## Introduction:

My motivation in writing this document was to provide beginner to intermediate level tips on managing those manual data conversion / transformation tasks that are usually managed using Excel. It is not unusual for these tasks to expand and take over more time than was budgeted at the time the task was identified.

## The problem:

The story starts with a ERP implementation project and invariably includes that other system that was not scoped out because the e-commerce processes were deemed complex to build in the first roll out of the global solution.

There is an example of a file which provides an e-commerce system that needs the mapping of the Legacy customer numbers to the ERP customer numbers in the future state. This information will be stored in a table on the e-commerce server and is needed because e-commerce processes were not implemented on the ERP system and it was not implemented on the ERP system because … you get the picture.

The folks on the Data team have managed to map Legacy customer data over to the ERPP system. The crew from the e-commerce side is interested in continuing the seamless experience to the customer and therefore need the map from the Legacy customer to the SAP customer. This information is then assigned to the web user so that any given user can enter orders for their organization.

Table 1 File header for e-commerce system

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **UserName** | **CurrBillTo** | **CurrShipTo** | **SoldTo** | **ShipTo** | **BillTo** | **Payer** |
| Jim\_Bob\_ Davis | 894567 | 0 | Blank | Blank | Blank | blank |

Table 2 File header for customer data from ERP system

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CNUM** | **NAME** | **AGRP** | **SGRP** | **PDIV** | **CHL** | **LNAME** | **LNUM** | **LDIV** | **YY\_CCDB\_NUM** | **CRDAT** |
| 205001 | HALLS STORE LLC | ZPST | 2223 | 10 | 10 | HALLS STORE LLC | 1013510 | 94 | 101351 |  |

So far it looks simple enough for a consultant to pull two Excel files at the least and start putting together pivot tables and other such to compile an e-commerce map that has 132,000 entries on it. Alas, if it were that simple it would be the last thing the consultant would do before closing shop on Friday. So what makes this problem interesting and questionable as a manual task? I will get to that in short order.

To begin with the customer number on the e-commerce side is one or more of the ‘CurrBillTo’ or ‘CurrShipTo’ columns. For example, a customer that embodies a buying role, a ship to location and a payer role would be based off the ‘CurrBillTo’ customer. However, many of the retail locations are managed as ship to customer locations and these are considered a combination of a ‘Bill-To’ number and an increment in the ‘Ship-To’ number. So the Legacy customer number is really ‘8945670’ if it is a ‘complete’ customer id or it is for example ‘894567100’ if it is the 100th retail location for this customer. Further a ‘complete’ customer that embodies the sold-to, ship-to, bill-to and payer roles will have the same ERP customer number for all of the roles. On the other hand, ship-to only locations will be maintained as ‘ship-to’ customers in the ERP system. You are probably starting to think of joins already. Consider that we have two Excel csv files that we start with.

Is it likely that when all is said and done more rules on matching and joining the data sets would have been deliberated, processed and implemented. So what is a good tool? There are many open source tools and licensed ones in the market. I will be explaining the approach using an open source tool, it was open source at the time I prepared the document.

The tool I was chosen for this exercise is ‘KNIME’. KNIME stands for **K**o**n**stanz **I**nformation **M**in**E**r and is pronounced: [naim] (that is, with a silent "k", just as in "knife"). It is developed by KNIME.com AG located in Zurich and the group of Michael Berthold at the University of Konstanz, Chair for Bioinformatics and Information Mining. According to Wikipedia from the web page ‘https://en.wikipedia.org/wiki/KNIME’, it is an open source data analytics, reporting and integration platform. A graphical user interface allows assembly of nodes for data preprocessing (ETL: Extraction, Transformation, Loading), for modeling and data analysis and visualization.

KNIME allows users to visually create data flows (or pipelines), selectively execute some or all analysis steps, and later inspect the results, models, and interactive views. These are also called workflows.

This document is not an exposition of the KNIME tool, instead I will focus on how a workflow can be setup for a simple to medium complex use case.

## Workflow / Data Pipeline:

The following flow diagram is the pipeline that I have created to process the use case.

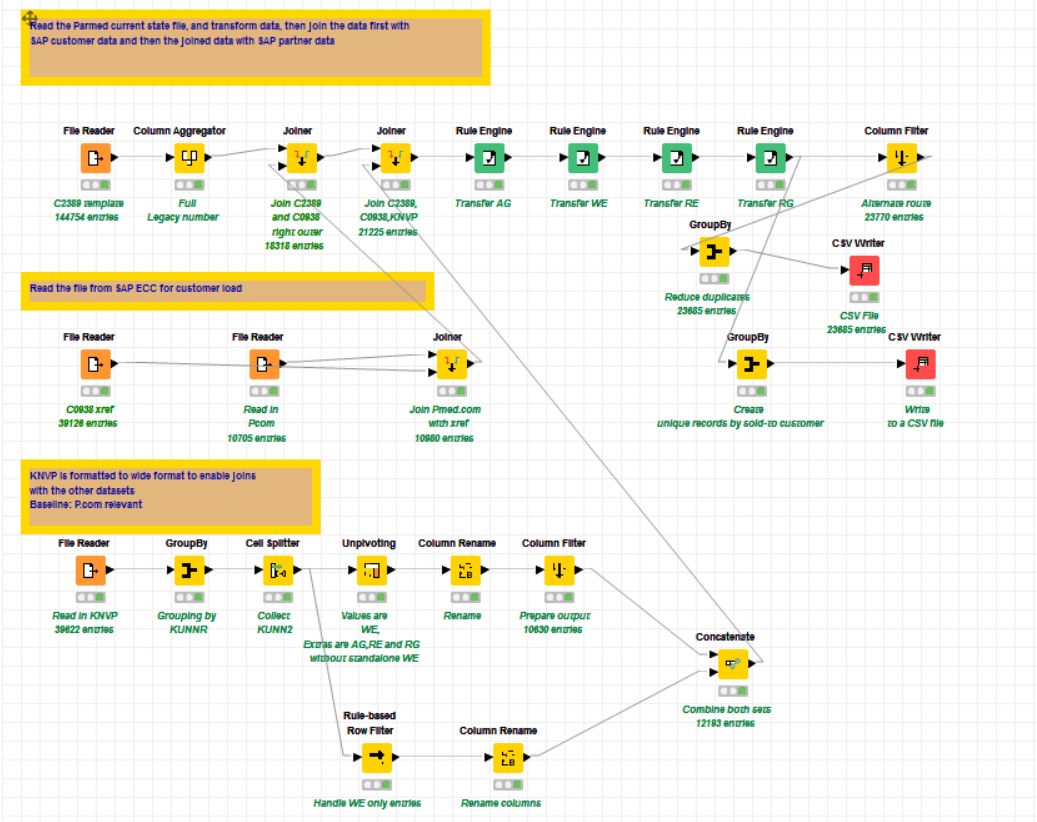


Figure Data pipeline to generate cross reference file automatically

I will go through the various elements of this workflow next.

