

Data Merger Guide

JACAMAR®

Version 5



Document Reference No.: Ka_000168
Version 1.110
Issue Date: 2019-01-10



Information in this document is subject to change without notice and does not represent a commitment on the part of Katla GmbH.

Katla's software and documentation have been tested and reviewed. Nevertheless, Katla makes no warranty or representation, either express or implied, with respect to the software and documentation included with Katla product. In no event will Katla be liable for direct, indirect, special, incidental or consequential damages resulting from any defect in the software or documentation included with this product. In particular, Katla shall have no liability for any programs or data used with this product, including the cost of recovering programs or data.

Katla JACAMAR® is a registered trademark of Katla GmbH Magdeburg, Germany.

Microsoft, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7 are either registered trademarks or trademarks of Microsoft Corporation.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

All trademarks and copyrights mentioned in this documentation are the property of their respective holders.

JACAMAR® – Data Merger Guide

<http://www.jacamar.biz>

JACAMAR® Copyright © 2009-2019 Katla GmbH. All rights reserved.

Katla GmbH

Immermannstr. 28

D-39108 Magdeburg

Deutschland



CONTENTS

1 Introduction.....	3
1.1 Structure of the guide.....	3
1.2 The function of Data Merger.....	4
1.3 Source data.....	4
2 Using Data Merger.....	5
2.1 Starting Data Merger.....	5
2.2 The Data Merger dialog.....	5
2.3 The Data Merger scheme.....	7
2.3.1 Source data types and conditions.....	7
2.3.2 Cascade information about types.....	7
2.3.3 Mapping column information.....	8
2.3.4 Adding and Deleting rules.....	10
2.3.5 Functions in Property rules.....	10
2.3.6 Regular expressions in Column rules (Advanced).....	11
2.3.7 Setting merge criteria and unique assignment.....	12
2.3.8 Save the scheme for re-use.....	13
2.4 Performing a data merge.....	14
<i>Highlight the area in the source spreadsheet and copy to the clipboard.....</i>	14
<i>Paste the clipboard in Data Merger.....</i>	14
<i>Control of merged elements in the preview and a „corresponding view‘.....</i>	15
2.5 The Merge preview.....	15
2.5.1 Layout.....	15
2.5.2 Merge preview functionality.....	16
<i>RMB context menus contain:.....</i>	16
<i>Colour Coding and actions explained:.....</i>	16
<i>Additionally with open views (corresponding views).....</i>	19
<i>Counter-action.....</i>	19
<i>Drag and drop.....</i>	20
2.6 General Merging Features.....	21
APPENDIX.....	22
<i>Summary of Type rule condtion expressions.....</i>	22
<i>Summary of regular-expression constructs.....</i>	22
<i>Groups and capturing.....</i>	24

1 Introduction

1.1 Structure of the guide

This guide describes the operation and application of JACAMAR® Data Merger. In the introductory chapter the function of Data Merger is briefly explained and assistance in the preparation of the source data that is to be merged.

The second chapter describes the use and detailed operation of Data Merger, including creating a merge

scheme and how the data is loaded into a JACAMAR® preview. It is also explained how the data can be edited, assigned or removed and, once the preparation is complete then implemented. All stages are explained and illustrated with screenshots to display the system behaviour.

The Appendix at the end of this guide provides a summary of Type „rule condition‘ expressions, a summary of advanced Column rule regular-expressions constructs and information on capturing groups.

Throughout this guide we will be providing snapshots and examples of the Data Merger functionality based on the „Wine Supply.xls‘ Excel spreadsheet and „WineSupply.jcmr‘ repository as a practical approach to the introduction and use of Data Merger, in line with the Administrator Guide. These files, and related ImportScheme files, are available to download from the following link:

<http://dl.jacamar.biz/Sample/WineSupply.zip>

1.2 The function of Data Merger

Data Merger allows the transfer of structured data from a tab separated file, such as an Excel spreadsheet, into a JACAMAR® repository. The source data of the file is thus written into JACAMAR® based on the design and format of the MetaModel in the repository.

From this perspective Data Merger in JACAMAR® has two consecutive steps:

1. The data from the source file can be recognised and treated as a cascade structure.
2. The elements and data can be mapped directly into the model of the JACAMAR® repository and into the corresponding types, relations and properties.

1.3 Source data

The source data to be merged can be a flat list table or a cascade structure. **The copied area can include column headings, to be ignored later, or by only selecting the data, as shown below.**

This diagram shows a basic cascade whereby column A shows the account numbers, the line below in column B the invoice numbers and the next line down in columns C to F the order details that specifically relate to the corresponding invoice and account.



Source data that is cascaded must be organised in separate columns and lines for it to be recognised.

It is also advisable to ensure that the source data and Meta Model design and format of the data correspond with one another before starting a data merge as it will save time in the long run e.g. Numeric values to correspond with the repository Property data format Integer.





