

XPATH PARSE

Description

The XPath Parse will parse XML nodes from existing XML & create a new XML document from the parsed XML.

App Usage

The app can process repeating nodes into individual XML documents, which means that it is very useful for processing XML's that are baskets of orders from an eCommerce site.

In addition to processing repeating nodes of XML data, single item XML nodes can be prefixed & postfixed to the repeating nodes of data.

This means that additional information, like delivery address information can added to an ordered item's information in the XML document being output.

Usage with XPath Search

There is a companion called app XPath Search which can be used with XPath Parse to be able to build an advanced flow that uses both the apps to achieve an ultimate goal.

One example would be where one would need to know the possible number of final output XML, where the final target is a repeating XML node inside another repeating XML node.

The final targeted repeating node can be counted initially, with the count placed on the XML as private data. The XML can then be split out until the final result is achieved with the number of XML's being a known quantity, by accessing the initial private data of the count.

What follows is configuration information for the app. The tables will explain more in depth the configuration, and screenshots will illustrate actual configurations that have been used. These need to be adjusted to the individual use case.

Example flows are provided.

Overview of configurator options

Specify output format	xml
Set custom root name	root
Base key for private data	xpathKey2
Dataset name of outgoing file	split_artwork_nodes
XML encoding	utf8
Nodes to prefix	Yes
<i>Specify list of prefix nodes</i>	PARENT_NODE /root/SubJobCard CHILD_NODES JobCardNumber
Repeating node	Yes
<i>Path of parent xml node to query</i>	/root/SubJobCard/ClientArtwork/ClientArtworkAsset
<i>Repeating child node/s</i>	ArtworkFileName
Nodes to postfix	Yes
<i>Specify list of postfix nodes</i>	PARENT_NODE /root/SubJobCard CHILD_NODES BrandingCode SpecialInstructions BrandingDepart...

Overview table of configurator

Element	Options	Element Type	Description
Name		String	Name of app element in Switch.
Description		String	Description of element in Switch.
Specify Output format		Dropdown list	Specify the output format of the XML parsed. Dataset will be set according to output format.
	xml		
	json		
Set custom root name		String	This will set the root name of the XML, or root object name for JSON.
Dataset name of outgoing file		String	Set the name of the dataset of the file that is output by the app
XML encoding		Drop Down List	
	utf8		Default encoding.
	latin 1		Set to this if character encoding of XML is not utf8.
Nodes to prefix		Yes/No List	Set to Yes to specify the nodes to get. A very specific format is required for this. More detail on this later in the documentation.
Repeating node		Yes/No List	Set to Yes to specify the nodes to get. A very specific format is required for this. More detail on this later in the documentation.
Nodes to postfix		Yes/No List	Set to Yes to specify the nodes to get. A very specific format is required for this. More detail on this later in the documentation.

Table above gives an overview of the options available in the configurator. In depth configuration is discussed further on the next pages in the document.

Configuration for parsing XML in various ways

Nodes to prefix	Yes
<i>Specify list of prefix nodes</i>	PARENT_NODE /root/SubJobCard CHILD_NODES JobCardNumber
Repeating node	Yes
<i>Path of parent xml node to query</i>	/root/SubJobCard/ClientArtwork/ClientArtworkAsset
<i>Repeating child node/s</i>	ArtworkFileName
Nodes to postfix	Yes
<i>Specify list of postfix nodes</i>	PARENT_NODE /root/SubJobCard CHILD_NODES BrandingCode SpecialInstructions BrandingDepartment...

Configuration for parsing nodes

Element	Option	Element Type	Description
Nodes to prefix		Yes/No List	Set to Yes to use.
	Specify list of prefix nodes	Multi-line text	First line must start with PARENT_NODE followed by a single pipe symbol. The route to the parent must start with a /, & each node name preceded with a /. No / at the end of the string. Next line must start with CHILD_NODES followed by a single pipe symbol. After the pipe symbol the names of the nodes to be prefixed are to be specified. If there is more than one node to be prefixed these node names should be preceded by a single pipe symbol. No pipe symbol at the end of the string of nodes.
Repeating node		Yes/No List	Set to Yes to use.
	Path of parent xml node to query	Single line text	Route to the parent of the node to query. Must start with a /, & each node name preceded with a /. No / at the end of the string.
	Repeating child node/s	Multi-line text	Specify the names of the repeating nodes to get. One node name per line.
Nodes to postfix		Yes/No List	Set to Yes to use.
	Specify list of postfix nodes	Multi-line text	The setup for the postfix is the same as for the prefix nodes. The formats need to be followed strictly else the parsing of the XML data will fail.

Note on configuration options

One does not have to follow the model of prefix nodes, repeating nodes & postfix nodes

If all that is required is the repeating nodes, then these can be parsed by themselves.

Likewise if only single nodes are to be parsed then these can be parsed as prefix nodes with the other 2 options set to No..

Likewise the Repeating node option can be set to Yes & be used with one of the other options set to Yes.

NOTES ON PRIVATE DATA

The private data placed on the job will look like this:

For successful processing:

[private data name].baseXMLName

[private data name].nodeCount - the number of nodes matched

For errors:

[private data name].statusOfJob.status- value will be: fail

[private data name].statusOfJob.message - error message giving some indication of why the error occurred