### FileReader

*FileReader* will read in data from an ASCII file or a URL. Various typical options as well as more advanced ones are available to drive the storage of data.

Here I read in the comma delimited data with a header.

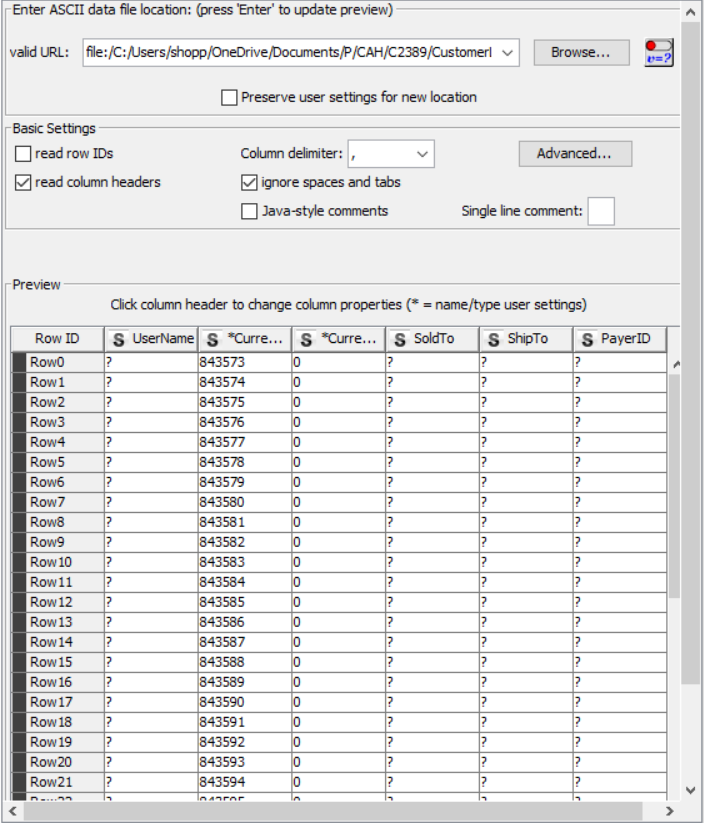


Figure Configuring the file reader

Various options are available such as quote control, missing value mapping, decimal value separators and so on. The result of this process is that data is read from the file into memory. Another file reader is used to store the ERP customer data into memory.

### Column Aggregator

One of the transformations that we have to do with the e-commerce data is the generation of the Legacy customer number. For this we use the *Column Aggregator.* The aggregator combines the columns holding the Bill to and Ship to numbers to generate the Legacy customer number column.

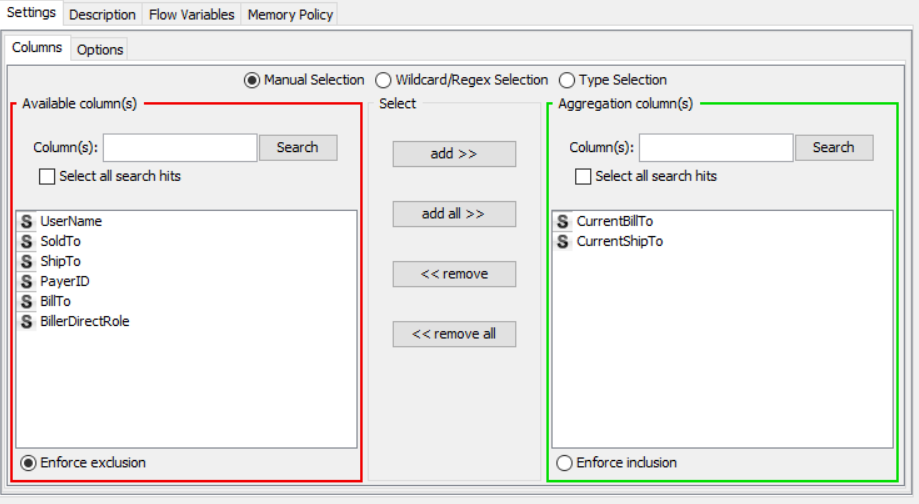


Figure Column Aggregation to generate Legacy customer numbers

Here is an excerpt of the output.

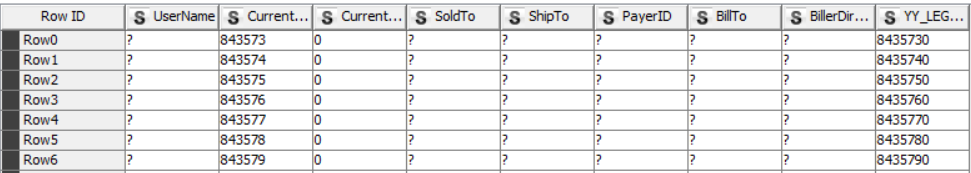


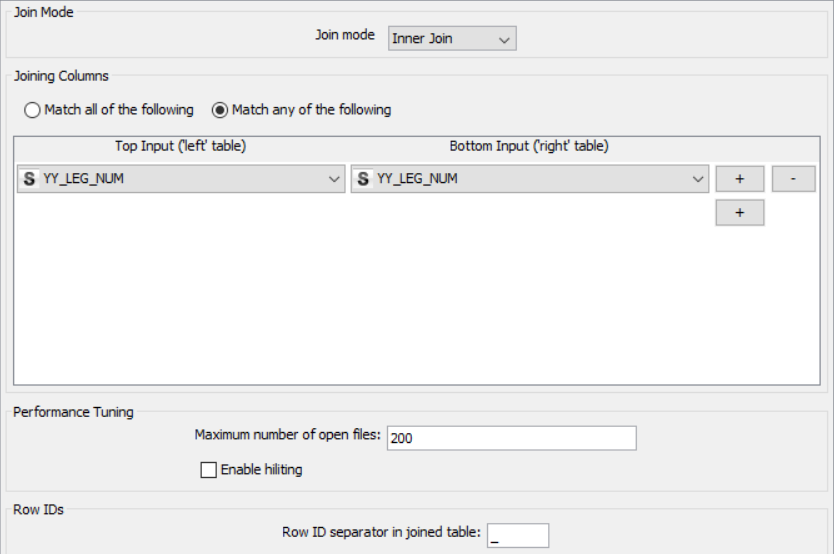
Figure Excerpt of the Column Aggregator process

The utility of these pipeline tasks lies in the fact that each process holds its own copy of the operation that it was tasked with. So, if you need to change a subsequent process you do not typically need to change upstream steps unless the data model or rules have to be changed.