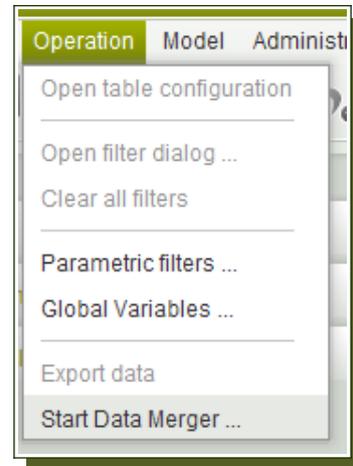
	A	B	C	D	E	F
1	Account No	Invoice number	Order Number	Items	Quantity	Price sold
2	1					
3		1				
4			1	Amarone	6	5,00
5			2	Brunello	6	6,00
6						
7		2				
8			3	Cabernet Franc	4	6,00
9			4	Cabernet Sauvignon	10	5,20
10						
11	2					
12		3				
13			5	Montepulciano	4	5,00
14			6	Fragolino	10	6,00
15			7	Soave	6	5,00
16						
17		4				
18			8	Bianco di Custoza	6	6,00
19			9	Brunello	6	6,00
20						

2 Using Data Merger

2.1 Starting Data Merger

The 'Data Merger' feature can only be enabled by JACAMAR® administrators through the User Administration function within the repository, as explained in chapter 9 of the Admin Guide. The 'Data Merger' role must be assigned to the administrator or user before the menu option will become available.

Data Merger is then opened from the **Operation-> Start Data Merger...** menu option as shown.



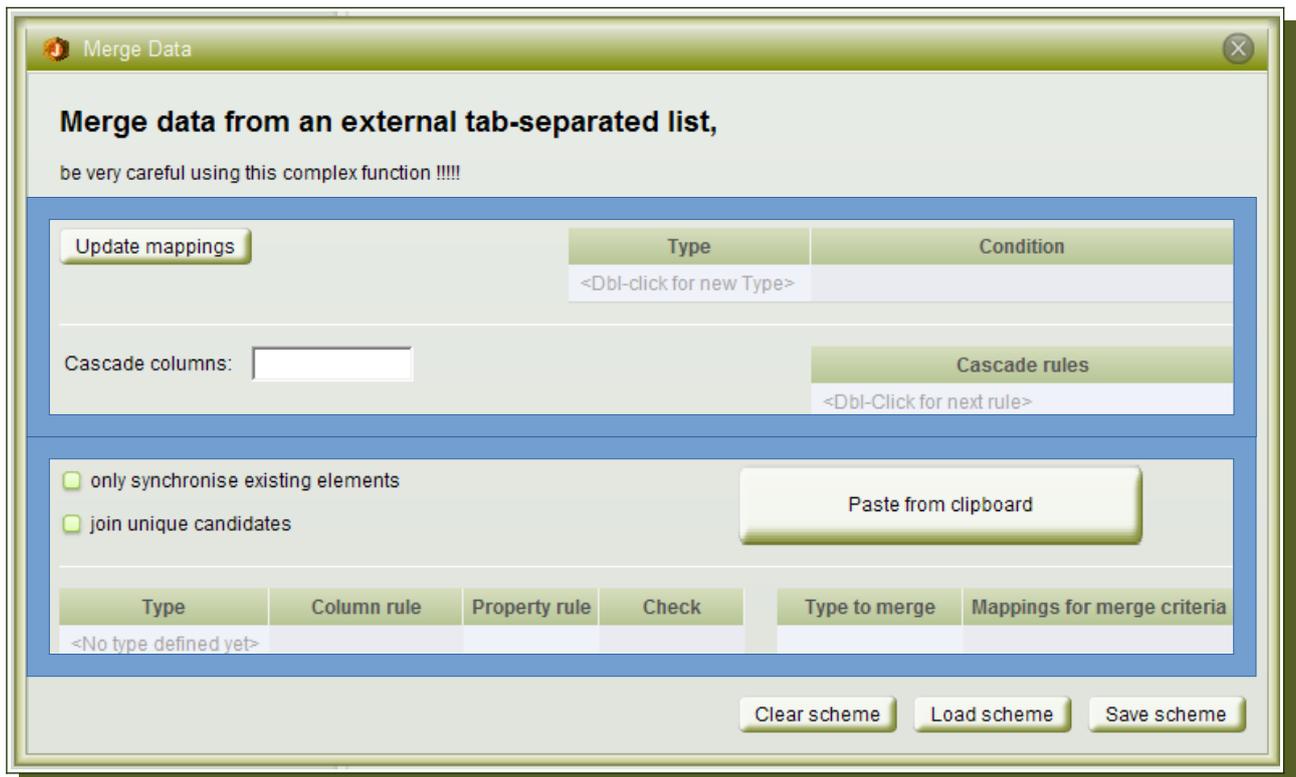
2.2 The Data Merger dialog

The Data Merger dialog is divided into two sections.

The top half is where the Type(s) are defined and the conditions, if any, that need to be met for the merged source data to be recognised, see section 2.3.1.

Cascade structure source data also needs to be defined here whereby the cascaded columns in the source data and the paths of the related lines in sequence must be specified, see section 2.3.2.

