MultiFragmentTraxTransformer

All XSLT transformers cause problems when transforming large documents (hundreds of thousands of XML elements). They tend to run out of memory when processing these documents, and increasing the heap size of the JVM has its limits (approximately 3.5 GB without 64-bit extensions). When a lot of memory is used, the operating system will swap a lot of pages, which makes your Cocoon application not just big, but slow as well.

At first sight, this may seem strange, especially if you think of a pipeline as a stream of SAX events. Unfortunately, an XSLT transformer must keep a representation of the whole document in memory. This is the only way it can compute XPath expressions such as

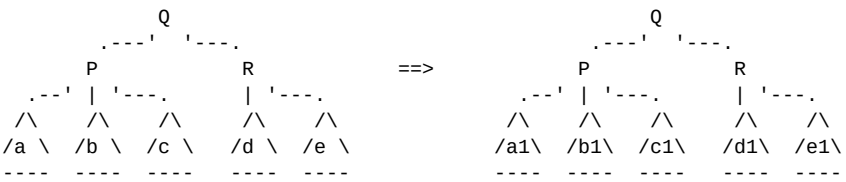
```
<xsl:copy-of select="//somewhere[@location='far away']"/>
```

In effect, this turns your efficient SAX-event processing pipeline into a kind of DOM processing monster, albeit a rather efficient one.

In some cases, this problem can be solved without sacrificing the convenience of using XSLT. These cases are similar to the 'map' operation on lists, where a function is applied to each element of the list, keeping the list structure:

```
f( [a. b. c. ...] )  =  [f(a), f(b), f(c), ...]
```

The pattern we are looking for occurs when the same transformation is applied to a number of sub-documents of the same form, and the super-structure remains the same:



The sub-documents in this picture are a, b, c, d, e.

If we could apply an XSLT stylesheet to each sub-document, and keep the super-structure, the XSLT engine only needs to store the subdocuments in turn, resulting in big memory savings.

The MultiFragmentTraxTransformer implements the map-like pattern, by applying the same stylesheet separately to all sub-documents, or fragments. The fragments are characterized by their root-element, which is a parameter of the MultiFragmentTraxTransformer. It is also possible to specify a namespace for the fragment-root.

The MultiFragmentTraxTransformer is declared similarly to the normal TraxTransformer:

```
    <map:transformer logger="sitemap.transformer.xslt" name="multifragment-xslt"
        pool-grow="2" pool-max="32" pool-min="8"
        src="org.apache.cocoon.transformation.MultiFragmentTraxTransformer">
      <use-request-parameters>false</use-request-parameters>
      <use-session-parameters>false</use-session-parameters>
      <use-cookie-parameters>false</use-cookie-parameters>
      <xslt-processor-role>xalan</xslt-processor-role>
      <check-includes>true</check-includes>
    </map:transformer>
```

The 'xslt-processor-role' may be xalan or saxon; both have been verified to work.

Usage in a pipeline is simple:

```
    <map:transform type="multifragment-xslt" src="multifragment.xsl">
      <map:parameter name="fragment.namespace" value="http://namespace.uri/fragment"/>
      <map:parameter name="fragment.element" value="fragment"/>
      <!-- parameters for the stylesheet -->
    </map:transform>
```

The fragment.namespace is optional; by default the fragment is in no namespace.

The code for the MultiFragmentTraxTransformer is attached to this page, and is free to use. You can get the attachment by clicking on "More Actions" below, and choosing "Attachments". If you have any questions or improvements, please send them to the mailing list.