### Joiner

The next step in the process is to consolidate the data or merge the data from the e-commerce and erp data sets. We use the *Joiner* step for this purpose. The join or merge process is based on columns in both data sets.

Various join options are available; I will be using the ‘inner join’. Admittedly this is not the best choice, but keep in mind that I have created an example. In reality one would need to possibly consider a ‘left-outer’ join. I leave it to the readers to evaluate the options.



The result of this operation generates a data set like the one below.

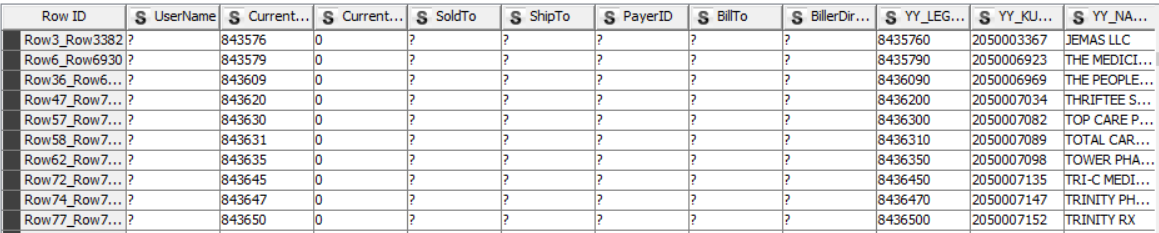


Figure Joined data set based on Legacy customer number

There is a neat little trick that KNIME uses to identify the rows from the two data sets that were merged. See if you can locate them.

Let us take quick stock of where we are. We have managed to merge the two data sets. Good work. What remains to be done in our example is to transfer the ERP customer number in the column ‘YY\_KUNNR’ to the Sold-to, Ship-to, Bill-to and Payer columns based on the customer role. The customer role is defined by values in the column ‘YY\_KTOKD’.

### Rules Engine

In this *Rules Engine* step, we create rules that are used by the tool to evaluate every row from the joined dataset and make changes to the corresponding column values.

Here is an example of a rule that updates the ‘Sold to’ column. The rule bases its update on the value of the customer role grouping from the ERP system. If there is a match the value of the ERP customer is transferred to the ‘Sold-to’ column of that row.

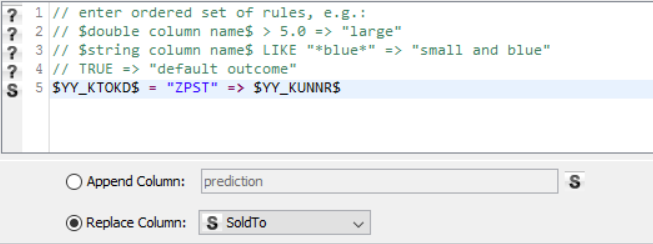


Figure Rule to update Sold-to column

Here is an example of the results.

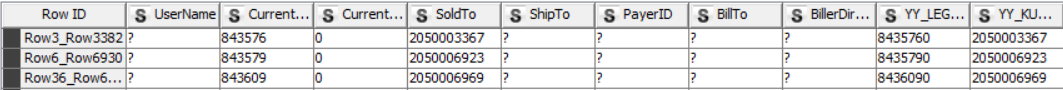


Figure Example of updated Sold-to column

Similar rules are defined to update the Ship-to, Bill-to and Payer columns for the ‘Sold-to’ customer. A separate rule is defined to update the ‘Ship-to’ column only for ‘Ship-to’ customers.

The end result looks like the following list.

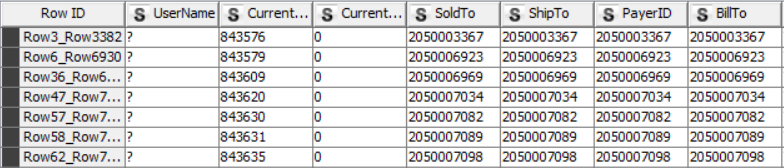


Figure Updated dataset

## Read SAP File

The SAP xref file follows the following template and includes all of the customers loaded into SAP ECC.

Figure Template for customer conversion



### Read SAP relevance for BU

The reference is the table for sales area, we use this to restrict our dataset to BU relevant customers.

Figure Restrict to BU relevant customers



## Join customer legacy cross reference with BU relevant customers

We use a left outer join to focus on BU relevant customers.