The bottom half (referred to as mapping) is where to map the source data columns with their corresponding types, related types and property paths, see section 2.3.3. Each line of the source data is analysed for elements and transferred in accordance with the mapping rules according to the JACAMAR® Meta Model and repository. Merge criteria can also be identified in the adjacent table so matching elements can be consolidated, see section 2.3.6.



The Update mappings button adjusts the rows of the mapping table in the lower section once the Types have been defined.

The only synchronise existing elements check box prevents the installation of new elements therefore only existing relations and elements will be considered.

The join unique candidates check box should be selected for linking a unique element in the presence of a unique candidate to automatically assign them together, see section 2.5.2.

Once the entire Data Merger scheme is completed and after copying the source data, then by clicking on the Paste from clipboard button the data from the clipboard will have the Data Merger criteria applied and be transferred to a merge preview.

Clear scheme allows you to clear the fields in the scheme dialog and start a new scheme.

Load scheme allows you to load a saved scheme.

A defined mapping scheme can be saved as an XML file using the Save scheme button.



2.3 The Data Merger scheme

A scheme is a specific configuration for the data merge process and it determines the column information to be mapped and to which target information. In simple terms, each line from the tab-separated source data file must belong to a JACAMAR® repository Type.

2.3.1 Source data types and conditions

In line with the WineSupply repository and the source data example in section 1.3, there is a cascade structure present. Here, based on the source data, the Meta Model Types are selected from the proposal list, as shown below, and placed in source data cascade sequence with defined conditions, which are manually input.



Type	Condition
Account	A != ""
Invoice	B != ""
Order	<Add a condition. One may be left blank for the
<Dbl-click for new Type>	

The conditions shown here will recognise that when column A != "" (is not empty) it is assigned to Type Account, however if it is empty then the next condition applies therefore when column B is not empty it is assigned to Type Invoice and when both column A and B are empty all other elements in the structure will be considered as part of the last cascade from Type Order, columns C-F in our example.

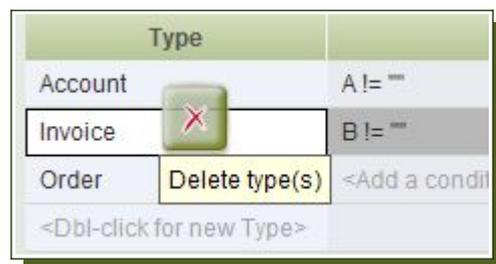
See the Appendix on page 23 for a summary of Type rule condition expressions for example: != (not equal to) and <.. (starts with).



If a flat list source data file is being used then only one Type, with or without conditions, can be defined here (see Wine.importScheme).

To add a new Type or condition simply Dbl-click in the next empty field.

To delete a listed Type simply RMB on the Type (or condition) and select the red cross.



2.3.2 Cascade information about types

The next step for the structure of cascaded source data is to assign the columns and the cascade rules to be applied. In our example cascade information (relationships) are found in columns A-C and must be manually input into the Cascade columns field

The Types and Relations are selected through a drop-down menu, as shown below. The cascade rules follow the base Type, the first Type listed in the conditions table above, so in our example an Account contains Invoices and Invoices contain Orders. These rules must conform to the design of the target Meta Model repository, as shown below.



The Cascade columns and Cascade rules fields only need to be completed when a cascade is present in the source data

Cascade columns:

Cascade rules
Account.invoice
Invoice.order
<DbI-Click for next rule>

Account.invoice

Invoice.

<DbI-Click for next rule>

account

order

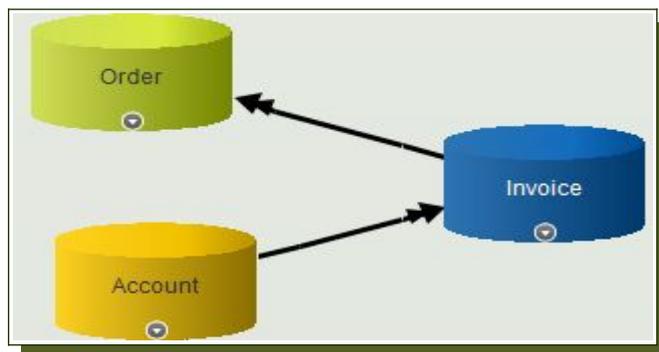
inv #

allElements()

ascending()

average()

▲ Invoice		
account	Account	to-one relation
order	Order	to-many relation
inv #	Integer	
▲ Account		
address	Address	to-many relation
invoice	Invoice	to-many relation



2.3.3 Mapping column information

Based on the Types listed in the top section, and using the Update mappings button, the mapping table will display the Types that are to be recognised from the source data and it is here that the columns and their specific properties and paths must be mapped.

Type	Column rule	Property rule	Check
Account	<DbI-click for new...>		
Invoice	<DbI-click for new...>		
Order	<DbI-click for new...>		

The Column rules are manually input based on the source data columns and the Property rules are configured from the proposal lists using repository navigation paths from the corresponding Type, in line with the results displayed in the source data.



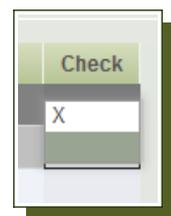
Type	Column rule	Property rule	Check
Account			
	A	acc #	
	<DbI-click for new...		
Invoice			
	B	inv #	
	<DbI-click for new...		
Order			
	C	order #	
	D	wine.variety	
	E	quantity	
	F	selling price	
	<DbI-click for new...		

