Root
 - Customer Hierarchy
 - Customer A
 - Customer B
 - Axel Kiers Vej 11
 - John Smith**

Node Details

Contact Information Potential Duplicates Confirmed Non Matches Revisions Data Visualization

Contact Name

Phone

Select all
 Clear filter
 Apply view
 Clear view
 Add Reference
 Multi edit view

	ID	Contact Name	Object Type	Reference type
<input type="checkbox"/>	Stibo Canada	ADD_184139	Address	Address
<input type="checkbox"/>	Patty Smith	CON_109923	Contact	Contact to Contact
<input type="checkbox"/>	Jimmy Smith	CON_149944	Contact	Contact to Contact
<input type="checkbox"/>	Contact US	CON_149968	Contact	Contact to Contact

If applicable, the new object will automatically be initiated into a workflow or workflows based on existing workflow rules. For more information about workflows and auto-initiation, see the **Auto-Initiation of Tasks in Workflows** topic in the **Workflows** documentation. It is advisable to also make use of standard STEP tools like business rules and workflows to manage the referenced objects that are created in this manner.

Configuring Find Similar on Initiate Item Screen

Prerequisites

All the steps provided in this topic assume the Web UI designer is in design mode and on the applicable Properties screen prior to starting the configuration process. It is also assumed that all users have the proper privilege to work with these features. For more information about privileges and user setup, see the **Users and Groups** topic and **Adding User Privileges for a Group** topic of the **System Setup / Super User Guide** documentation.

Properties (edited)

Configuration Web UI style

initiate Save Close New... Delete Rename Save as...

Initiate Item Properties

Duplicate Handler Duplicate Handler Edit...

Object Type ID* CD_Contact ...

