

## TXT pickup

### Description

Metadata is often available in text format, e.g. as text in a mail. The disadvantage is that this information is not structured and not browsable as it is the case for XML-formatted metadata.

This app converts any text file to an XML dataset and it offers methods for detecting key/value pairs so the resulting XML is nicely structured and becomes more easily browsable.

### Compatibility

Switch 13 update 1 and higher.

### Connections

The app takes a text file or an opaque dataset as input and attaches a new XML dataset to the job. There is only a single output connection possible.

Here is a screenshot of a flow that receives the body text of a mail, converts it to XML and then uses the XML data to populate properties in the “HTTP request” element.



This is an example of what the body of the text could look like:

```
From: "Somebody" <somebody@acme.com>
Date: Tuesday 10 November 2015 09:48
To: "Enfocus support" <support@enfocus.com>
Subject: File upload from Acme: Brochure.pdf
```

Dear Recipient,

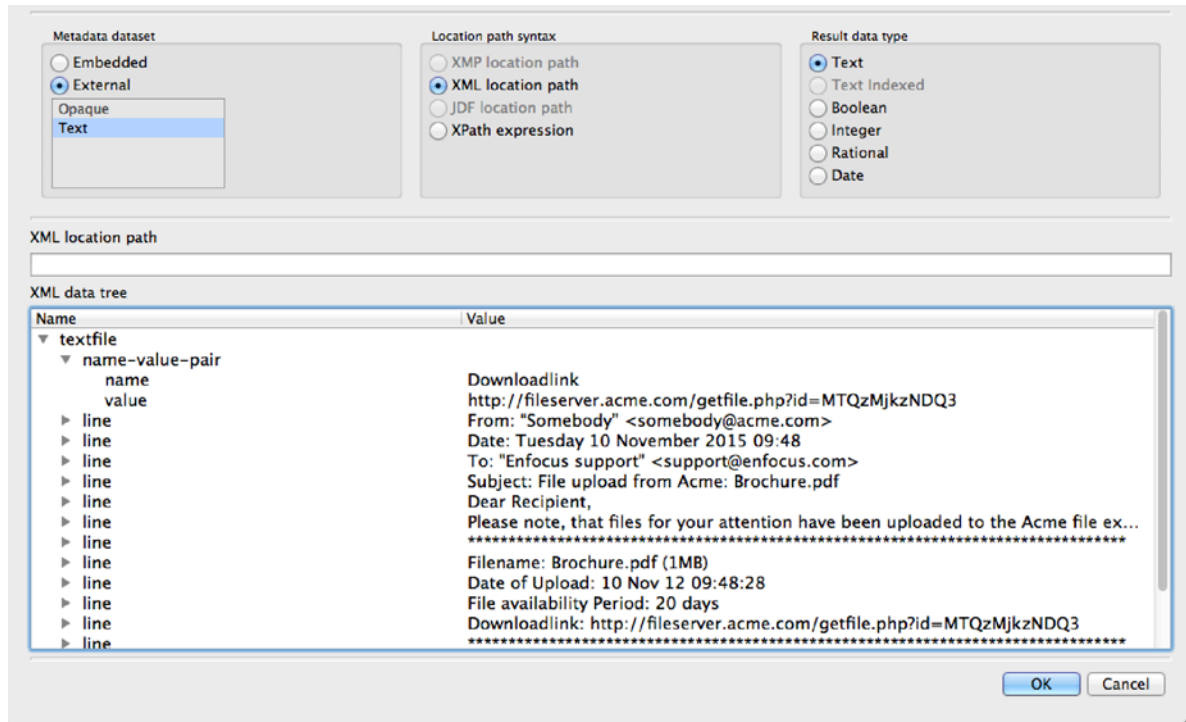
Please note, that files for your attention have been uploaded to the Acme file exchange server.

```
*****
Filename: Brochure.pdf (1MB)
Date of Upload: 10 Nov 12 09:48:28
File availability Period: 20 days
Downloadlink: http://fileserver.acme.com/getfile.php?id=MTQzMjkzNDQ3
*****
```

Kind regards,

Somebody at Acme

And here is a screenshot of the resulting XML in the metadata pane of the Switch variables pane.



As you can see the text info is now nicely formatted in two ways. At the end all the lines of the text file are available as individual `<line>` tags in the XML. Take for example the line with the download link: it is clear one may be interested in that piece of information, but on what line number is it? And the word "Downloadlink" has to be removed. That is why one or more of the lines can also be made available as key/value pairs. The word "Downloadlink" is a good filter for finding the line we are interested in and by breaking it up in a key/value pair based on a colon we have an easily searchable access to the required piece of metadata. You can see the result at the top of the XML in the `<name-value-pair>` tag.

## Properties detailed info

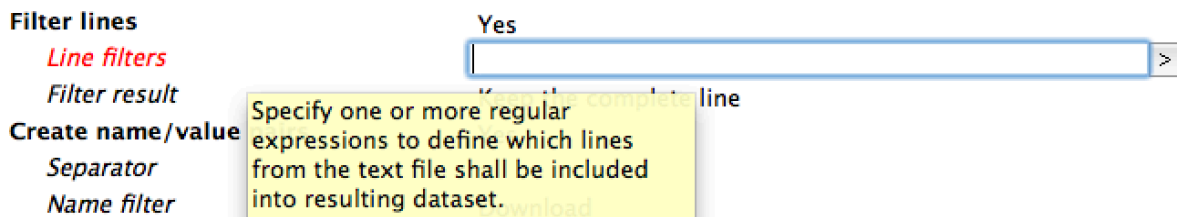
Property	Value
Name	TXT pickup
Description	
Pickup mode	Metadata is asset
Filter lines	No
Create name/value pairs	Yes
Separator	:
Name filter	Download
Name filter result	Keep the complete string
Value filter	None
Dataset name	Text

## Flow elements properties

- **Pickup mode**

This property is a drop-down list offering two choices: “Metadata is opaque dataset” and “Metadata is asset”. The latter is used in the example shown above to convert the mail body text that is injected into the flow by the “Mail receive” element.

- **Filter lines**



When the “Filter lines” property is set to “Yes” you can define one or more **regular expressions** (!) in the “Line filters” property to define which lines from the text file shall be included in the resulting dataset. This can be useful when the text file is very big: you could perhaps get rid of most lines making the resulting dataset smaller and easier to analyze. Or when the filtering of lines allows you to pick up that one line that contains the metadata you are interested in; that makes the location path to the required information very easy.

The “Filter result” property has two options: “Keep the complete line” and “Keep the matched string”. Suppose you are using a filter that matches mail addresses. It is conceivable that you only want the mail address itself in the dataset in which case you choose the option “Keep the matched string”. The other option will keep the complete line. Filtering out lines that do **not** contain a certain string can easily be done by using negative regular expressions.

- **Create name/value pairs**

---

<b>Create name/value pairs</b>	Yes
<i>Separator</i>	:
<i>Name filter</i>	Download
<i>Name filter result</i>	Keep the complete string
<i>Value filter</i>	Multi-line text with variables defined
<i>Value filter result</i>	Keep the matched string

When this property is set to “Yes” you can specify how the name/value pairs can be detected in the text file.

The “Separator” property will usually take a single character like “=” or “:” (as in the example above), but it also accepts a regular expression. This is interesting when the separator is a special character, like a tab (\t), or when you want to use a list of possible separators (=|:\t).

The “Name filter” and “Value filter” work in the same way as the line filter described above.

- **Dataset name**

This is the name of the new dataset that is attached to the job.