This diagram from the Wine Supply example shows the mapping table with the property rules aligned to the corresponding columns. When resulting column data belongs to another related Type, as highlighted above in Type Order column D, then the property rule includes the relation from Order to Wine followed by the resulting property (*variety*) to reflect the source data result in line with the Meta Model design.



Any columns copied from the source data that are not listed in this mapping table will be ignored.

If a mapping rule is **only** to be used as a reference and therefore no changes to those elements is to be applied then the **Check** field must be activated for the corresponding column rule by selecting the „X’ from the proposal list, as shown here.



Proposed changes will still be shown in the Merge preview for these columns but when implemented they will remain unchanged in the repository.

Despite cascaded source data being displayed as tab-separated cascades it is possible for the merged data to be displayed in the preview with the cascades in the same column. This is achieved by typing „HIER’ for the cascaded columns, see below, and this changes the display of the merge preview as shown later in section 2.5.1.

Type	Column rule	Property rule	Check
Account			
	HIER	acc #	
	<Dbl-click for new...		
Invoice			
	HIER	inv #	
	<Dbl-click for new...		
Order			
	HIER	order #	
	D	wine.variety	
	E	quantity	
	F	selling price	

2.3.4 Adding and Deleting rules

To add a new rule simply double click on a new line, under the corresponding Type, or by using RMB on the Type or existing rule and selecting the green plus symbol.

And to delete a rule simply RMB on the rule and select the red cross.



2.3.5 Functions in Property rules

When source data elements represent values that are configured in the repository with functions in their column definition path this must also be reflected in the Property rules in the mapping table.

Therefore assuming in our source data file we have an additional column that contains the values 'small' and 'large' to correspond with the Wine bottle size and in the repository these values are configured with the IF statement function in the column definition then the property rule for the column in Data Merger follows the same configuration and navigation path.



	A	B	C	D
2	<u>Variety</u>	<u>Style</u>	<u>Bottle Size /ml</u>	<u>Size description</u>
3	Amarone	dry	75	small
4	Barbaresco	medium-dry	100	large
5	Barbera	sweet	75	small
6	Bardolino	dry	100	large

Additional column in source data with bottle size description

Column definitions

Wine

variety	style	bottle size
column_1	column_2	column_3

```
if(bottle size=75,"small","large")
```

Column definition in the repository is validated using the IF function in the navigation path

Type	Column rule	Property rule
Wine		
	A	variety
	B	style
	C	bottle size
	D	if(bottle size = 75, "small", "large")

Therefore the Property rule for the column is the path recognised in the repository so the values can be mapped against each other.

2.3.6 Regular expressions in Column rules (Advanced)

An advanced Data Merger feature for JACAMAR® users who are familiar with regular-expression constructs is the option to extract a string from within a data source element and merge only that string.

In the original data source values there may be elements that are labeled or recorded differently to the saved repository values i.e. The source data value might read: C_C#20140101_\$/2 and the only part of this string you need to be recognised and merged is the „20140101’, therefore constructs can be written in the Column rules to identify and extract the part of the string that is required. The diagram below shows an example of regular-expression constructs being used to identify a pattern in column A.

Type	Column rule
Account	
	pattern(A, "I+SO (.{8})", 1)

The Appendix on page 23 provides a summary of the regular-expression constructs.



2.3.7 Setting merge criteria and unique assignment

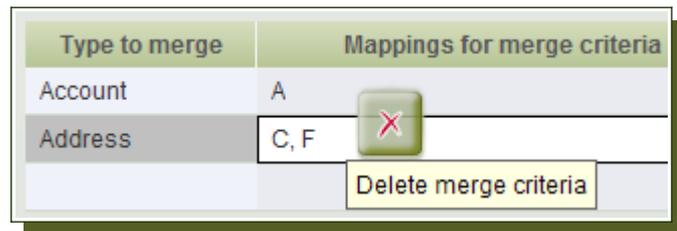
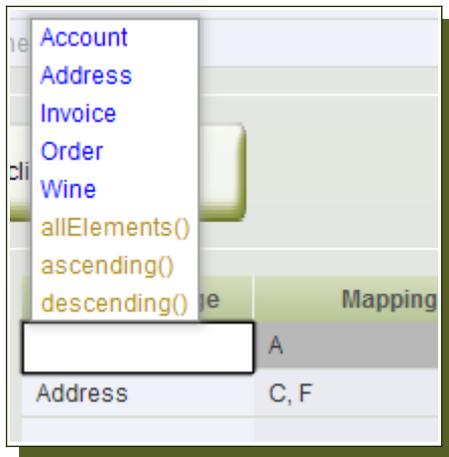
On the basis of clear criteria you can merge mappings from the source data to the target elements. It may happen that information in an individual column is insufficient to provide clear information about the mapping. More column information must be merged to produce a uniqueness. A typical example are the Address properties in the Account-Address.importScheme in the WineSupply example.