Root step://entity?id=ContactsRoo ... Clear

Root Product ID

▼ Screen configurations

Forwarding Screen CD-SimpleMatching Add

▶ Business Conditions

▶ Styling

Child Components

Main Columns Control go to component

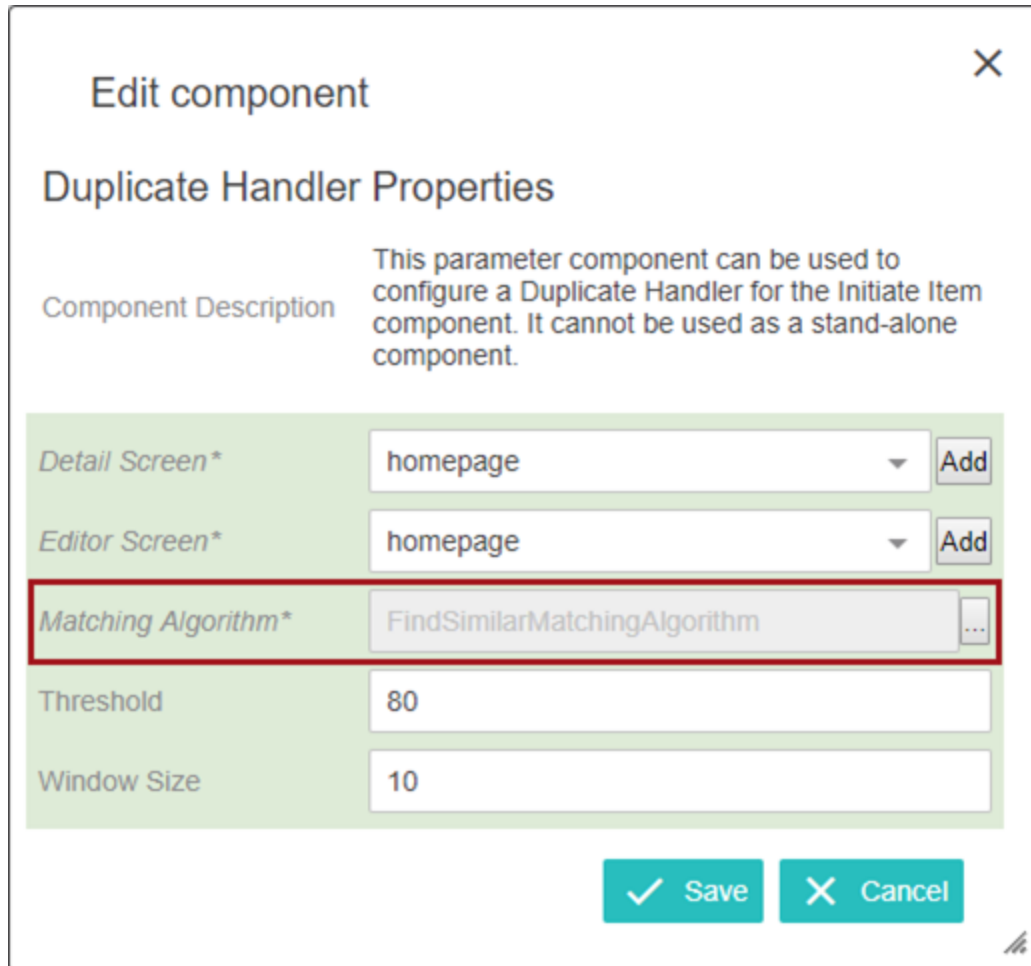
Buttons Buttons go to component

Breadcrumb <Select a child com go to component

1. Create a new 'Initiate Item' screen or use an existing one.
2. In 'Initiate Item Properties,' select 'Duplicate Handler' in the dropdown for the 'Duplicate Handler' parameter.

The Duplicate Handler Properties screen will display. The only required setting is the matching algorithm parameter.

3. Click the ellipsis button (...) to the right of the value field and select a matching algorithm. Click Save.



Edit component ✕

Duplicate Handler Properties

Component Description This parameter component can be used to configure a Duplicate Handler for the Initiate Item component. It cannot be used as a stand-alone component.

<i>Detail Screen*</i>	homepage	▼	Add
<i>Editor Screen*</i>	homepage	▼	Add
<i>Matching Algorithm*</i>	FindSimilarMatchingAlgorithm		...
Threshold	80		
Window Size	10		

✓ Save
✕ Cancel

4. On the Initiate Item properties dialog, for Object Type ID, if needed, click the ellipsis button (...) to the right of the Object Type ID parameter to select the object type of the objects that will be initiated into the system.
5. Click the ellipsis button (...) to select a Root directory. New items will be created below the designated root folder. A default value will be populated.
6. Select a 'Forwarding Screen' option. This screen is what is shown once an object is created.
7. For the 'Main' child components option, choose 'Columns Control' from the dropdown. Click on 'go to component' to configure this option. Select 'Node Editor' in the Columns Control Properties.
8. Double click the Node Editor component to go to the Node Editor Properties dialog. On this dialog, the search attributes need to be configured within the Child Components > Rows value field using the add and remove buttons. These search attributes are for searching direct objects.

Properties (edited)

Configuration Web UI style

Initiate Save Close New... Delete Rename Save as...

Node Editor Properties [go to parent](#)

Component Description The Node Editor can be configured with a range of components for displaying different information and details about a selected object.

Child Components

Rows	Name
	Name Value (Name)
	Attribute Value Component (Description)
	Attribute Value Component (Manufacturer Part N
	Attribute Value Component (Brand Name)
	Attribute Value Component (Country of Origin)
	Store Single Referenced Target

Add.. Remove Up Down

- If the use case requires searching on referenced objects in addition to direct objects, users must add the 'Store Single Referenced Target' component. Double click on 'Store Single Referenced Target' to configure the component. The 'Object Type,' 'Parent,' 'Reference Type,' and 'Child Component > Rows' are required parameters / components. Go through each parameter one-by-one to complete the configuration.

