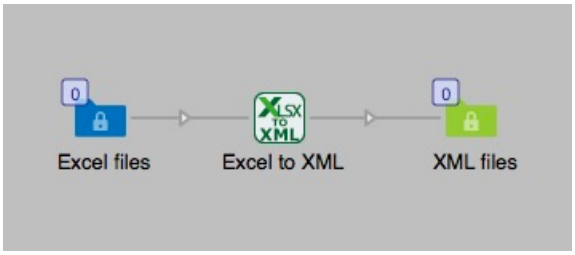
Excel to XML v11

Description
Excel to XML will let you submit an Excel file in the format .xlsx to a Switch flow where it will be converted to XML and/or metadata sets. It will accept Excel files with multiple sheets. You have different options of output either as XML-files or to pickup the Excel data and embed as a dataset in the same way as the XML-pickup element works.

Version 9 require Switch 2020 and up. It is based on Node.js and don't require any external scripts.

New in version 11 is automatic selection of head row if the Excel sheet have merged columns as a headline row. For "One XML-file for each row" you will now get leading zeros in the number suffix, this will make the sorting of the XML-files better.



To just output XML-files that later can be imported in to InDesign for automatic production of documents from the XML-data works without the Metadata module. This works in a simple situation where you just import the XML-file to InDesign as long as you don't need any information in the XML for controlling InDesign.

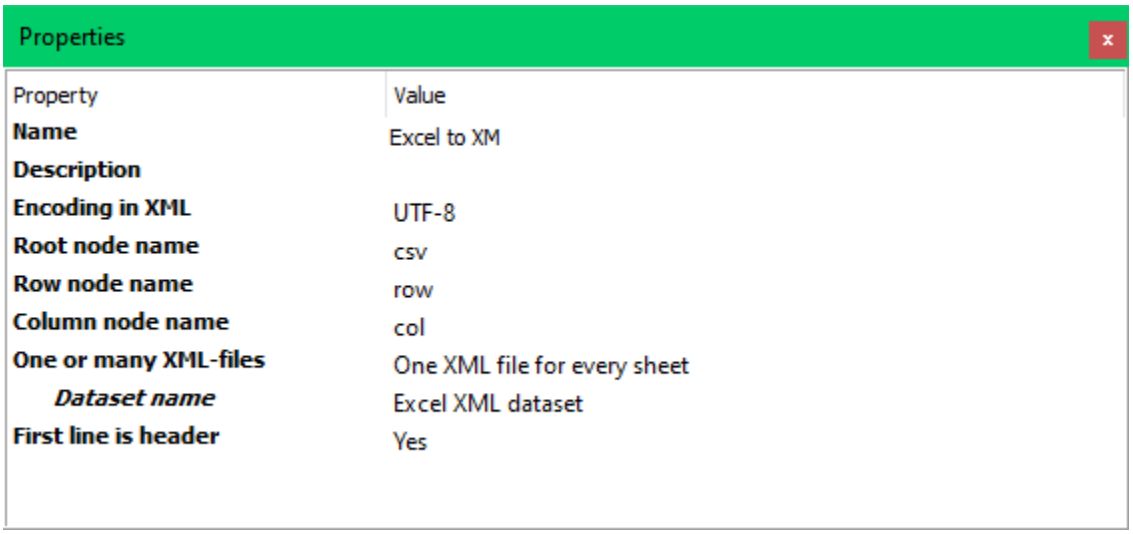
Compatibility
Switch 2020 Windows or Mac OSX.