Merge criteria is required to prevent new elements being created when matching target elements are present. From the source data we know that each Account number (#) is unique and the corresponding Account name will be mapped with it. The Address properties; street, city and postcode will always map to one another, however the Address category may be different for the same address details but under a different account. Therefore the criteria for Type Address, Properties street and category, need to be merged together, in line with the source data shown below, so that the mapping process looks for an account with a street and corresponding category as a matching candidate.

Type	Column rule	Property rule	Check	Type to merge	Mappings for merge criteria
Account				Account	A
	A	acc #		Address	C, F
	B	name			
	<Dbi-click for new...				
Address					
	C	street			
	D	city			
	E	postcode			
	F	category			
	<Dbi-click for new...				

A	B	C	D	E	F
<u>Account</u>	<u>Company</u>	<u>Address</u>			<u>Address category</u>
Number	Name	Street	City	Postcode	
1	Waldorf Ltd	34 Bexhill St	London	SW5 5TR	Invoice
		3 Station Road	Liverpool	L2 3DE	Delivery
2	Redstar Ltd	34 Bexhill St	London	SW5 5TR	Invoice
		Unit 3 Cowgate	Edinburgh	EH6 3SL	Delivery
3	Mayflower Co.	35a Bridge Avenue	Newcastle	NE12 8TN	Both
4	Harry King & Sons	72 High Street	Birmingham	B4 2CA	Both
5	Goodyears	80 Brown Place	Newcastle	N4 7UB	Both
6	Tippletons Ltd	Flat 3b 160 John Street	Manchester	M18 6JA	Invoice
		3 Station Road	Liverpool	L2 3DE	Both
7	Grapeful Ltd	Fox House Langdon Estate	London	N14 2LS	Invoice

To add a Type to the merge criteria table simply double-click on an empty field and choose a Type from the proposal list. To assign rule criteria to the table select the column rule and, when the mouse cursor hovers on the bottom line of the rule and changes to a hand symbol, drag and drop the rule to the required field in the table. To delete merge criteria simply RMB on the line you wish to delete and select the red cross.



To identify and automatically assign unique matches the join unique candidates check box must be ticked before pasting from the clipboard as detailed in section 2.5.2.

The only synchronise existing elements check box, which when selected, prevents the installation of new elements. Only existing relations and elements will be updated.

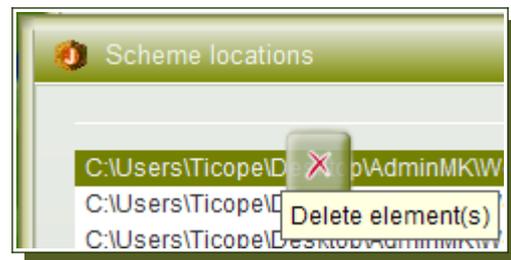
2.3.8 Save the scheme for re-use

Schemes can be saved for re-use and for other JACAMAR® repositories using the Save Scheme button. Once the scheme dialog is complete click on the Save Scheme button and a browser will open for you to select a location and give the scheme a filename.

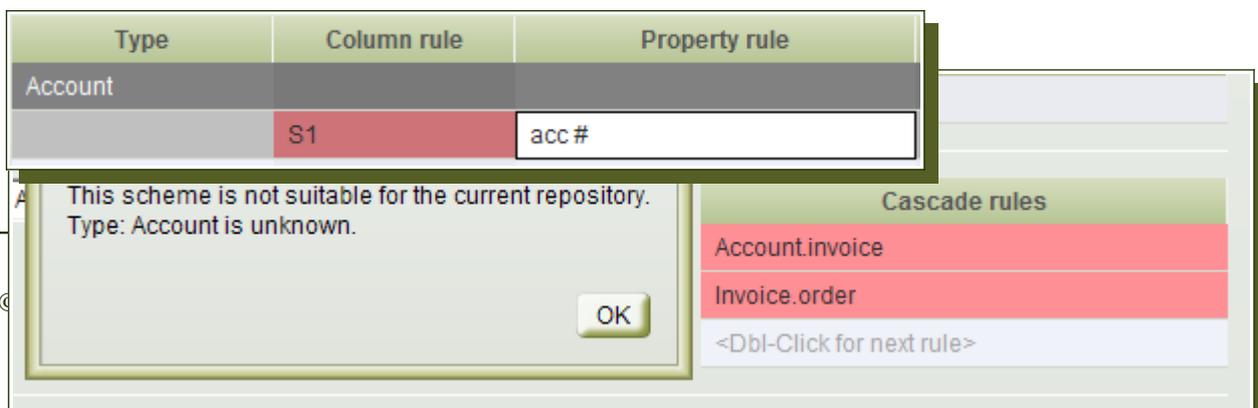
To load a saved scheme click on the Load Scheme button and the *Scheme locations* dialog will open with quick links to the last ten recently used schemes. If your scheme is not on the list click the *Other scheme...* option and a browser will open for you to search for and open your required scheme.

To close the Scheme locations dialog click the red cross in the top right corner and you will be returned to the Data Merger dialog.

To remove an old or redundant scheme from the list simply RMB on the scheme path and select the red cross.



