Parsing Complex XML to all levels and converting to a Tabular format

Have you run onto challenges when it comes to parsing XML files and trying to convert data to a 'meaningful' tabular format? I had my share of fun and challenge. Especially if the XML is nested to multi-level. Analysis on data from XML files is one of the issues I have faced recently. The file (nodes and data) was next 13 levels deep with ~350 data fields to capture; and goal was to convert it to a tabular data and share on a SQL server to allow our partners to be able to use this data for different use cases including Quality Validation.

There are several macros available for parsing XML data in Alteryx community. I used Parse XML which is a powerful macro which can parse an XML file for all levels. This macro if used properly is awesome and reduces a lot of stress to deal with XML files.

I am going to show you an example where I will parse the "<body>" of the XML which generally has the most complex data. You can view another example of this on Alteryx Community by clicking here.

The example I am using is simple XML on 'Donuts' 😊

Every XML is different, but this macro will parse all nodes and data. After parsing, different techniques can be used to convert 'parsed' to a 'meaningful' data in tabular format.

I have provided some tips below on how to start parsing complex XMLs. A workflow is also attached to convert "Donuts.xml" to a tabular data format. If the data is required to be loaded to a Database server, I would recommend that you NORMALIZE data before loading. In the donuts example, you can have different tables for "Donut Master", "Filling Master", "Topping Master" and "Batter Master" and then 1 transaction file for "Donut Transactions".

Download the Parse XML macro here:

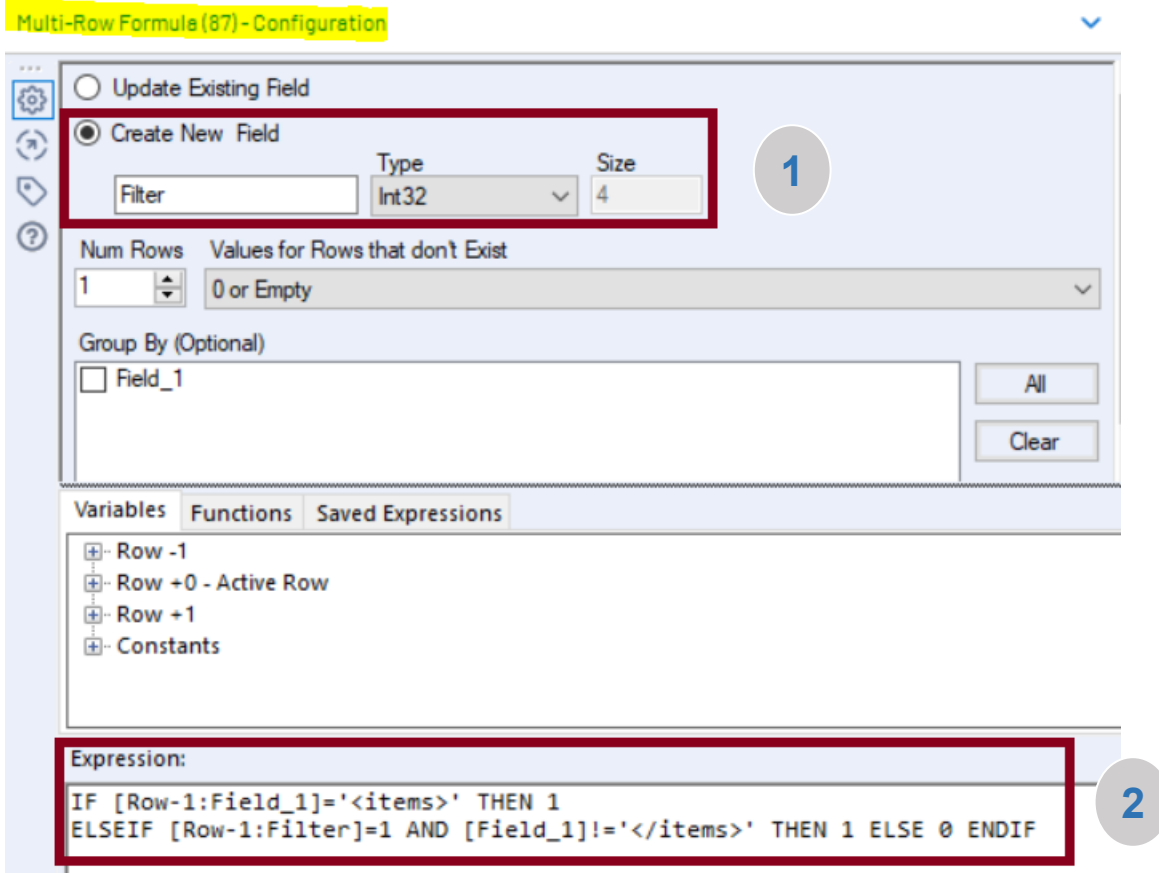https://gallery.alteryx.com/#!app/Parse-XML/584b5589f499c704689e363b

Credits: Hildebrando Souza (https://community.alteryx.com/t5/user/viewprofilepage/user-id/51984)

Chris Smallwood (https://community.alteryx.com/t5/user/viewprofilepage/user-id/68765)

Tip 1 for Parse XML macro: Load XML file as a CSV file with separator as EOL (\n)

