

## Excel to XML

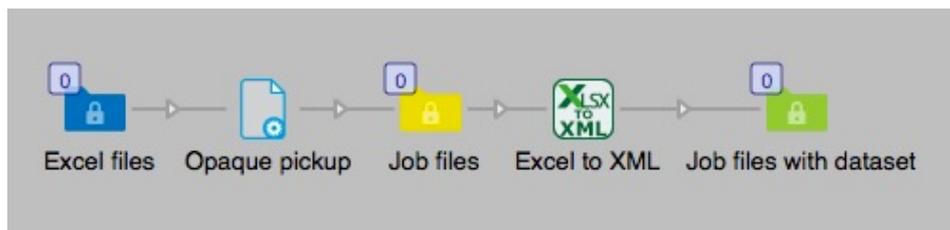
### Description

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 element XML pickup works.



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.

You can also pickup the data in the Excel document and embed it in a job file that you send in alongside with the Excel document. In this case you need the Metadata module and the element Opaque pickup in the flow before the Excel to XML app.



### Compatibility

Switch 13 update 1 and higher. Windows or Mac OSX.

#### Compatibility third-party applications

This app uses Python to process scripts, for Windows Python is included in the app and for Mac OSX it uses the Python that is in Mac OSX.

You don't need to install any other applications for this app. It is tested and runs with Python version 2.7x.

## Connections

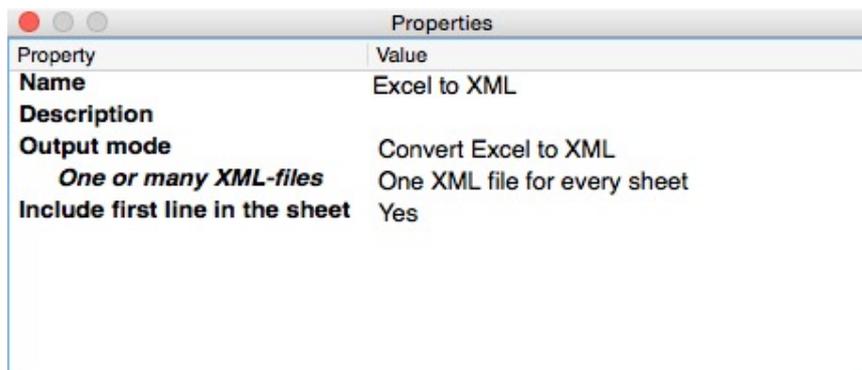
Excel to XML can have several input connections but there is only one outgoing connection. No settings are available of the outgoing connection.

## Properties detailed info

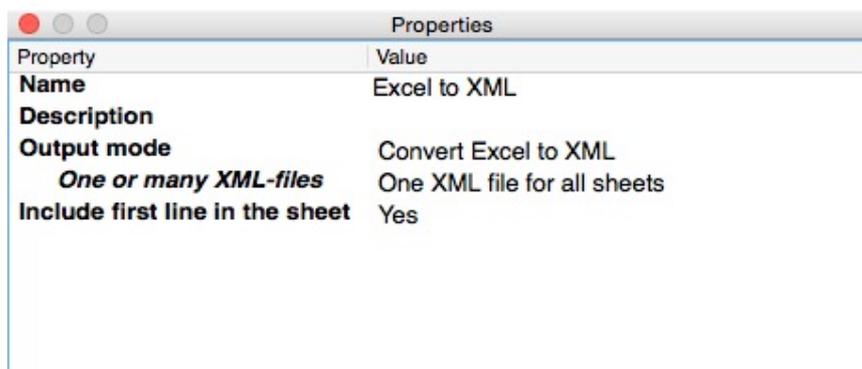
If you are using this app to just output XML files you just have to set the properties in the way you like. If you want to send Excel files along job files you need the element “Opaque pickup” that are included in the Metadata module. The reason for the “Opaque pickup” is to be able to pair a job file with the data in the Excel file.