Incorrect assignments are displayed in red when loading a scheme.



When a loaded scheme is not suitable for the repository the following message will appear.

2.4 Performing a data merge

Once the scheme is complete the data merge is done in four steps:

1. Copy the source information
2. Insert on the Data Merger dialog
3. Merge Preview: control and changes
4. Implement and save

Highlight the area in the source spreadsheet and copy to the clipboard

	A	B	C	D	E	F
1	Account No	Invoice number	Order Number	Items	Quantity	Price sold
2	1					
3		1				
4			1	Amarone	6	5,00
5			2	Brunello	6	6,00
6						
7		2				
8			3	Cabernet Franc	4	6,00
9			4	Cabernet Sauvignon	10	5,20
10						
11	2					
12		3				
13			5	Montepulciano	4	5,00
14			6	Fragolino	10	6,00
15			7	Soave	6	5,00
16						
17		4				
18			8	Bianco di Custoza	6	6,00
19			9	Brunello	6	6,00
20						
21						
22						

Paste the clipboard in Data Merger

only synchronise existing elements

join unique candidates

At this stage the data merge scheme is processed and all the rules and criteria applied to the repository and the dialog closes to display a merge preview.



Control of merged elements in the preview and a „corresponding view“

Merge preview X					
hide unchanged		hide ignored		implement changes	
(A) acc #	(B) inv #	(C) order #	(D) wine.variety	(E) quantity	(F) selling price
Account No					
▲ 1					
▲	1				
		#1	Amarone	6	5,00
		2	Brunello	6	6,00
		10	Gavi	5	6,50
		1	Amarone	6	5,0
▲	2				
		3	Cabernet Franc	4	6,00
		4	Cabernet Sau...	10	5,20
▲ 2					
▲	3				
		5	Montepulciano	4	5,00

2.5 The Merge preview

2.5.1 Layout

Menu bar buttons explained

- hide unchanged All unchanged rows will be hidden (without breaking the structure)
- hide ignored All the rows marked *ignored* will be hidden
- implement changes All changes are applied to the repository

The cascade columns and the mapping of the other columns from the source data are displayed in the preview and it is indicated by the background colours, whether

- a mapping was possible
- a clear allocation/assignment was possible (*join unique candidates*)
- an item cannot be assigned
- incompatibilities occur



When 'HIER' is used in the mapping table the preview will appear with the cascades in the specified column, as shown below with cascaded columns A, B and C mapped to „HIER' and therefore displayed in the same column.

(HIER) acc #	(D) wine.variety	(E) quantity	(F) selling price
▲ 1			
▲ 1			
1	Amarone	6	5,00
2	Brunello	6	6,00
▲ 2			
3	Cabernet Franc	4	6,00
4	Cabernet Sauvignon	10	5,20
▲ 2			
▲ 3			
5	Montepulciano	4	5,00
6	Fragolino	10	6,00
7	Soave	6	5,00
▲ 4			
8	Bianco di Custoza	6	6,00
9	Brunello	6	6,00

2.5.2 Merge preview functionality

RMB context menus contain:

- Explanation of the background colour → what will happen when implemented
- Brief information about the possible actions

Colour Coding and actions explained:

White	Changes are present <i>Specific action:</i> option to <u>reset text to ORIGINAL</u> for differences found to a property that you do not want to apply
Grey	No change – still assigned to: Element(ID)



(C) bottle size	(D) unit price
20	2.00
20	
20	
20	

Text change will be applied to: Wine(0)

[Un-assign related element](#)

[Reset text to ORIGINAL: "75"](#)

[Ignore this line](#)

(A) variety	(B) style	(C) bottle si
Amarone	dry	20

No change; still assigned to: Wine(0)

[Un-assign related element](#)

[Ignore this line](#)

Yellow	New element will be created
Red	<p>No match for the element in the cascade and the relation to the element on the related line above will be removed („cut’).</p> <p><i>Specific action:</i> option to keep element which then changes to grey and keeps the element assigned to related line above in the cascade</p>

(A) variety	(B) style	(C) bottle
Sauvignon Blanc	dry	20

No match found. New Wine to be created.

[Ignore this line](#)

2	Brunello	6
---	----------	---

To cut from parent: Order(9)

[keep element](#)

[Ignore this line](#)

Green	<p><u>Auto-join to matched candidate</u> – when the <u>join unique candidates</u> checkbox is ticked and merge criteria is mapped all unique candidates will be automatically assigned to their corresponding match.</p>
Sand	<p>Matching elements are already present which can be assigned in the repository or a new element will be created.</p> <p><i>Specific action:</i> option to Assign to a matching candidate which changes to grey and stays assigned. If multiple matching elements exist then a <please select> drop down list is available for selection</p>