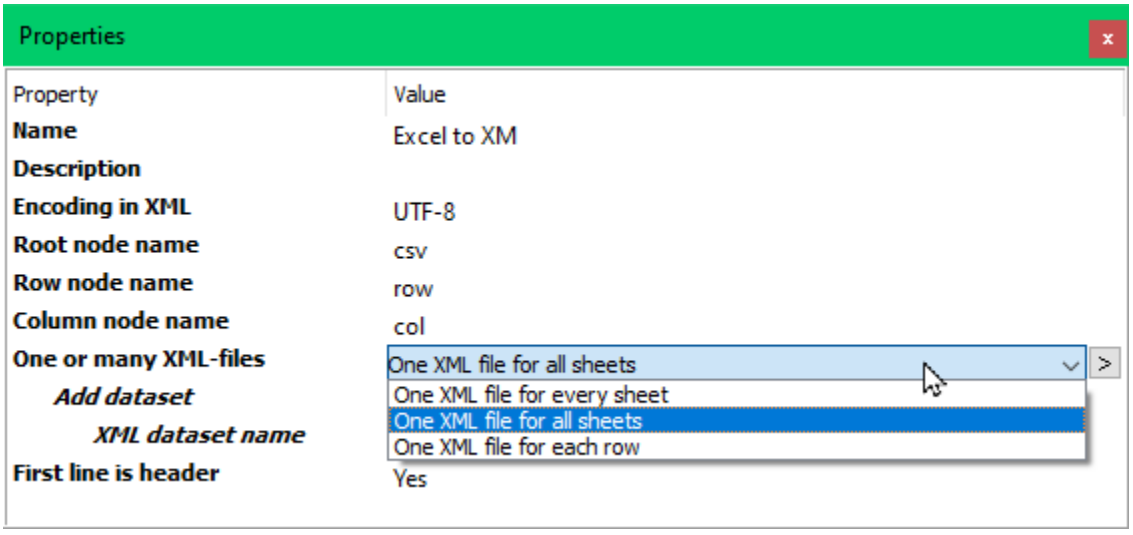
Connections
Excel to XML can have several input connections but there is only one outgoing connection. No settings are available of the outgoing connection.