Figure Combine cross reference with BU relevant information

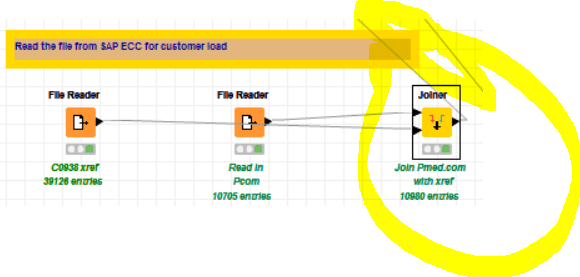
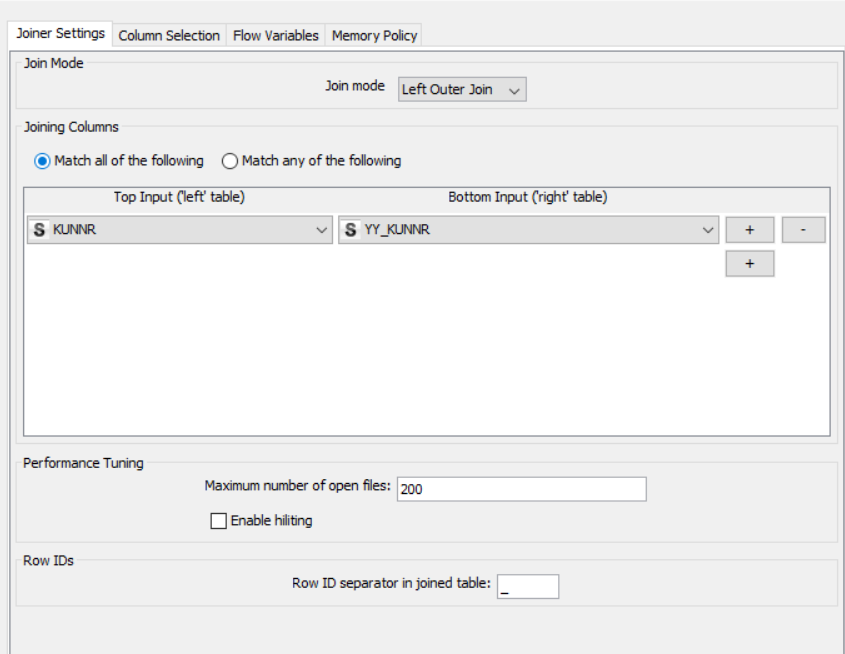


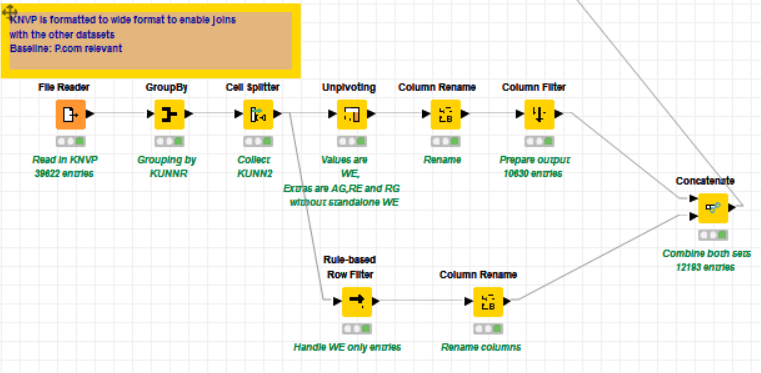
Figure Logic for joining the tables



## Using customer partner functions

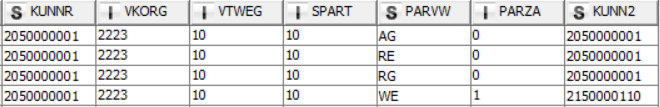
The BU relevant customers have to be listed along with the business partner roles for the sold-to customers. We start by extracting partner information.

Figure Partner workflow



The SAP partner table provides rows for each partner role by customer.

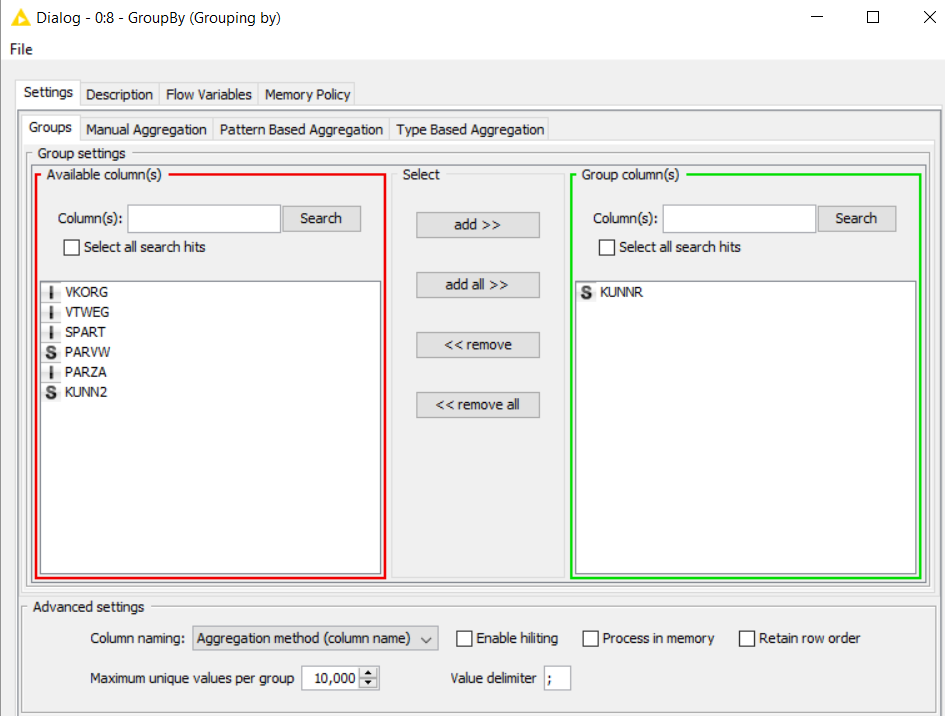
Figure SAP partner role example



Our next task is to provide roles per customer, by transposing the rows into columns.

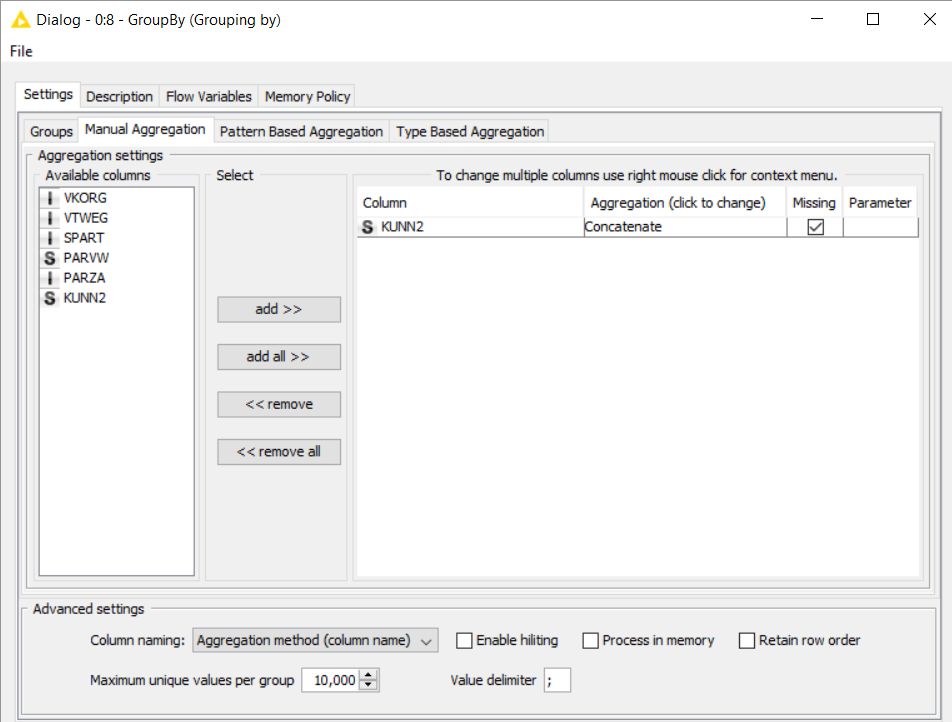
We start by grouping our partner entries by customer.

