XML Formatter For Grouping Records

> **Version Available**
> This feature is available from version 2.9

At times, we have to group records to construct a single message to send to an application adapter. This is often the case when processing records that have multiple sub-records attached to it.

One area where we see this the most is while processing Invoices. Some applications - like QuickBooks, MS Dynamics (GP) or Intacct - need Invoice requests as a single request. However, a user might have this as a set of records in a single database table or view.

Database queries, or CSV File readers, often have records that are grouped together as flat responses. The responses are not easy to convert or map, into an XML structure.

For example:-

invoice_num	date	PO	id	product_code	prod_qty	prod_amt	total
1	1/1/2011	123	1	ABC	10	100	200
1	1/1/2011	123	1	DEF	20	200	200
1	1/1/2011	123	1	GHI	30	300	200
2	2/1/2011	123	2	ABC	10	100	300
2	2/1/2011	123	2	DEF	10	100	300

or in a CSV file as:

```
invoice_num,date,PO,idproduct_code,prod_qty,prod_amt,total
1,1/1/2011,123,1,ABC,10,100,200
1,1/1/2011,123,1,DEF,20,200,200
1,1/1/2011,123,1,GHI,30,300,200
2,2/1/2011,123,2,ABC,10,100,300
2,2/1/2011,123,2,DEF,10,100,300
```

In the above example, we have a set of records that represents two invoices. The first (invoice_num=1) has 3 line items. And, the second (invoice_num=2) has 2 line items. When a request is sent to the target application, it expects to see 2 records request instead of 5 as described above.

To get the right grouping, make use of our **XML-Formatter** property capability. XMLFormatter property takes the response from the Reader and, groups it to form the right XML structure. To do so, the Reader will have a property "xmlformatter" and will be set as follows:

```
invoice_num,date,po,id,[product_code,prod_qty,prod_amt],total
```

Any value within "[" and "]" will be treated like a sub-item. The column before "[" will be the driving column - i.e. a change in that item will create a new XML. In this example "id" becomes the **grouping column**.

The split creates a node on the element just before "[" with "list". So, in the above example, the XML will look like the following:

```
<row>
 <invoice_num>value</invoice_num>
 <date>value</date>
 <po>value</po>
 <id id="">
 <list>
 <product_code>value</product_code>
 <prod_qty>value</prod_qty>
 <prod_amt>value</prod_amt>
 </list>
 <list>
 <product_code>value</product_code>
 <prod_qty>value</prod_qty>
 <prod_amt>value</prod_amt>
 </list>
 </id>
 <total>value</total>
 </row>
```

Now, Map the target loop structure to "id/list" along with the field maps.

Limitations:

- Only one level of grouping is allowed - i.e. there can only be one "[" and one "]"
- The grouping column (in the above case, "id") should not be mapped to any field - other than to the **loop** field in the mapping.