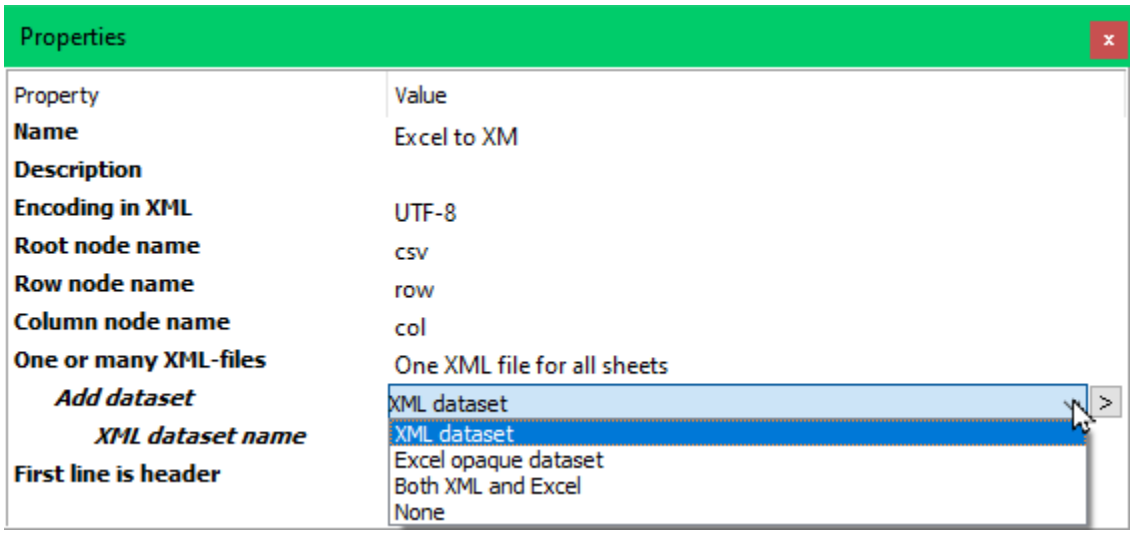
Flow element properties

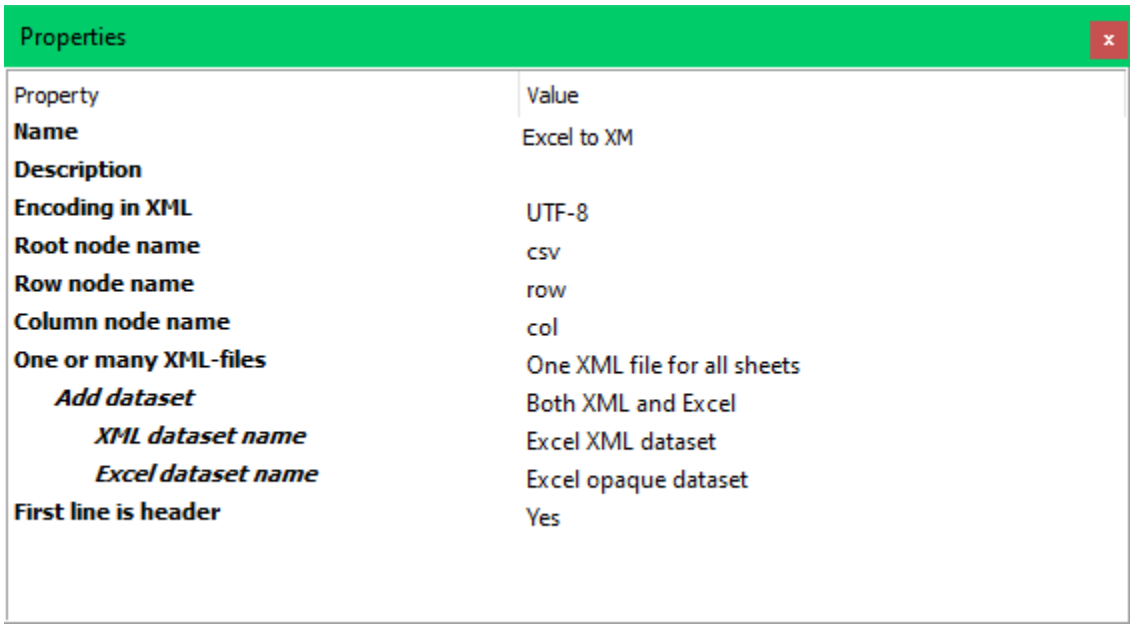
- Encoding in XML
  - Choose between UTF-8 or iso-8859-1. In some cases on Windows text with special characters will be encoded in iso-8859-1. If the encoding is set to UTF-8 in such cases the resulting XML will not parse correctly or not at all.
- Root node name
  - Give the root node a name of your choice, default is ***csv***.
- Row node name
  - Give the row node a name of your choice, default is ***row***.
- Column node name
  - Give the column node a name of your choice, default is ***col***.
- One or many XML-files
  - One XML file for every sheet.
    - Dataset name, set the name for the XML-dataset.
  - One XML file for all sheets.
    - Add dataset, dropdown menu
      - XML Dataset.
        - XML Dataset name, give a name to the XML Dataset.
      - Excel opaque dataset.
        - Excel dataset name, give the Opaque dataset a name. The Excel file will be saved as an Opaque dataset for later use.
      - Both XML and Excel
        - XML Dataset name, give a name to the XML Dataset.
        - Excel dataset name, give the Opaque dataset a name. The Excel file will be saved as an Opaque dataset for later use.
      - None, no dataset will be saved to the job.
  - One XML file for each row.
    - Dataset name, set the name for the XML-dataset.
    - Naming of files, Dropdown menu
      All file names will get a suffix of the row number to prevent over writing of duplicates, it will be separated with an underscore.
      - From values in sheet
        - First part of file name, column #
          You can name the output file with values from the column. The property should be the column number. Default is 1
        - Second part of file name, column #
          This property will let you add another columns value to the file name, the two values will be separated with an underscore. Value 0 means that there will be no second part. Default is 0.
      - From other sources
        - Name from sources. This will let you name the XML-files with a name or variable.
  - Leading zeros, will add leading zeros to the number suffix to the file names.