Figure Grouping by customer

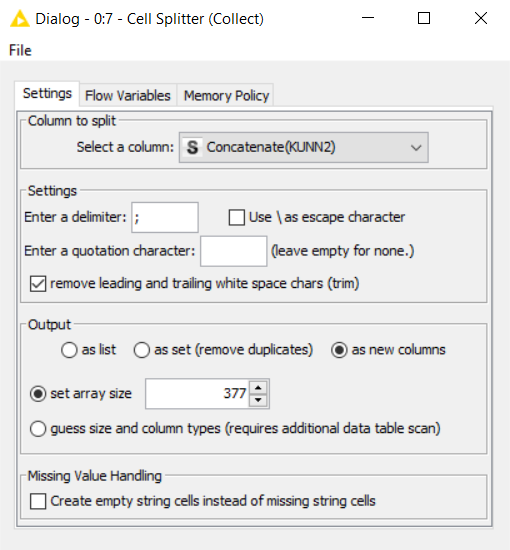


We collect all partner roles per customer.

Figure Collect partner roles per customer



Following the collection of roles per customer, we split the collection into individual columns.



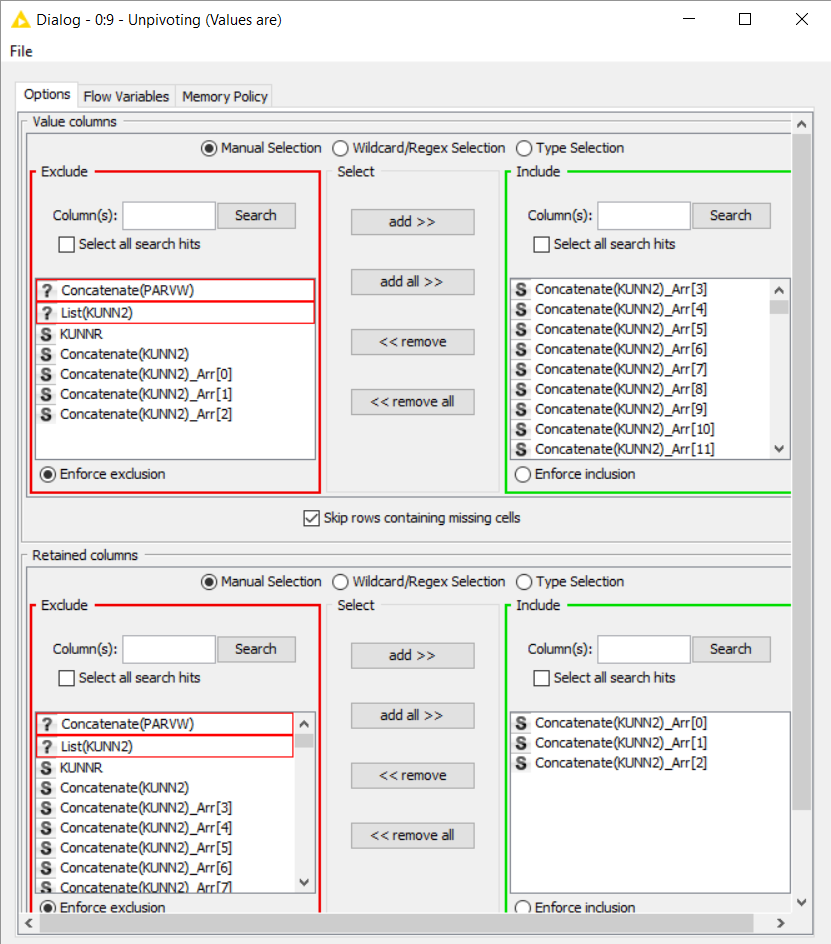
This results in a single row per customer.

Figure Result of the split function



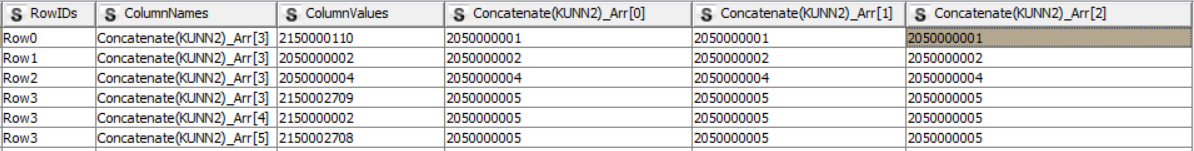
We then take our result from the rows and un-pivot the rows, holding onto the “AG”,”RE” and “RG” roles which seldom change.