✕

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.
 Note: This component has required child components that can not be configured in this dialog

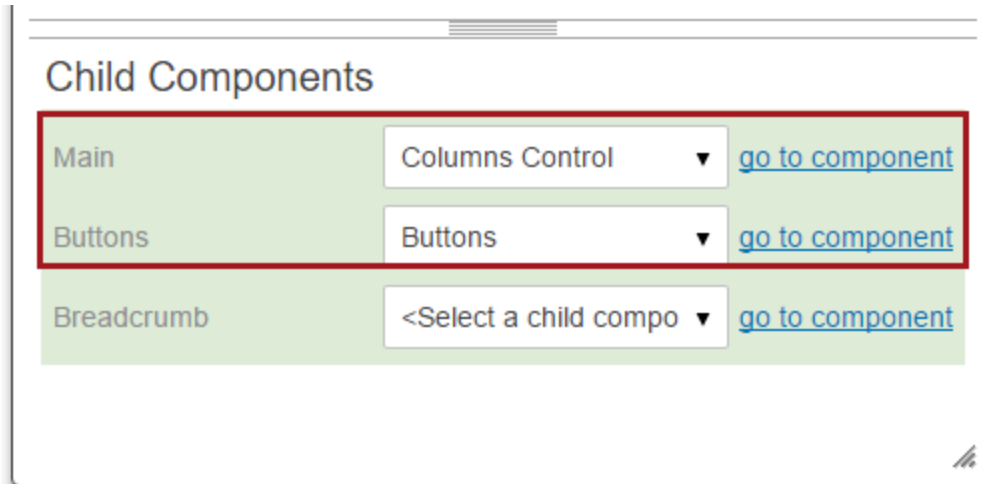
Store Single Referenced Target Properties

Component Description This component is intended used for find similar search on Initiate Item. The component can be used to create a single reference target with attributes and create a reference to it from selection

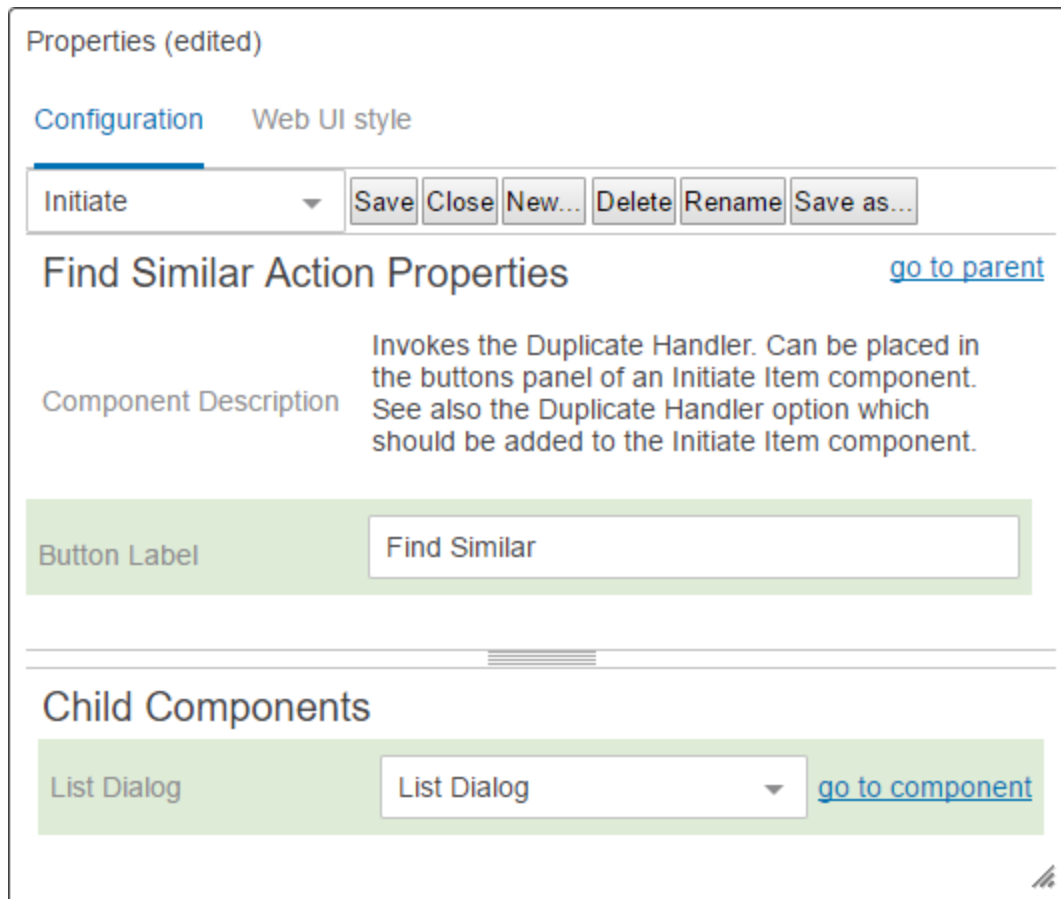
<i>Object Type*</i>	<input type="text" value="CD_Contact"/> ...
<i>Parent*</i>	<input type="text" value="step://entity?id=ContactsRoot"/> ...
<i>Reference Type*</i>	<input type="text" value="ContactToContact"/> ...
Duplicate Handler	<input type="text" value="<Select an option>"/> <input type="button" value="Edit..."/>
Hide Section If Empty	<input type="checkbox"/>
Section Default Open	<input checked="" type="checkbox"/>
Section Title	<input type="text" value="Contact Search"/>

