Talend User Component tNormalizeSchema

Purpose **http://www.cimt-ag.de**

This component normalize a wide schema into a smaller one. It has the same result like the build in tSplitRow component but provides these advantages:
* the field mapping is much easier and simple be done with a few settings
* byte code foot print is lower (helps to avoid violating the 64k method byte size in Java)
* the memory foot print is lower because the component does not keep the occurrence of fields in memory.

Talend-Integration

This component can be found in the palette under Processing->Fields
This component provides an output flow and several return values.

Parameters

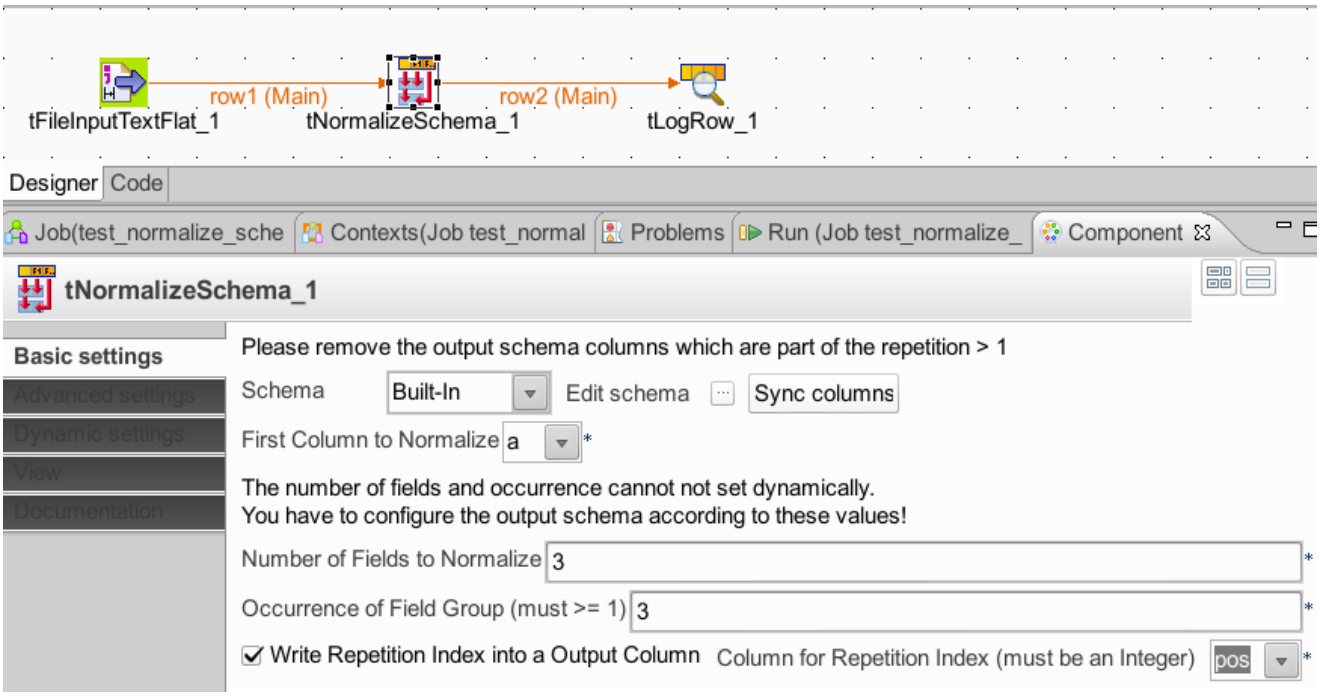
Property	Content
Schema	At first the input schema, but need configuration.
First Column to Normalize	The first (or the only) column which occurce more than one (and therefore should to be normalize).
Number of fields to normalize	In case there are more then one column to normalize (they have to follow after the first column) please specify here the number of columns.
Occurrence of field group	Specify how often the field or the group of fields occurse.
Write repetition into a output column	The number of occurrence can be saved into a output column.
Column for repetition index	Specify here the column (must be an Integer type) for the repetition index.