## Flow element properties

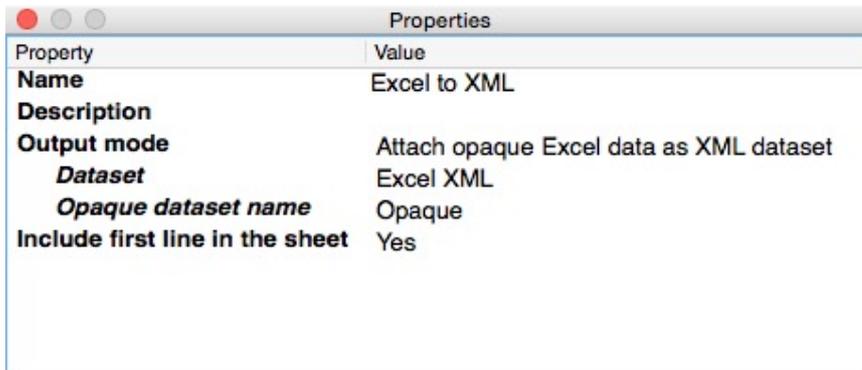
- Output mode
  - Convert Excel to XML.
  - Attach opaque Excel data as XML dataset.
- Convert Excel to XML
  - One XML file for every sheet.
  - One XML file for all sheets.
- Attach opaque Excel data as XML dataset
  - Dataset, the name of the dataset that you can use in variables in Switch.
  - Opaque dataset name, it is important that this property has the same name as the setting in the Opaque pickup elements “dataset name” property. If these two are not the same the job will fail.
- Include first line in the sheet
  - In some cases the Excel document has its first row as column header names, if so you can select if you want to include this row or not.



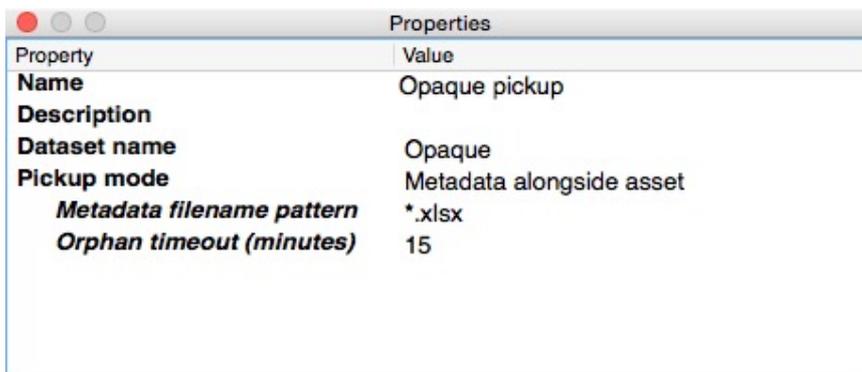
Property	Value
<b>Name</b>	Excel to XML
<b>Description</b>	
<b>Output mode</b>	Convert Excel to XML
<i>One or many XML-files</i>	One XML file for every sheet
<b>Include first line in the sheet</b>	Yes



Property	Value
<b>Name</b>	Excel to XML
<b>Description</b>	
<b>Output mode</b>	Convert Excel to XML
<i>One or many XML-files</i>	One XML file for all sheets
<b>Include first line in the sheet</b>	Yes



Property	Value
<b>Name</b>	Excel to XML
<b>Description</b>	
<b>Output mode</b>	Attach opaque Excel data as XML dataset
<i>Dataset</i>	Excel XML
<i>Opaque dataset name</i>	Opaque
<b>Include first line in the sheet</b>	Yes



Property	Value
<b>Name</b>	Opaque pickup
<b>Description</b>	
<b>Dataset name</b>	Opaque
<b>Pickup mode</b>	Metadata alongside asset
<i>Metadata filename pattern</i>	*.xlsx
<i>Orphan timeout (minutes)</i>	15

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.

The XML structure for the multiple XML files will have the Xpath structured per each row and each column as follows: /csv/row/col

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.

### Note

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

```

<?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.