



## tFileExcelSheetInput

### Purpose

This component reads Excel sheets.

This component needs the components tFileExcelWorkbookOpen (open a file or creates a workbook)

Advantages of this component:

- The columns to read can be set also with gaps
- The column position can automatically configured by a header line and additional by the use of regularly expressions
- Can read reliable all possible data types and tries also to convert it into the schema target column type
- Can read comments
- Can fill the output stream for empty cells with the last not empty value
- Can ignore cell read errors e.g. in case of a type conversion is impossible
- Uses always the latest Apache POI API

### Talend-Integration

This component can be found in the palette under File/Spreadsheet

This component provides several return values.

### Parameters for tFileExcelSheetOutput

Property	Content
Workbook	Choose the tFileExcelWorkbookOpen component holding the Apache POI Excel workbook
Schema	The schema of the output
Sheet name	The name or the index of the sheet. Please take care of a valid sheet name or simply type the index of the sheet your want to write. If the sheet does not exist, it will be created automatically. You can take the sheet name from the return value of the tFileExcelSheetList.
Row start index	The component starts reading data with this row (1-based)
Stop at empty row	If no values from the row was received (only all configured columns) the component stops reading.
Column start	Is visible only if no individual column configuration takes place
Use individual column configuration	If chosen it shows the configuration of columns and its position can be configured individual
User header to configure position of columns	If true the column position will be configured according to its position in the header line. The name will be found not case sensitive.
Header line	In the column configuration it is possible to configure the column position by the header line. Here set the index of the header line (1-based)
Column configuration	You can specify the columns in the Column Configuration in the column Sheet Column Name. Here you can use the Excel letter reference ("A" for the first column) or an index (0 for the first column). It is possible to have gaps between the different columns (unlike the build-in Talend components).  <b>Column:</b> Name of the schema column <b>Sheet Column Name:</b> Column position as letter (starts with "A" or 0) <b>Name in Header:</b> if position should be found in header, set here the name in the header (also with the use of regex) <b>Read cell comment:</b> If checked the comment will taken as value <b>Use last value for empty:</b> If checked an empty cell will be filled with the last known value <b>Ignore Errors:</b> If something goes wrong this option avoids aborts

## Advanced setting parameters for tFileExcelSheetOutput

Property	Content
Language / Country for number format	Number formats are different for different languages/countries. In case of the number is stored in a text typed cell and the schema expects a number this local will be used to find the correct format pattern for the text-to-number conversion.
Return Hyperlink URL	The normal value for a cell is always the visible cell value. If the cell has an underlying hyperlink, this option must be switched on to get the hyperlink instead of the value.
Concatenate Label URL	The label of the hyperlink is the visible cell value and if both (label and hyperlink) are needed the component read both values and concatenates them with a pipe symbol.
Trim columns	If checked all textual content will be trimmed (leading and trailing spaces, tabulators or line breaks will be removed)

## Return values of the component:

Value	Content
NB_LINE	Number of lines read
ERROR_MESSAGE	Error message if something went wrong
LAST_ROW_INDEX	Index of the last read row in this sheet.

## Scenario 1: Read from cells referenced by the Excel column names

The screenshot displays the configuration of the **tFileExcelSheetInput\_1** component. The **Basic settings** tab is selected, showing the **Workbook** set to **tFileExcelWorkbookOpen\_1**. The **Advanced settings** tab is also visible, showing the **Sheet name or index** set to **0**, **Row start index (starts with 1)** set to **2**, and **Limit rows to** set to an empty field. The **Column configuration** table is shown with the following columns: **Column**, **Sheet column name**, **Read cell comment**, **Use last value for empty**, and **Ignore Errors**. The table has four rows: **date** with sheet column name **"A"**, **value** with **"B"**, **id** with **"D"**, and **result** with **"E"**. The **Read cell comment** checkbox is checked for **id**. The **Ignore Errors** checkbox is checked for **value**.

The cell can be addressed with the well-known Excel column name (starting with "A") or the cell index (starting with 0).

## Scenario 2: Read from cells referenced by the header line

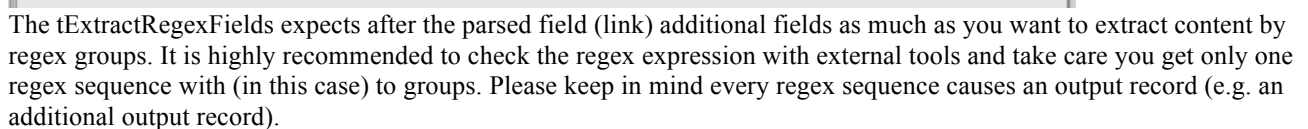
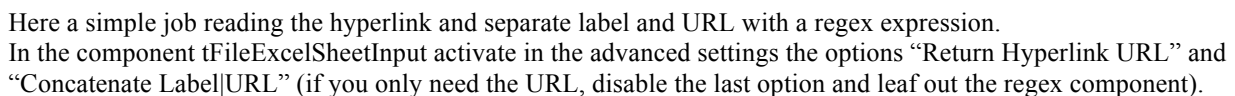
The screenshot displays the Talend Studio interface for configuring the **tFileExcelSheetInput\_1** component. The workflow diagram at the top shows a sequence: **tFileExcelWorkbookOpen\_1** → **OnSubjobOk** → **tFileExcelSheetInput\_1** (row1 (Main)) → **tLogRow\_1**.

The configuration panel for **tFileExcelSheetInput\_1** is open, showing the **Basic settings** tab. The **Header line** is set to **1**. The **Column configuration** table is visible, showing columns **date**, **value**, **id**, and **result** with their corresponding header names and configuration options.

Column	Name in Header	Ignore if missing	Read cell comment	Use last value for empty	Ignore Errors
date	"Create\date"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
value	"Cost per piece"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
id	"Customer\{A-Z}*"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
result	"Calculated Costs"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

In this scenario the column the header line will configure positions. The component tries to find the column by its name (case insensitive) or by regularly expressions (also case insensitive).

The file to read looks like this:



The output of the job:

Page 4