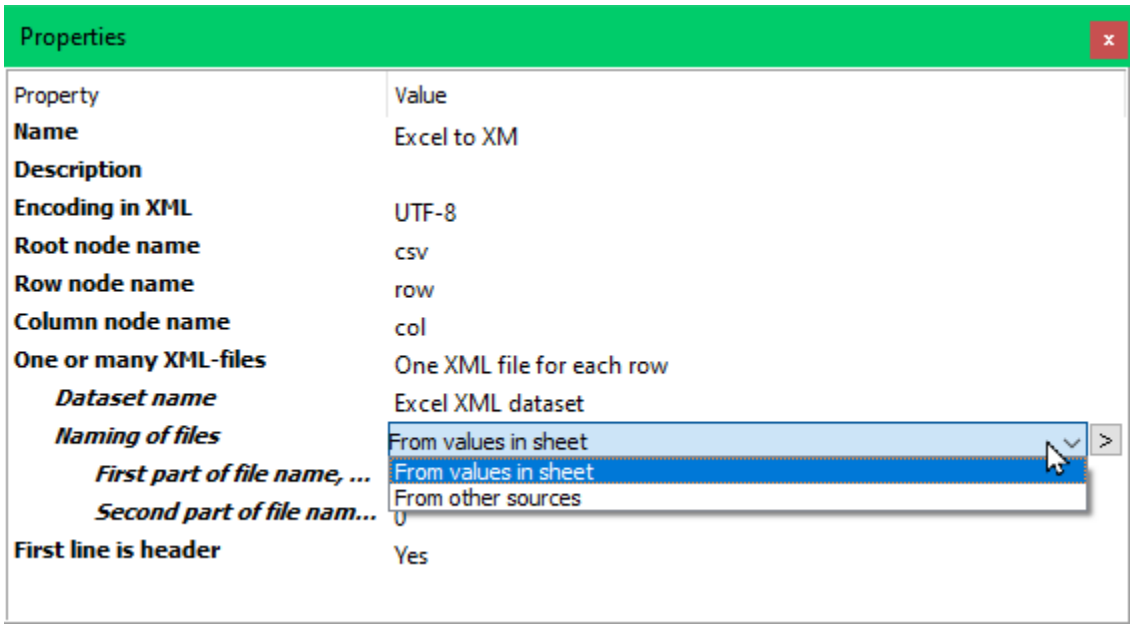
- First line is header
    - Yes, in this case the value of each column header will be used as a node tag in the XML, if the column header is “first_name” it will be like this: <col name=”first_name”>John</col>
    - No, then the first line is not a header and the values of the first line will be treated as all other rows in the Excel file.
        - None, The first row in the sheet will have values for the XML-file
        - Header row, and the select the row number in the next field. This is useful if the sheet has merged cells as a headline for the sheet. The XML will be generated as if you select Yes and the header rows values will be attributes.
    - Auto, will find the first row that is the header row and used as attributes. This can be useful if the sheet has merged cells as a headline for the sheet and you don’t know exactly which row is the headline.

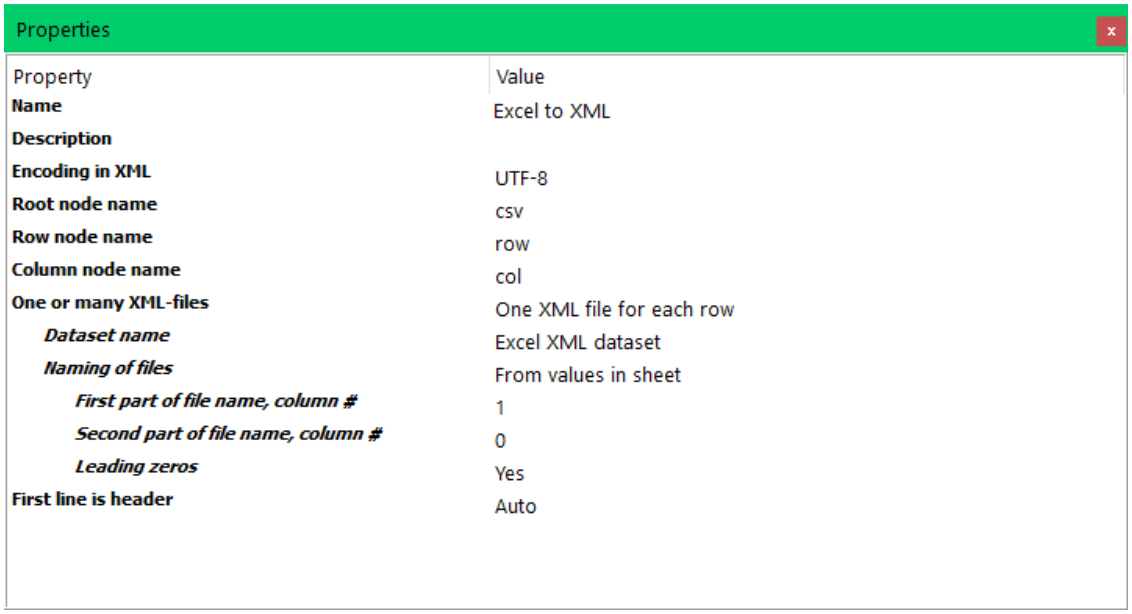





Properties
Property
Value
Name
Excel to XM
Description
Encoding in XML
UTF-8
Root node name
csv
Row node name
row
Column node name
col
One or many XML-files
One XML file for all sheets
Add dataset
XML dataset
XML dataset name
XML dataset
Excel opaque dataset
Both XML and Excel
None
First line is header


