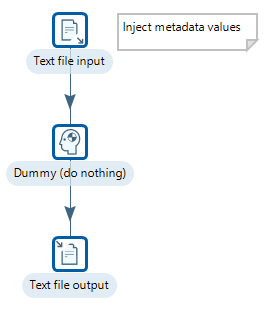
Exercise 4-1-6: Retail Sales

|  |  |
| --- | --- |
| Introduction | The Transformation Executor enables dynamic execution of Transformations from within a transformation. This allows you to easily create a loop and send parameter values or even streams of data to the (Sub) Transformation. Originally this was only possible at a Job level. |

|  |  |
| --- | --- |
| Objectives | In this Exercise, you will define 3 Transformations:   * **Master** Transformation that reads the metadata store and sends the group of records relevant for each input file to the Metadata Injection Transformation. * **Metadata Injector** Transformation which receives the metadata for each file and injects it into the template transformation. * **Template**: ETL workflow: The Text File Input loads the files, based on the injected Metadata properties, and Outputs the results. The required settings have been ‘mapped’ in the ETL Metadata Injection Step |

Step 1 – Template

The template provides the ETL workflow appending the two data sources (sales & stock).



Text file output

* Just enter the path for the file ouput and select append. No fields have been specified to ensure output of any of the fields.
* Save Transformation: tr\_template.ktr

Step 2 - Master Transformation

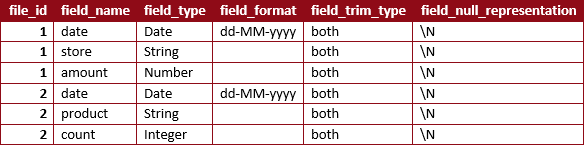
Text file inputs

The Metadata Injection is comprised of:

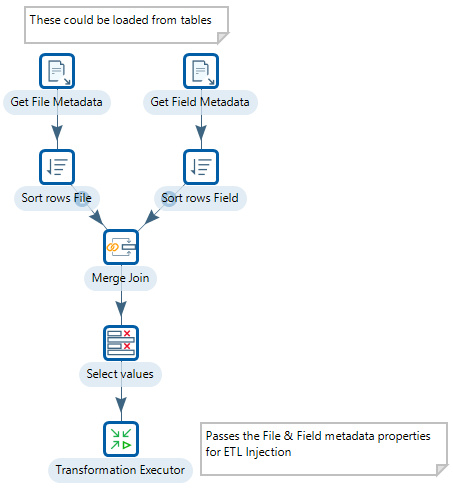
* File Metadata: Concerned with location of the Sales and Stock files, their file\_type, whether a has\_header row is present, etc..



* Field Metadata: Injects the metadata properties; field\_name, data\_type, etc..