Input Data (53) - Configuration

Connect a File or Database
donuts.xml   **1**

Options

	Name	Value
1	Record Limit	
2	File Format	Comma-Delimited Text Files (*.csv) **2**
3	Search SubDirs	☐
4	Output File Name as Field	No
5	Delimiters	\n **3**
6	First Row Contains Field Names	☐
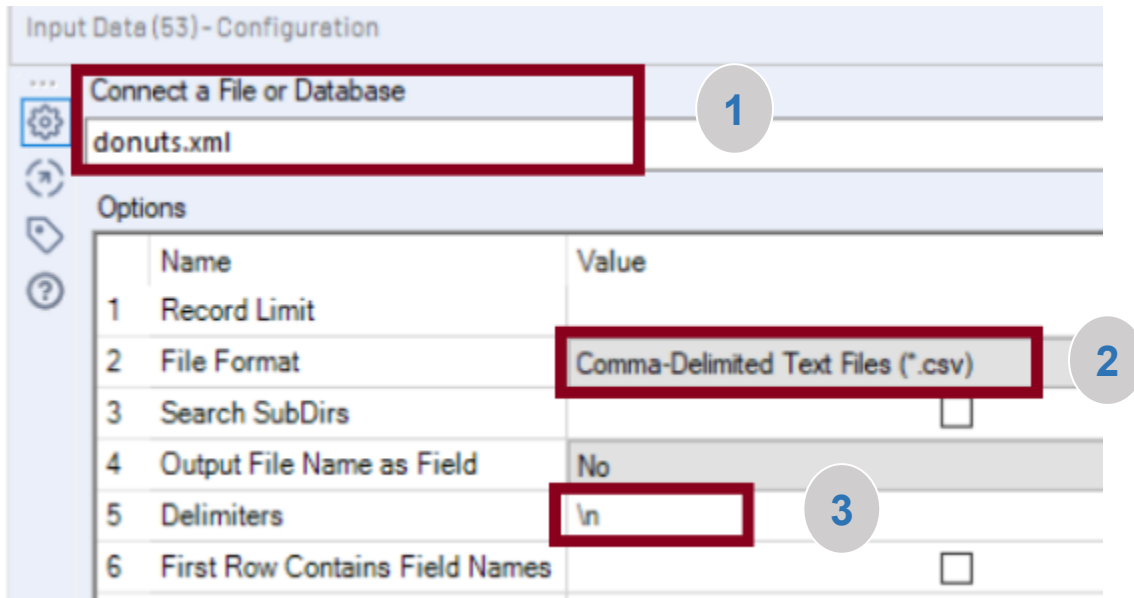
Tip 2 for Parse XML macro: Identify "<Body>" from XML, Node that has data

Multi-Row Formula (87) - Configuration   ⌄

○ Update Existing Field
◉ Create New Field
　　　　　　　　　　　　Type　　　　　　Size
　Filter　　　　　　　　Int32　　⌄　　4　　　　　**1**

Num Rows   Values for Rows that don't Exist
1   ⬍   0 or Empty   ⌄

Group By (Optional)
☐ Field_1   All
                                                             Clear

Variables   Functions   Saved Expressions
⊞ Row -1
⊞ Row +0 - Active Row
⊞ Row +1
⊞ Constants

Expression:
```
IF [Row-1:Field_1]='<items>' THEN 1
ELSEIF [Row-1:Filter]=1 AND [Field_1]!='</items>' THEN 1 ELSE 0 ENDIF
```
2

Tip 3 for Parse XML macro: Filter data based on the "new field" created in Tip 2. The summarize data using the "field" from XML (step 1).

Read the XML file as CSV using a line break as a separator.

Create a value of 1 for all lines related to <body> section

[Filter] = 1

Concatenate the information back to a single cell
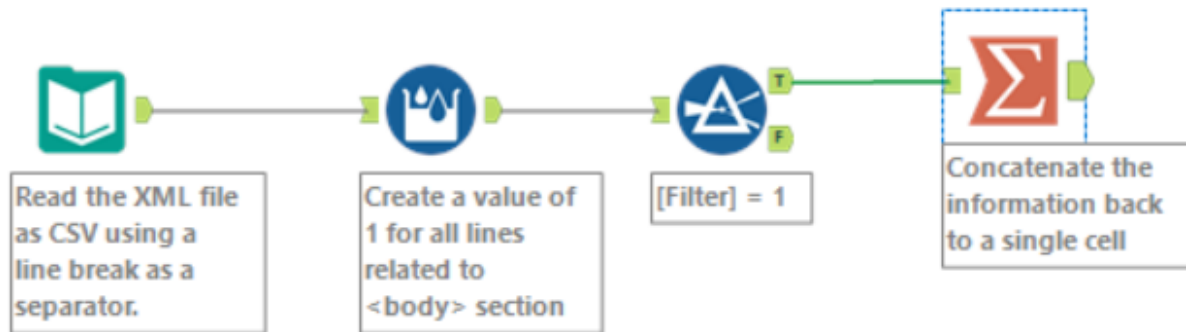
Tip 4 for Parse XML macro: Pass the summarized data from Tip 3 to the 'Parse XML' macro.

Concatenate the information back to a single cell

Parse XML macro: Pass the XML info
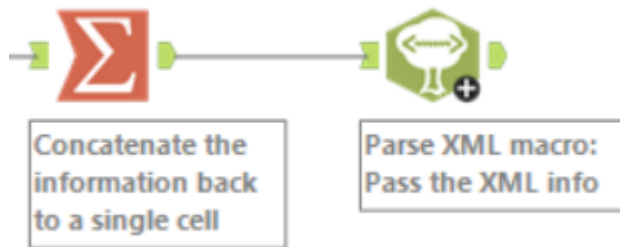
I would love to hear more from you on your challenges and solutions. And reply back to this post by sharing your use cases or ideas on XML parsing 😊