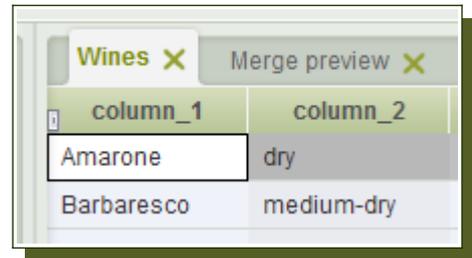
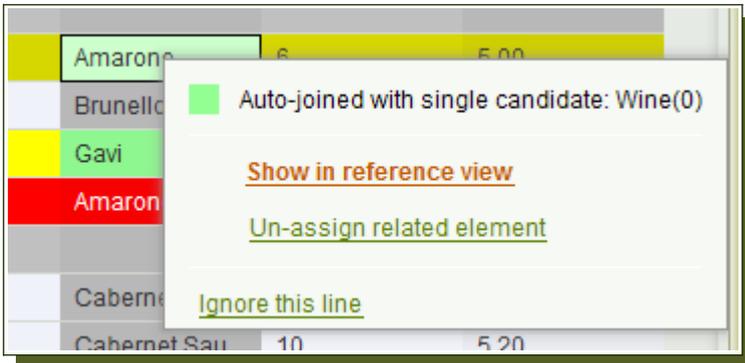
Violet	<p>The line, including its related lines below, and any preview changes to them are ignored</p> <p><i>Specific action:</i> option to <u>Activate line again</u> for Merge preview changes to be possible</p>
Burgundy	<p>Suspect Elements – no assignment is possible</p>

Common RMB actions:

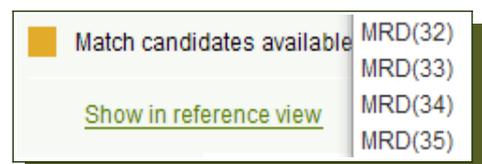
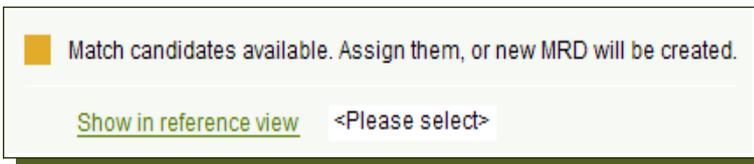
1. Un-assign related element – gives you the option to un-assign the element from its match in the repository, the element then changes to sand colour and the option assign to becomes available. (This can be used to change the assigned element in the repository when multiple matches are available.)
2. Ignore this line – gives you the option to ignore the line, its elements and any Merge preview changes before implementing. The line becomes violet and the option Activate line again becomes available.

Additionally with open views (corresponding views)

- When a matching candidate is found in an open view a further option will appear in the Merge preview context menu → **Show in reference view** and when selected this element is searched and displayed in the corresponding view, as shown below. This option will only appear if the „line items‘ occur as elements in an open view.



- All open views, in accordance with their elements and associated ID, are searched and the result displayed, as shown above: Wine(0). The best mapping is calculated, taking into account in which structure the element „hangs‘ and in the event that an item occurs more than once a drop down list is available to select a matching candidate to show in the open view, as shown below.



Counter-action

For every RMB context menu action there is a kind of counter-action (like an undo)

e.g. If you click reset text to ORIGINAL a new option will appear



for you to Reset text back to the IMPORTED value.

And when keep element is selected then Undo: keep element becomes available:

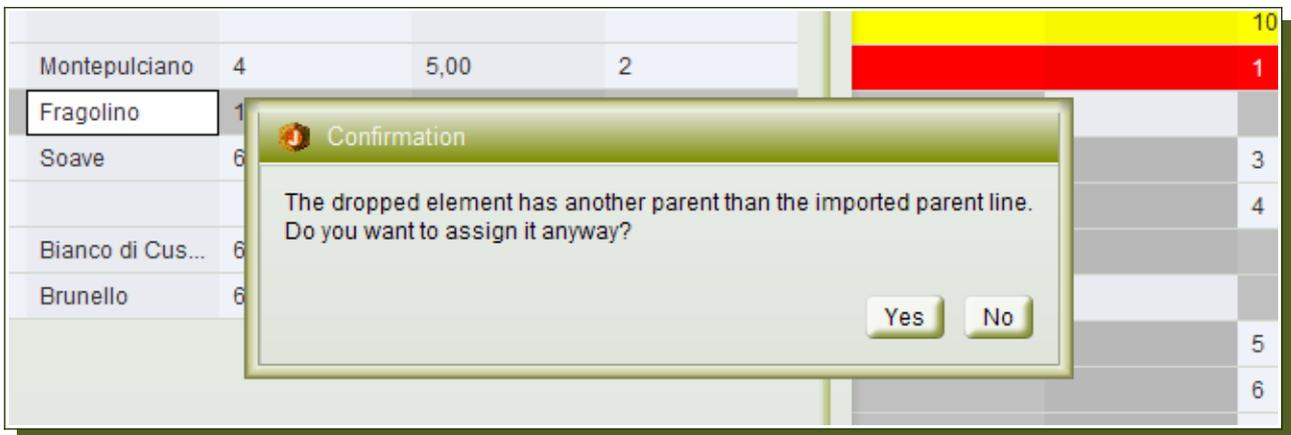
Undo: keep element

Drag and drop

You can drag elements from a reference view to a line in the Merge Preview. This process takes the element (ID) from the reference view and assigns it to the line in the preview where it is dropped. When implemented the line and values in the preview are assigned to the element (ID) and its previous values are replaced. The diagram below shows Wine(0) being dropped and assigned to the new element in the preview. The preview line becomes grey and the element and its values, when implemented, become the properties of Wine(0).