⌵

10. On the parent screen's 'Initiate Item Properties' screen, go to 'Child Component > Buttons' and click on 'go to component.' In the 'Actions' value field, select 'Add' and select 'Find Similar Action.'

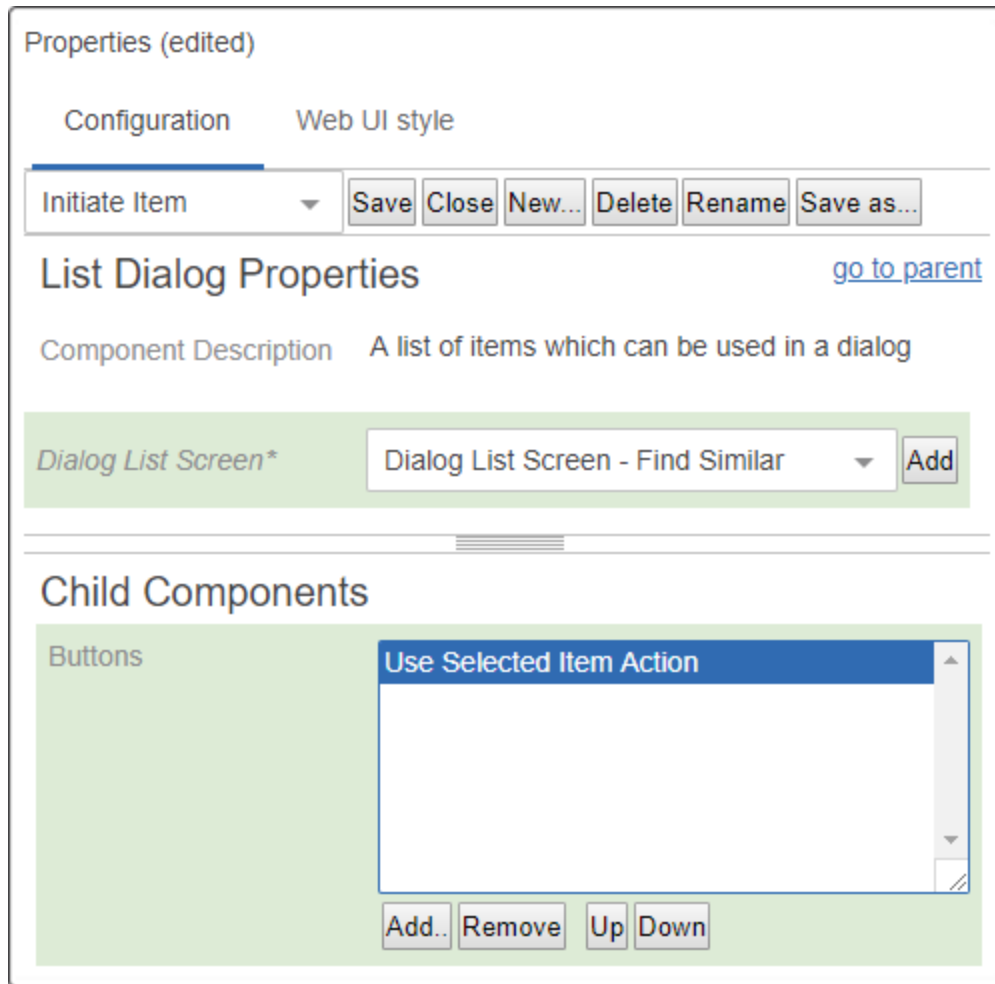


- Double click on the 'Find Similar Action' title and open the 'Find Similar Action Properties' dialog.