Figure Unpivot the partner roles grouped by customer



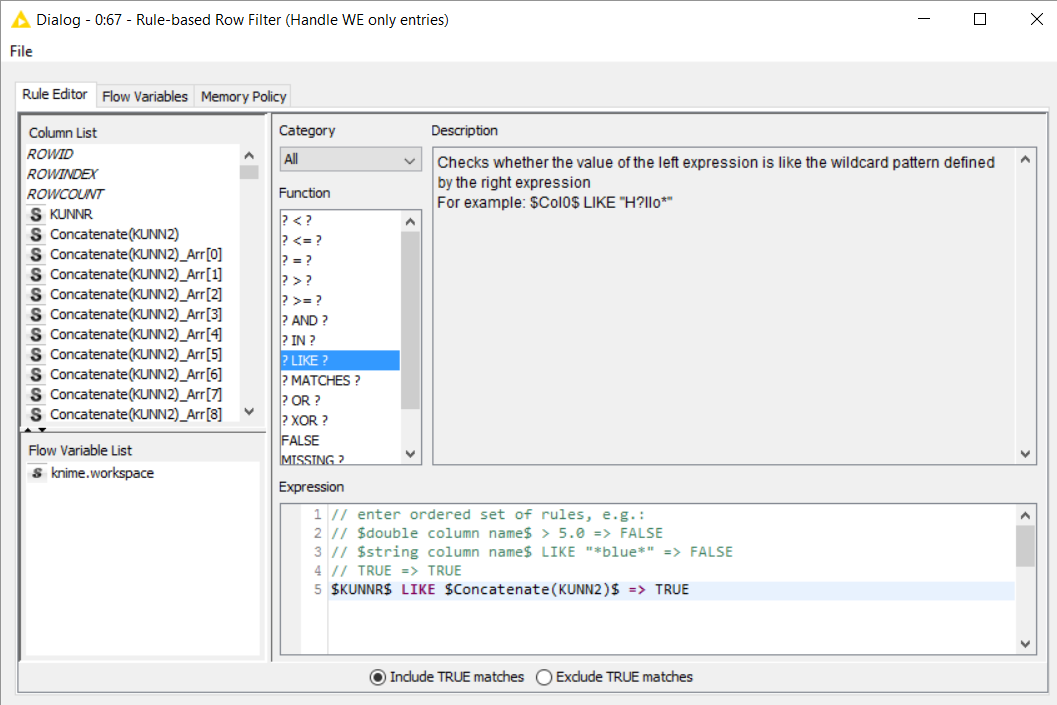
The following is an example of the un-pivoting process.

Figure Example for un-pivot



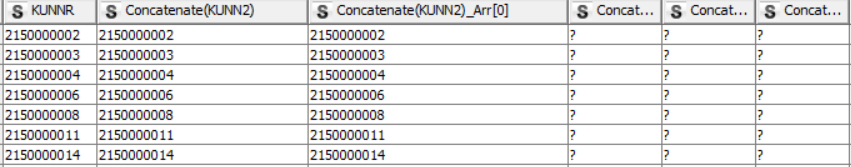
In performing this step, however we have excluded customers which are standalone ship-to parties. For these we take a subset of the collected partner roles and store customer numbers.

Figure Restrict to customers which are standalone ship-to customers



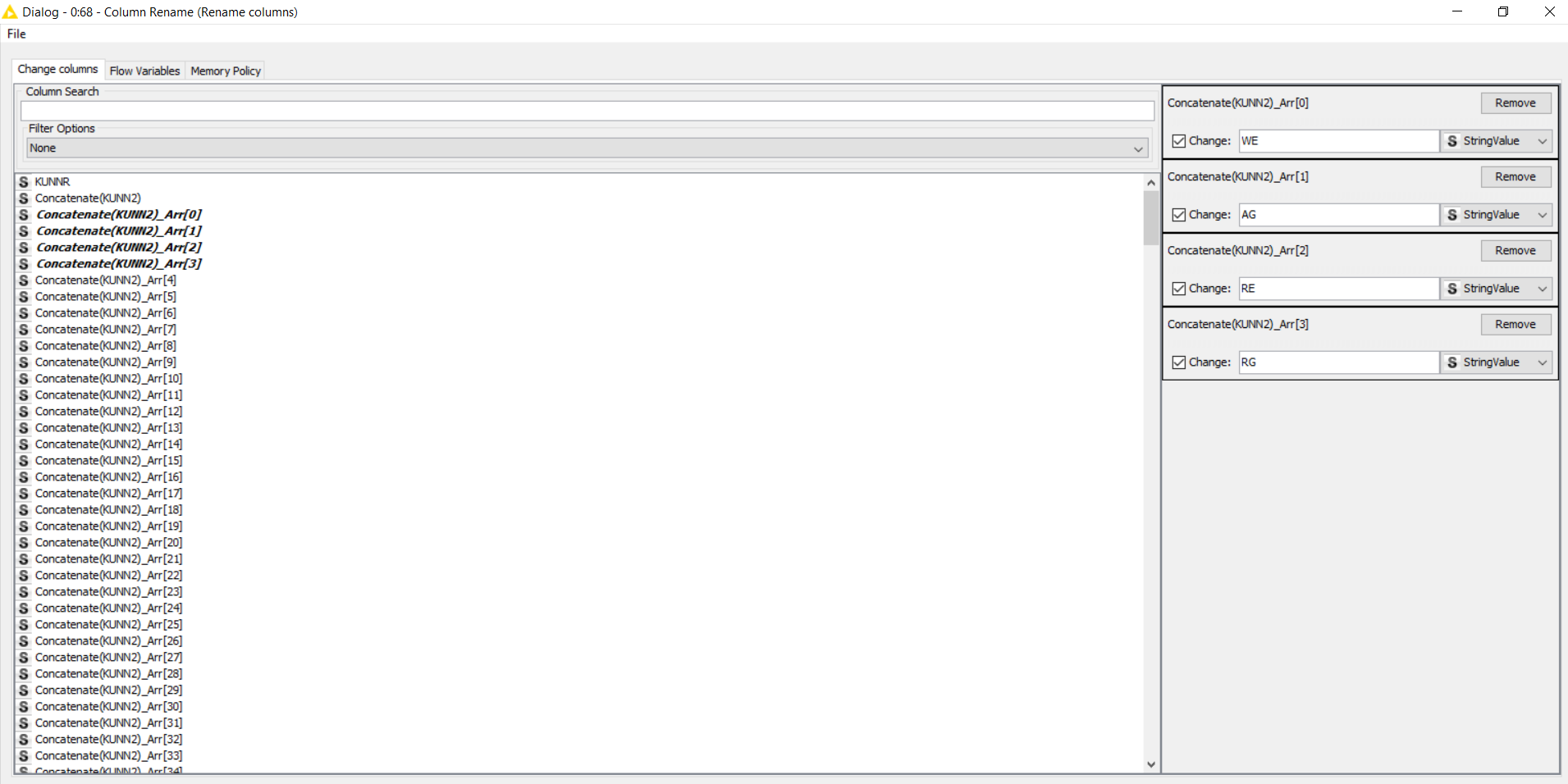
Here is an example of the result.

