

Splitting an XML Document Using the LiveCycle ES2 XSLT Service

The attached LiveCycle ES2 process demonstrates a technique for splitting a structured XML document into multiple XML documents. The process splits an XML document of customers, depicted below on the left, by customer. The XML document below on the right is the first of three XML documents produced by the split.

```
- <customers>
- <customer id="1">
  <name>Adobe Systems Incorporated</name>
  <address>345 Park Ave.</address>
  <city>San Jose</city>
  <state>CA</state>
  <zip>95110</zip>
</customer>
- <customer id="2">
  <name>Apple</name>
  <address>1 Infinite Loop</address>
  <city>Cupertino</city>
  <state>CA</state>
  <zip>95014</zip>
</customer>
- <customer id="3">
  <name>Google Inc.</name>
  <address>1600 Amphitheatre Parkway</address>
  <city>Mountain View</city>
  <state>CA</state>
  <zip>94043</zip>
</customer>
</customers>
```

Source XML Document

```
- <customers>
- <customer id="1">
  <name>Adobe Systems Incorporated</name>
  <address>345 Park Ave.</address>
  <city>San Jose</city>
  <state>CA</state>
  <zip>95110</zip>
</customer>
</customers>
```

Split XML Document 1

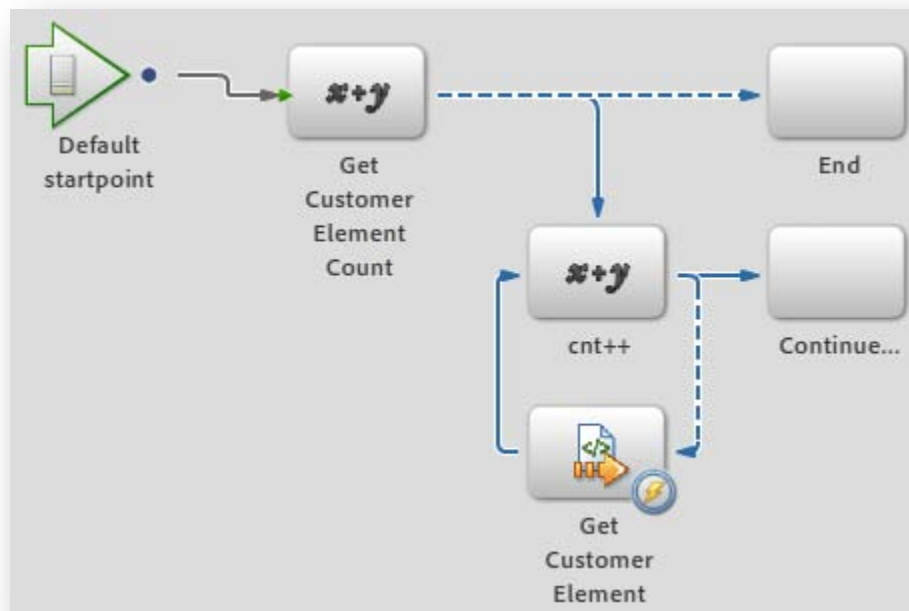
The XSL used in the process is depicted below. The XSL is embedded in the Transform operation as a template.

```
- <xsl:stylesheet version="1.0">
  <xsl:output method="xml"/>
  - <xsl:template match="/"> ← match the root and call the next template
    <xsl:apply-templates/>
  </xsl:template>
  - <xsl:template match="customers"> ← match a 'customers' element and call the next template
    - <customers>
      <xsl:apply-templates/>
    </customers>
  </xsl:template>
  - <xsl:template match="customer"> ← match a 'customer' element and
    - <xsl:if test="@id='{$/process_data/@cnt$}'"> ← test the occurrence of 'customer' by 'id' using the
      - <customer id="@id"> ← add the 'id' attribute from the source document
        <xsl:apply-templates/>
      </customer>
    </xsl:if>
  </xsl:template>
  - <xsl:template match="name | address | city | state | zip"> ← match all the children elements and copy the text
    <xsl:copy-of select="."/>
  </xsl:template>
</xsl:stylesheet>
```

- match the root and call the next template
- match a 'customers' element and call the next template
- match a 'customer' element and
 - test the occurrence of 'customer' by 'id' using the process variable '{@/process_data/@cnt\$}', where 'cnt' is the loop controller,
 - add the 'id' attribute from the source document to the output and
 - apply the next template
- match all the children elements and copy the text nodes to the 'customer'

Splitting an XML Document Using the LiveCycle ES2 XSLT Service

The process map is depicted below.



The sole purpose of the process is to split the source XML document. The process has the following characteristics:

1. The process has a single input variable of type 'xml'. There is no output variable defined.
2. The Set Value instance 'Get Customer Element Count' sets the process variable 'cnt' of type 'int' to zero, uses the Node function 'count' to get the number of occurrences of 'customer' and assigns the result to the process variable 'customerCnt' of type 'int'.
3. The route between 'Get Customer Element Count' and the abstract activity 'End' is a test for zero instances of 'customer' (a 'customerCnt' of zero).
4. The process then iterates by 'cnt' over the source XML document checking for a 'cnt <= customerCnt'.
5. The XSLT service Transform instance 'Get Customer Element' does an XSLT match by the 'customer' attribute 'id' where 'id' equals 'cnt'. If the source XML document did not include an attribute 'id' the match could be done by the relative position of 'customer' in 'customers' also. The output from 'Get Customer Element' is assigned to a process variable of type 'xml'. This process does not act upon each XML instance.
6. Once 'cnt' equals 'customerCnt' the Set Value instance 'cnt++' routes to the abstract activity 'Continue...'. The rest is up to you.

The process archive (Split_XML_1-0_2010-07-29-1720.lca), sample customer XML document (customers.xml) and a copy of the XSLT (customers_split.xsl) are attached to the PDF.

If you have any questions, please contact stwalker.adobe@gmail.com.