- Edit the Button Label, if desired. Under Child Components > List Dialog, select 'List Dialog' from the dropdown menu.

13. On 'List Dialog' Properties, select a Dialog List screen. If a Dialog List screen does not exist, follow the next two steps to create one.
14. Click Add to the right of the Dialog List screen value field.
15. Create a new screen by selecting Dialog List screen from the list of available screen types / components. Enter a screen ID, and then, click Add. The screen ID will automatically populate in the Dialog List Screen value field.



16. On the List Dialog Properties screen, in the Child Components > Buttons field, click Add and select Use Selected Item Action. Click Add, and then click Save in the designer window.
17. Before exiting design mode, select the Dialog List Screen ID of the screen you just created from the dropdown menu. In the example above, the screen ID is 'Dialog List Screen - Find Similar.'
18. Configure the Child Components > Headers section of the Dialog List Screen Properties. Click Add under the value field to set up the attributes that will display in the Find Similar Search tab for the results list. The Help Text is editable and will display at the top of the results list.

Dialog List Screen Properties

Component Description Screen with a list which can be used in a popup dialog

Help Text	Select the wanted choice and click OK
-----------	---------------------------------------

Child Components

Headers	<ul style="list-style-type: none">Id List HeaderTitle List HeaderAttribute List Header (Contact Name)Attribute List Header (Phone Number) <p>Add.. Remove Up Down</p>
---------	--

19. Click Save and Close design mode to return to normal Web UI mode.

Find Similar getSimilarObjects on Add Reference Action

The getSimilarObjects Find Similar functionality can be used to search for and identify similar objects prior to adding references and creating target objects within a Multi-Reference Editor.

This topic include the following sections:

- Using Find Similar on an Add Reference Action
- Configuring Find Similar on Add Reference Action

For other configuration and uses, see the **Find Similar** topic.

Using Find Similar on Add Reference Action

The following section details an example of a configured 'Find Similar Search' tab.

While using a 'Multi-Reference Editor' component, click 'Add Reference,' and then, click the node picker icon on the 'Add reference' dialog that appeared. The Find Similar Search tab is displayed in the 'Select Node(s)' dialog. The end user enters data into the configured attribute fields and clicks OK. The algorithm runs in the background and the configured Dialog List Screen displays. Remember that this is not a standard search and results are based on the matching algorithm running in the background.

Select all Clear filter Apply view Clear view 1 Add Reference

Add Reference

Reference Type: Contact to Contact

Reference Target: Select options... 2

OK Cancel

Select Node(s)

Browse Search 3 Find Similar Search

- Company Hierarchy Data Root
- Customer Root
- Entity Root
- GDSN
- GDSN Receiver
- Merge_Golden_Root
- Promotions

OK Cancel

×
Select Node(s)

Browse
Search
Find Similar Search

Enter values to assist in finding similar

Contact Name 1

Phone Number

Reference Contact Name

×
Find Similar 3

Select the wanted choice and click OK

ID	Contact Name	Phone Number
CON_245286	John Smith	770 555 2578
CON_249849	John Simith	770 555 9876
CON_253640	John Smith	770 555

4

2

Important: The key to Find Similar functionality is the matching setup that the customer creates and uses for duplicate handling. This is important because the attributes being searched must be part of the list of attributes that match codes are generated for. If not, the search will not work as expected.

A maximum of fifty (50) objects are shown on the results list.

If the user clicks the row of a reference on the results list the OK button is enabled. Clicking 'OK' creates the reference. If the user does not find a result to use, click cancel to create a new reference using the Create or Create from Template functions, defined in **Add Reference Action** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.

Add Reference ✕

Reference Type ▼
Contact to Contact

Reference Target ✕
Patty Smith (CON_109923)

✓ OK
✕ Cancel

Success
✕

1 reference was created!