Return values

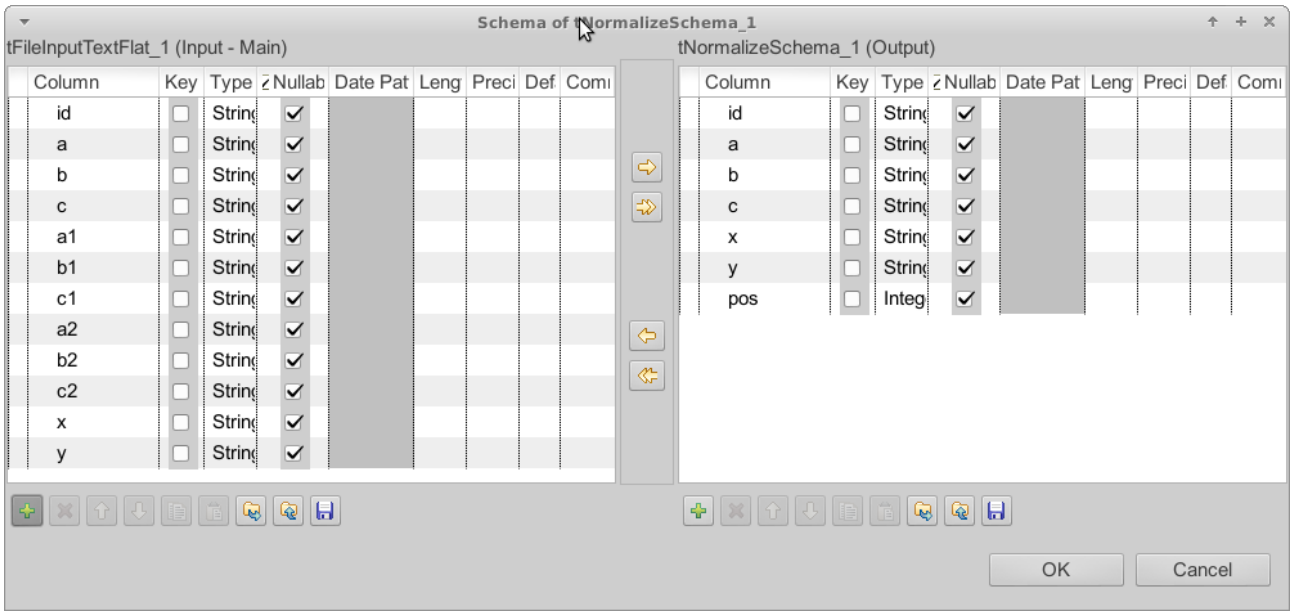
Return value	Content
ERROR_MESSAGE	Last error message
NB_LINE_IN	Number of input lines
NB_LINE_OUT	Number of output lines

Szenario

Normalize the given schema, read from a text file.



Here the schema:



Remove all columns which are used in the normalization.
It works with any kind of data types. This picture shows a very basic usage with only Strings used.
But there is no limitation in data types.

Here the result of a demo job:

Demo data input:

```
id;a;b;c;a1;b1;c1;a2;b2;c2;x;y
1id;1a;1b;1c;1a1;1b1;1c1;1a2;1b2;1c3;1x;1y
2id;2a;2b;2c;2a1;2b1;2c1;2a2;2b2;2c3;2x;2y
3id;3a;3b;3c;2a1;3b1;3c1;3a2;3b2;3c3;3x;3y
4id;4a;4b;4c;3a1;4b1;4c1;4a2;4b2;4c3;4x;4y
```

Demo data output:

```
.---+---+---+---+--+--+---.
|        tLogRow_1         |
|=--+---+---+---+--+--+--=|
|id |a  |b  |c  |x |y |pos|
|=--+---+---+---+--+--+--=|
|1id|1a |1b |1c |1x|1y|0  |
|1id|1a1|1b1|1c1|1x|1y|1  |
|1id|1a2|1b2|1c3|1x|1y|2  |
|2id|2a |2b |2c |2x|2y|0  |
|2id|2a1|2b1|2c1|2x|2y|1  |
|2id|2a2|2b2|2c3|2x|2y|2  |
|3id|3a |3b |3c |3x|3y|0  |
|3id|2a1|3b1|3c1|3x|3y|1  |
|3id|3a2|3b2|3c3|3x|3y|2  |
|4id|4a |4b |4c |4x|4y|0  |
|4id|3a1|4b1|4c1|4x|4y|1  |
|4id|4a2|4b2|4c3|4x|4y|2  |
'---+---+---+---+--+--+---'
```