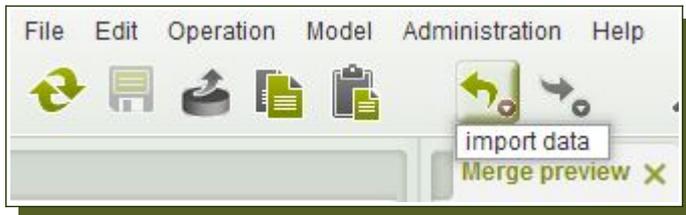


Take precaution when dragging elements directly into cascades or onto elements that have a many to many relation. In instances where the dropped element belongs to a different related line above a warning message will appear, as shown below.



2.6 General Merging Features

- After clicking on the implement changes button the 'undo last action' option in the toolbar becomes active and you are able to undo the entire Data Merger process.



- It is guaranteed that nothing is saved



APPENDIX

Summary of Type rule condition expressions

<i>Expression</i>	<i>Condition requirements</i>
>..	starts with ... (for text)
..<	ends with ... (for text)
>.. <lt;< td=""> <td>includes ... (for strings)</td> </lt;<>	includes ... (for strings)
=	equal to

A'!' in front of the operators above inverts it e.g. „!=‘ for „doesn't equal‘

>=	greater than or equal to
>	greater than
<=	less than or equal to
<	less than
has()	column has a value e.g. has(A) for column A

Summary of regular-expression constructs

<i>Construct</i>	<i>Matches</i>
------------------	----------------

Characters

<i>x</i>	The character <i>x</i>
<i>\</i>	The backslash character
<i>\0n</i>	The character with octal value <i>0n</i> ($0 \leq n \leq 7$)
<i>\0nn</i>	The character with octal value <i>0nn</i> ($0 \leq n \leq 7$)
<i>\0mnn</i>	The character with octal value <i>0mnn</i> ($0 \leq m \leq 3, 0 \leq n \leq 7$)
<i>\xhh</i>	The character with hexadecimal value <i>0xhh</i>
<i>\uhhhh</i>	The character with hexadecimal value <i>0xhhhh</i>
<i>\t</i>	The tab character (' <i>\u0009</i> ')
<i>\n</i>	The newline (line feed) character (' <i>\u000A</i> ')
<i>\r</i>	The carriage-return character (' <i>\u000D</i> ')
<i>\f</i>	The form-feed character (' <i>\u000C</i> ')
<i>\a</i>	The alert (bell) character (' <i>\u0007</i> ')
<i>\e</i>	The escape character (' <i>\u001B</i> ')
<i>\cx</i>	The control character corresponding to <i>x</i>

Character classes

[abc]	a, b, or c (simple class)
-------	---------------------------



[^abc]	Any character except a, b, or c (negation)
[a-zA-Z]	a through z or A through Z, inclusive (range)
[a-d[m-p]]	a through d, or m through p: [a-dm-p] (union)
[a-z&&[def]]	d, e, or f (intersection)
[a-z&&[^bc]]	a through z, except for b and c: [ad-z] (subtraction)
[a-z&&[^m-p]]	a through z, and not m through p: [a-lq-z] (subtraction)

Predefined character classes

.	Any character (may or may not match line terminators)
\d	A digit: [0-9]
\D	A non-digit: [^0-9]
\s	A whitespace character: [\t\n\x0B\f\r]
\S	A non-whitespace character: [^\s]
\w	A word character: [a-zA-Z_0-9]
\W	A non-word character: [^\w]

Boundary matchers

^	The beginning of a line
\$	The end of a line
\b	A word boundary
\B	A non-word boundary
\A	The beginning of the input
\G	The end of the previous match
\Z	The end of the input but for the final terminator , if any
\z	The end of the input

Greedy quantifiers

X?	X, once or not at all
X*	X, zero or more times
X+	X, one or more times
X{n}	X, exactly n times
X{n, }	X, at least n times
X{n, m}	X, at least n but not more than m times

Reluctant quantifiers

X??	X, once or not at all
X*?	X, zero or more times
X+?	X, one or more times
X{n}?	X, exactly n times
X{n, }?	X, at least n times
X{n, m}?	X, at least n but not more than m times

Possessive quantifiers



$X?+$	X , once or not at all
X^*+	X , zero or more times
$X++$	X , one or more times
$X\{n\}+$	X , exactly n times
$X\{n, \}+$	X , at least n times
$X\{n, m\}+$	X , at least n but not more than m times

Logical operators

XY	X followed by Y
$X Y$	Either X or Y
(X)	X , as a capturing group

Groups and capturing

Capturing groups are numbered by counting their opening parentheses from left to right. In the expression $((A)(B(C)))$, for example, there are four such groups:

- 1 $((A)(B(C)))$
- 2 (A)
- 3 $(B(C))$
- 4 (C)

Group zero always stands for the entire expression.