Confirmed Non Matches
Revisions
Data Visualization

☑ Select all
🗑 Clear filter
👁 Apply view
🗄 Clear view
➕ Add Reference
☰ Mult

		ID	Contact Name	Object Type	Reference type
<input type="checkbox"/>	Peachtree Circle	ADD_187585		Address	Address
<input type="checkbox"/>	John Smith	CON_107842	John Smith	Contact	Contact to Contact
<input type="checkbox"/>	Patty Smith	CON_109923	Patty Smith	Contact	Contact to Contact
<input type="checkbox"/>	Patty Smith	CON_149940	Patty Smith	Contact	Contact to Contact
<input type="checkbox"/>	Jimmy Smith	CON_170627	Jimmy Smith	Contact	Contact to Contact

Note: Find Similar functionality is also available when working with company data hierarchies via the Hierarchy Representations component (found in Main (Screen) Properties). For more information on the setup, which is similar to Add Reference, see the **Optional Configurations for the Hierarchy Representation Component** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.

Configuring Find Similar on Add Reference Action

This setup information is also available in the **Add Reference Action** topic of the **Web User Interfaces / Using a Web UI** documentation.

Properties

Configuration Web UI style

productdetails Save Close New... Delete Rename Save as...

Duplicate Handler Duplicate Handler Edit...

Find Similar Search Fields

Add... Edit... Remove Up Down

Show Find Similar Tab

Temp Object Type For Find Similar ... Clear

Temp Parent For Find Similar ... Clear

Child Components

Create From Template List List Dialog [go to component](#)

Find Similar List Dialog List Dialog [go to component](#)

1. Edit the properties for an existing Add Reference Action configured for a Multi-Reference Editor, or add an Add Reference Action and complete the configuration.
2. In the Add Reference Action Properties, select Duplicate Handler in the dropdown for the Duplicate Handler parameter.
3. The Duplicate Handler Properties screen will display. The only required setting is the Matching Algorithm parameter. Click the ellipsis button (...) to the right of the value field and select a matching algorithm. Click Save.