Properties
Property
Value
Name
Excel to XM
Description
Encoding in XML
UTF-8
Root node name
csv
Row node name
row
Column node name
col
One or many XML-files
One XML file for all sheets
Add dataset
Both XML and Excel
XML dataset name
Excel XML dataset
Excel dataset name
Excel opaque dataset
First line is header
Yes


Properties
Property
Value
Name
Excel to XM
Description
Encoding in XML
UTF-8
Root node name
csv
Row node name
row
Column node name
col
One or many XML-files
One XML file for each row
Dataset name
Excel XML dataset
Naming of files
From values in sheet
First part of file name, ...
From values in sheet
From other sources
Second part of file nam...
0
First line is header
Yes

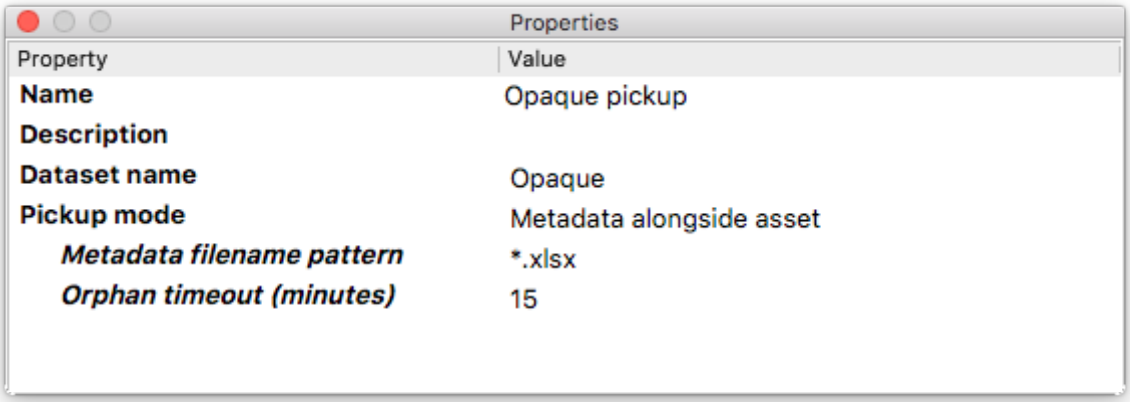

Properties
Property
Value
Name
Excel to XML
Description
Encoding in XML
UTF-8
Root node name
csv
Row node name
row
Column node name
col
One or many XML-files
One XML file for each row
Dataset name
Excel XML dataset
Naming of files
From values in sheet
First part of file name, column #
1
Second part of file name, column #
0
Leading zeros
Yes
First line is header
Auto

Properties

Property	Value
Name	Excel to XML
Description	
Encoding in XML	UTF-8
Root node name	csv
Row node name	row
Column node name	col
One or many XML-files	One XML file for each row
Dataset name	Excel XML dataset
Naming of files	From values in sheet
First part of file name, column #	1
Second part of file name, column #	0
Leading zeros	Yes
First line is header	No
Header row	Row number
Row number	2

Properties

Property	Value
Name	Excel to XML
Description	
Encoding in XML	UTF-8
Root node name	csv
Row node name	row
Column node name	col
One or many XML-files	One XML file for each row
Dataset name	Excel XML dataset
Naming of files	From values in sheet
First part of file name, column #	1
Second part of file name, column #	0
Leading zeros	Yes
First line is header	Yes



In the Opaque pickup element you have to set the properties as in the image above. Pickup mode must be “Metadata alongside asset”. And the “Metadata filename pattern” must be set to *.xlsx. In this property pane you can set the Dataset name for the Opaque pickup. It must be exactly the same here as you set in the Excel to XML property “Opaque dataset name”. If not the job will fail.

Extra information
If you use this app to output multiple XML files for later import in to InDesign for automatic production of documents you will need to adapt the XML to something useful for InDesign. To do that you have to use the Saxonica configurator and an XSLT-file. This will give you the possibility to produce business cards, tickets or product labels very quickly. You can do this without the Switch Metadata module.