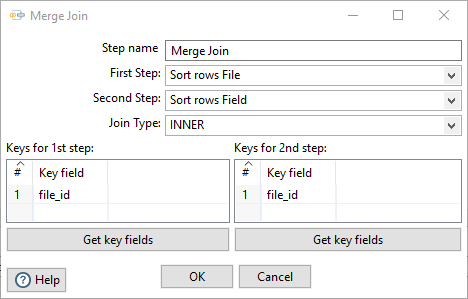


* Save the Transformation as: tr\_master.ktr



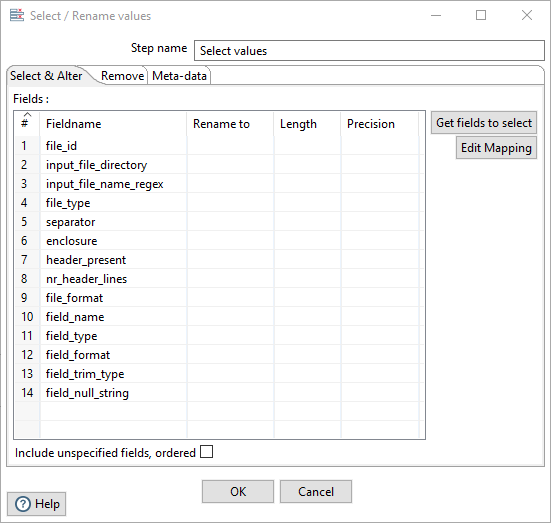
Merge Join

* Merge join on the file\_id



Select Values

* Select all the fields



* You will need to remove the replicated file\_id

Transformation Executor

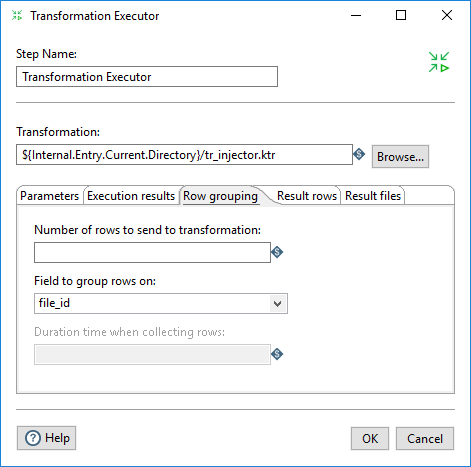
The Transformation Executor step allows you to execute a Pentaho Data Integration (PDI) transformation. It is like the Job Executor step, but works with Transformations. By default, the specified Transformation will be executed once for each input row. You can use the input row to set parameters and variables. The Executor step then passes this row to the Transformation in the form of a result row.

You can also pass a group of records based on the value in a field, such that when the value changes, the specified transformation is executed. In these cases, the first row in the group of rows is used to set parameters or variables in the transformation.

You can also launch multiple copies of this step to assist in parallel transformation processing.

1. Set the Sub -Transformation to Run

${Internal.Entry.Current.Directory}/tr\_injector.ktr



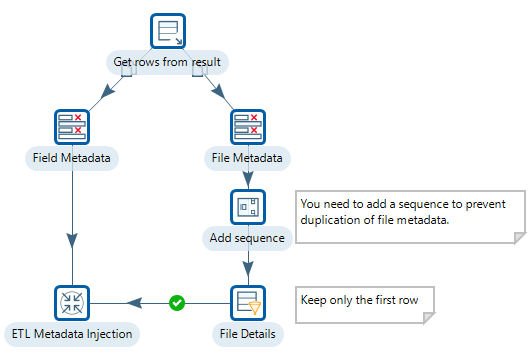
1. Select the Row grouping tab
2. From the drop-down box select:

file\_id

By setting the field to group the resultset on, each row from each data source

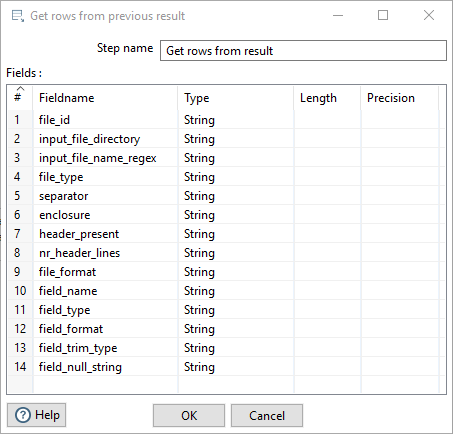
Step 3 - Metadata Injection

The Transformation will be executed several times by the Master Transformation.



Get rows from previous step

This step returns rows that were previously generated by another transformation, i.e. rows are streamed from the Master Transformation.



* Copy and paste the metadata property fields from the Select Values step of the Master Transformation

The output of the Get rows from result step is copied to two streams:

* One which will only keep the Field Metadata
* Another, the File Metadata

As the file metadatais duplicated (each field will be duplicated for each datasource – Sales & Stock), it makes sense to ensure that the file metadata is only sent once to the Metadata Injection step.

* A standard approach is to add a sequence number – Get Value From Sequence step - to the rows and filter the first row only.

ETL Metadata Injection

Metadata injection inserts data from various sources into a transformation template at runtime.

* Open the ETL Metadata Injection step and specify as transformation template:

${Internal.Entry.Current.Directory}/tr\_template.ktr.

* Enter the following mappings:

|  |  |  |
| --- | --- | --- |
| Target | Source step | Source field |
| Text file input |  |  |
| FILE\_TYPE | File Details | file\_type |
| SEPARATOR | File Details | separator |
| ENCLOSURE | File Details | enclosure |
| HEADER\_PRESENT | File Details | header\_present |
| NR\_HEADER\_LINES | File Details | nr\_header\_lines |
| FILE\_FORMAT | File Details | file\_format |
| FILENAME\_LINES |  |  |
| FILENAME | File Details | input\_file\_directory |
| FILEMASK | File Details | input\_file\_name\_regex |
| FIELDS |  |  |
| FIELD\_NAME | Field Metadata | field\_name |
| FIELD\_FORMAT | Field Metadata | field\_format |
| FIELD\_NULL\_STRING | Field Metadata | field\_null\_string |
| FIELD\_TYPE | Field Metadata | field\_type |
| FIELD\_TRIM\_TYPE | Field Metadata | field\_trim\_type |

* Save the Transformation: tr\_injector.ktr