Edit component
✕

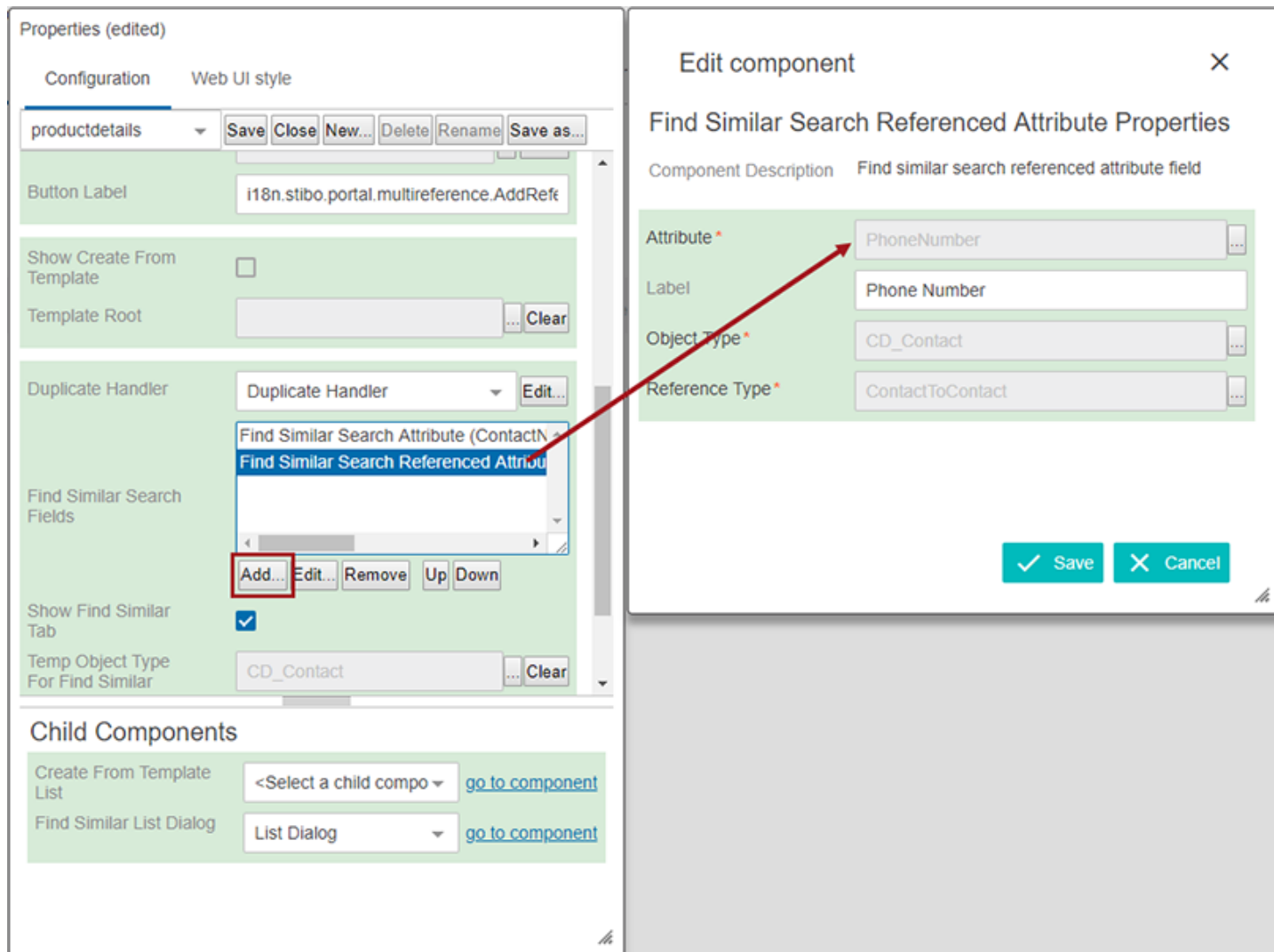
Duplicate Handler Properties

Component Description: This parameter component can be used to configure a Duplicate Handler for the Initiate Item component. It cannot be used as a stand-alone component.

Detail Screen *	<input type="text" value="homepage"/> ▼ Add
Editor Screen *	<input type="text" value="homepage"/> ▼ Add
Matching Algorithm *	<input type="text" value="FindSimilarMatchingAlgorithm"/> ⋮
Threshold	<input type="text" value="80"/>
Window Size	<input type="text" value="10"/>