You can also use the app Make XML to combine the dataset you get from Excel to XML and combine the XML from the Excel file with variables in Switch. In that way you can construct a new XML file with additional data for later use in Switch or sending to other systems.

To better process the resulting PDF-files that you have made with the XML-files from the app there is now two Private data that will help you.

- Private data key: ***SheetId***
  - This key will give you the name of the Excel sheet where the XML-data comes from.
- Private data key: ***NumberOfRecords***
  - This key will give you the number of records that are picked up from the Excel sheet. If you select one XML file for all sheets this Private data value will be the total number of rows for all sheets.
- Private data key: ***TotalNumberOfRecords***
  - This key will be useful with the option “One XML file for each row”. It will give you the total number of rows in the Excel file which is the same number of XML files generated. This value can be needed when you want to merge a number of PDF files that was made form the content in the XML file.

When you later in the flow will need to assemble the produced single PDF-files from each record these two Private data keys are needed.
In the Assemble job properties you will use the scheme “Custom” and then use the SheetId key as a job identifier and the property “Number of files” from the key “NumberOfRecords”

The XML structure for the multiple XML-files will have the X-path structured per each row and each column as follows: /csv/row/col but you can set your own name in the XML structure if you like.

Here is an example of an XSLT-file that can be used with the XML-files you get from the Excel to XML app. Each XML-file will have the name of the Excel workbook sheet.

XSLT example

Be aware that this might lead to overwriting XML-files where the default sheet name is used.

Note, with the new function in version 7 you don't need an XSLT to split XML-files to get one XML for each row.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
    <xsl:output method="xml" indent="yes"/>
    <xsl:template match="/">
        <xsl:apply-templates/>
    </xsl:template>
    <!--Splits the XML-file generated in to single XML-files, one for each row-->
    <xsl:template match="/csv">
        <xsl:for-each select="row">
        <!-- Selects which column value to use as filename for the resulting single XML-file. -->
        <!-- In this case column 2 is used. -->
            <xsl:variable name="filename"><xsl:value-of select="col[2]"/>.dita </xsl:variable>
            <xsl:result-document href="{col[2]}.xml" method="xml">
                <excel-row>
                <!--List the columns from the Excel-file in the order they will have in the resulting XML-file.-->
                <!--The name can be set to anything you like, for example the header used in Excel.-->
                    <col name="Column 1">
                        <xsl:value-of select="col[3]"/>
                    </col>
                    <col name="Column 2">
                        <xsl:value-of select="col[4]"/>
                    </col>
                    <col name="Column 3">
                        <xsl:value-of select="col[5]"/>
                    </col>
                    <col name="Column 4">
                        <xsl:value-of select="col[6]"/>
                    </col>
                    <col name="Column 5">
                        <xsl:value-of select="col[2]"/>
                    </col>
                </excel-row>
            </xsl:result-document>
        </xsl:for-each>
    </xsl:template>
</xsl:stylesheet>
```

If you choose to attach the Excel data to a job file as a dataset the XML structure will be as follows: /workbook/sheet/csv/row/col were each sheet node will have the name of the sheet.

New in version 2
Python script is no longer embedded due to licensing issues.

Better XML if first line is header.

New in version 3
App can now handle a one column Excel-file.

New in version 5
Two options to use Private data.

New in version 6
Option to set encoding in output XML.

New in version 7
Save one XML-file for each row in the Excel sheet, it is restricted to only one sheet in the Excel file. If there are any more sheets the job will fail.

Naming the nodes in the XML output.

New in version 8
You can now give XML files from each row a file name from other sources than the field values in the row. Name can be from variables or just a single line of text. All file names will get a suffix with the row number.

New in version 9
Version 9 only works in Switch 2020 and later and uses NodeJS for the app.
Some changes and additions in the properties.
Performance is a lot better for larger Excel files.

New in version 10
Fixed bug that prevented Excel files with only one column to work.
New Private data key added, TotalNumberOfRecords.

New in version 11
Adding leading zeros to the suffix when output one XML-file for each row.
Option to select rows that are headlines to be used as attributes when the headline is not the first row. It can be selected manually.
This selection can also be done automatically and the first line that looks like a headline will be used for XML attributes.

This app uses the SheetJS js-xlsx Node.js package.