Figure example of the standalone ship-to customers



A step that is performed before merging both sets of data is to rename columns.

Figure Rename columns from the un-pivoted sets



A column filter is useful in restricting the output format to specific standards.

Figure Column filter to restrict to partner roles

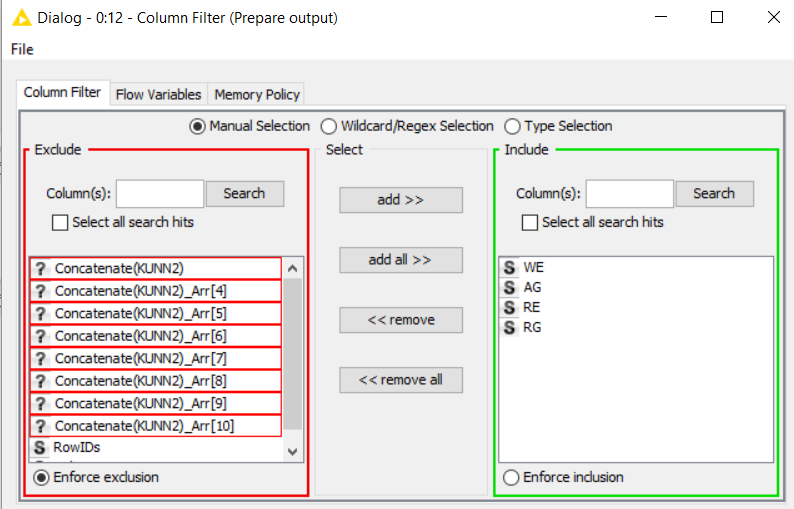
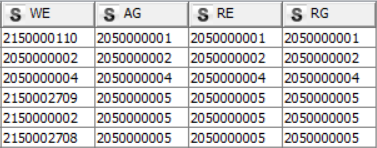
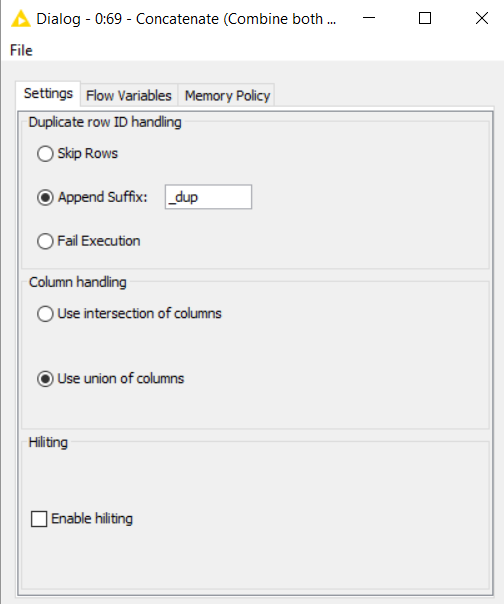


Figure Example of output format



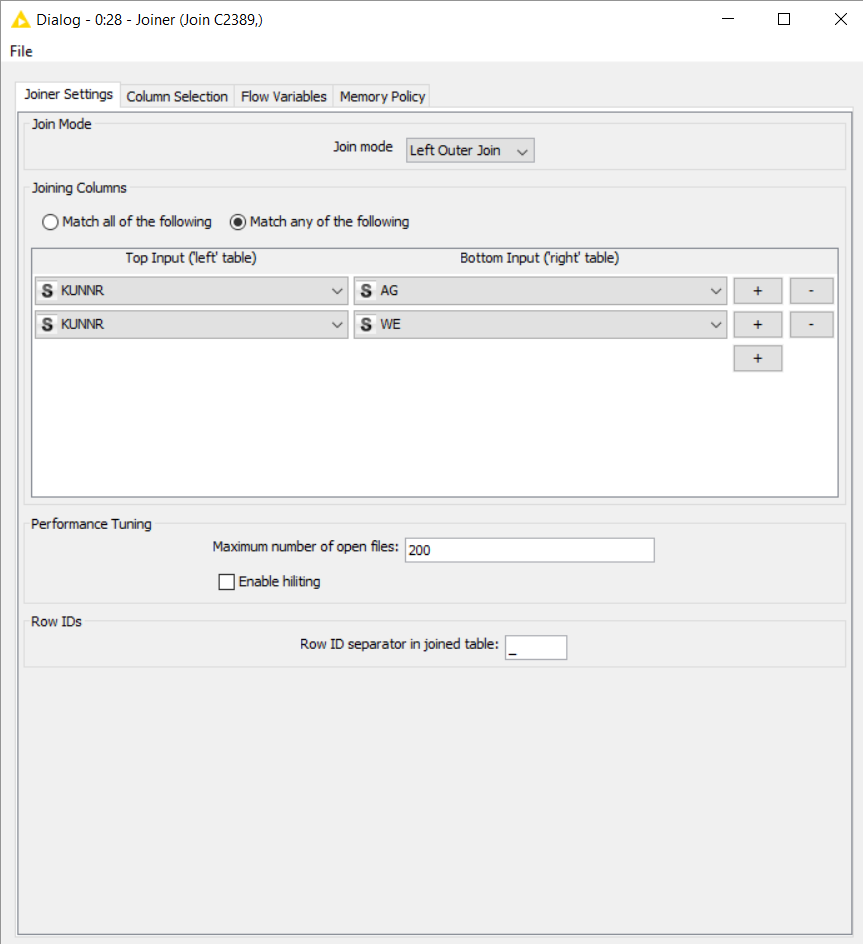
Combining the results is achieved via column concatenation.

Figure Column concatenation to merge both datasets



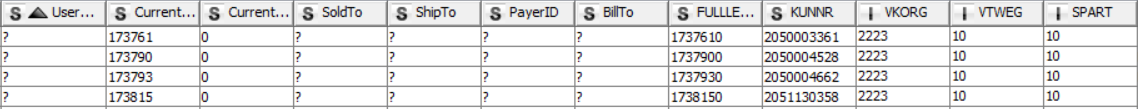
Following the collection of partner roles, we merge data from the user id table, the customer and partner tables. Since sold-to customers and standalone ship-to customers exist, a merge is based on both rules.