✓ Save
✕ Cancel

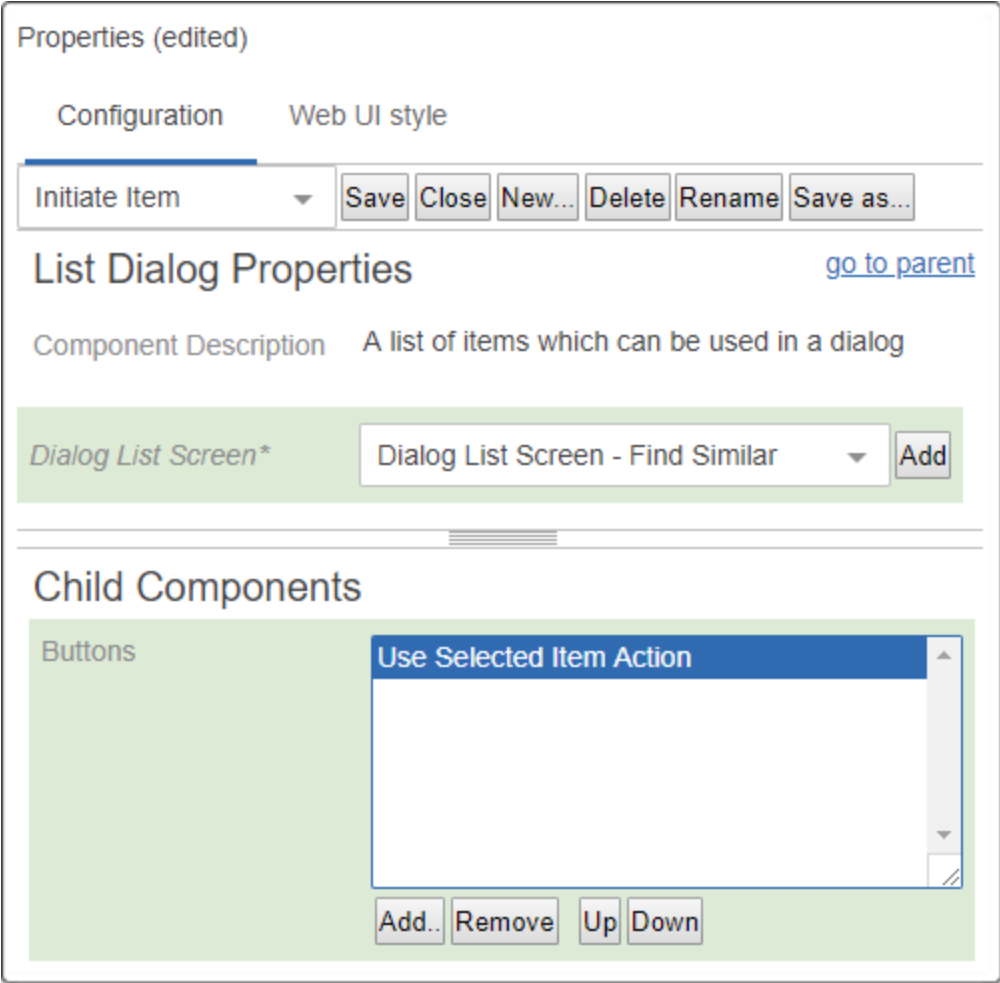
4. On the 'Add Reference Action Properties' again, click Add under the Find Similar Search Fields value box. Add, remove, and re-order Attribute and Referenced Attributes as desired. Remember that these search fields will need to be incorporated into the matching algorithm configured in the Duplicate Handler parameter.
5. It is important to fill in the Label fields during the Find Similar Search Fields configuration. These labels will appear on the Find Similar Search tab. If adding a Find Similar Search Referenced Attribute, fill in the Attribute, Label, Object Type, and Reference Type values. Save all changes.



6. Enable the Show Find Similar Tab setting back on the Add Reference Action Properties.
7. Click the ellipsis button (...) to make a selection for the Temp Object Type For Find Similar parameter and the Temp Parent For Find Similar parameter. (When the matching algorithm is run, it creates temporary objects based on the input in the search fields. These objects are then used in the algorithm to compare and find similar objects. The temporary objects need a parent and object type to be created. After the user finishes the operation, the temporary objects are deleted by the system. These parameters have to be configured for the functionality to work. The object types of both the direct objects and the referenced objects need to be made valid under the location used for the Temp Parent For Find Similar parameter.)

Important: The node selected for Temp Parent For Find Similar in the configuration, must also be included in the category specified in the match code definition if a category is specified. If this is not done, the match codes will not generate properly, and the match results will be incorrect.

8. Under Child Components > Find Similar List Dialog, select List Dialog from the dropdown menu.
9. On List Dialog Properties, click Add to the right of the Dialog List Screen value field.
10. Create a new screen by selecting Dialog List Screen from the list of available screen types / components. Enter an easily identifiable Screen ID, click Add. The screen ID will automatically populate in the Dialog List Screen value field.
11. If a Dialog List Screen already exists, skip the previous two steps, and on the List Dialog Properties, select the Dialog List Screen using the dropdown.
12. On the List Dialog Properties screen, in the Child Components > Buttons field, click Add and select Use Selected Item Action. Click Add, and then click Save in the designer window.



13. Before exiting design mode, select the Dialog List Screen ID of the screen you just created from the dropdown menu. In the example above, the screen ID is Dialog List Screen - Find Similar.
14. Configure the Child Components > Headers section of the Dialog List Screen Properties. Click Add under the value field to set up the attributes that will display in the Find Similar Search tab for the results list. The Help Text is editable and will display at the top of the results list.
15. Click Save and Close design mode to return to Web UI mode.

Find Similar - entities/find-similar

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 clicking 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, see the **Find Similar Web Service** topic in **Customer MDM Solution Enablement** documentation.