Figure Merge user id, customer and partner data sets



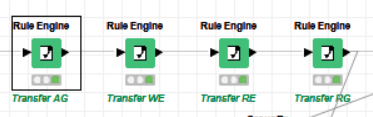
An example of the result is shown.

Figure Example of a joined set



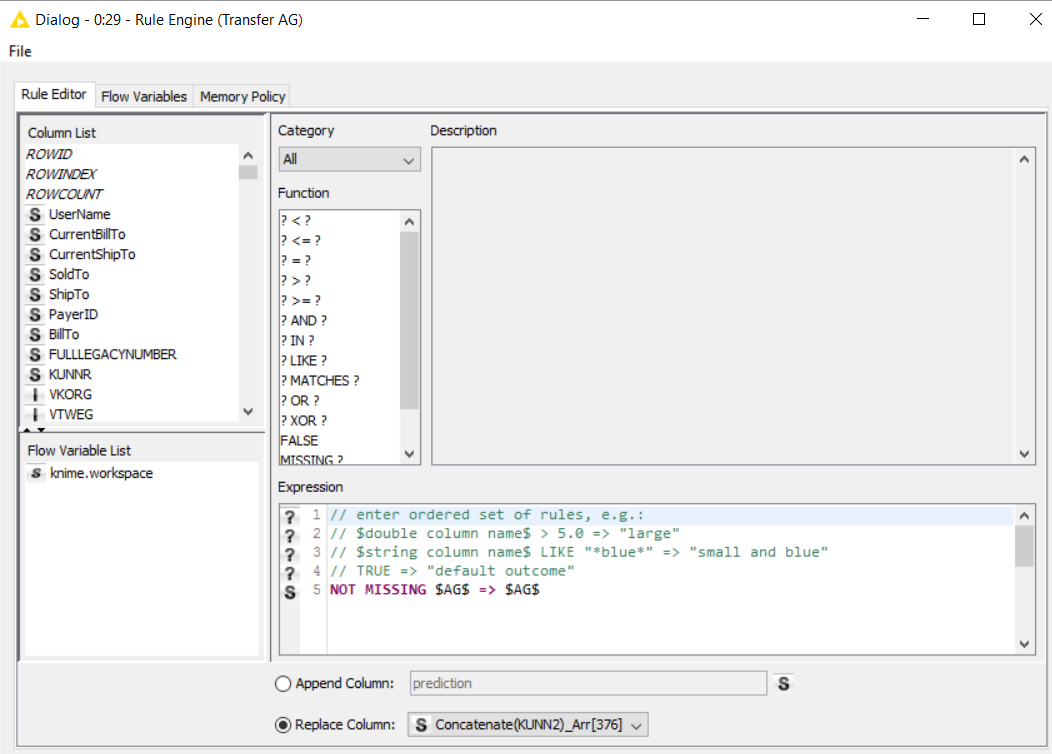
The above set will require partner numbers to be transferred.

Figure Rules to transfer partner roles to user id set



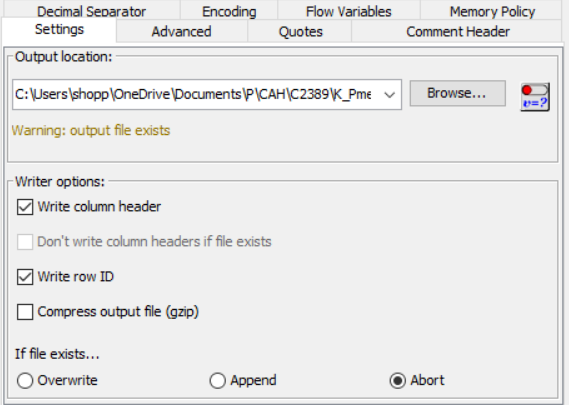
An example is shown for the sold-to party, the same principle is repeated for the remaining partner roles.

Figure Transfer partner roles



### CSV Writer

Finally, we would like to create another csv file to pull the merged updated dataset and provide it to the e-commerce team for loading into the e-commerce server. The *CSV Writer* is used to complete this task.



Various options are available to customize the output file.

## Concluding remarks

I have used production scale data sets in this example, admittedly your production scale need not be the same as mine. Let me provide some sense of the scale here. The e-commerce file had 130,000 rows. The ERP file only had 9000 records. The workflow performed without hiccups. I would recommend some solid validation to confirm robustness of the transformation. What I had set out to do was to show an easy way to handle tasks that could easily end up consuming ten times more time than originally budgeted. This workflow is scalable and can be replicated. My review of the task types available confirms the availability of many more kinds of tasks particularly ones of the input, output and transformation kind. I leave it to the user to explore more of the options.

## References

@INPROCEEDINGS{BCDG+07,

author = {Michael R. Berthold and Nicolas Cebron and Fabian Dill and Thomas R. Gabriel and Tobias K\"{o}tter and Thorsten Meinl and Peter Ohl and Christoph Sieb and Kilian Thiel and Bernd Wiswedel},

title = {{KNIME}: The {K}onstanz {I}nformation {M}iner},

booktitle = {Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)},

publisher = {Springer},

ISBN = {978-3-540-78239-1},

ISSN = {1431-8814},

year = {2007}

}

#### Footnotes[¶](https://render.githubusercontent.com/view/ipynb?commit=fa38f0fd2134f1076388fc901a018f021556fba2&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f73686f7070657276616e2f522d4c616e67756167652f666133386630666432313334663130373633383866633930316130313866303231353536666261322f446174612e4578706c6f726174696f6e2e7573696e672e522e6970796e62&nwo=shoppervan%2FR-Language&path=Data.Exploration.using.R.ipynb&repository_id=52122510#Footnotes)

1) author = {Michael R. Berthold and Nicolas Cebron and Fabian Dill and Thomas R. Gabriel and Tobias K\"{o}tter and Thorsten Meinl and Peter Ohl and Christoph Sieb and Kilian Thiel and Bernd Wiswedel},

title = {{KNIME}: The {K}onstanz {I}nformation {M}iner},

booktitle = {Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)},

publisher = {Springer},

ISBN = {978-3-540-78239-1},

ISSN = {1431-8814},

year = {2007}

}

2)Copyright © 2016 by Eswar Raman