

BetwixtTransformer

I just compared some XML-Binding frameworks like JAXB, Castor, XMLBeans, JiBX etc. The one I most liked was Betwixt. It is very easy to use, produces SAX events, it works with EJBs RemoteInterface, easy to use mapping files, clean XML, supports Maps and Collection in a way I like it. And it is more Beans-centric, good if you want to use your existing beans (Castor is more Schema-centric, I think).

But there is no [BetwixtTransformer](#) with cocoon, so I wrote one that is very analoque to the [CastorTransformer](#). It works also with EJBs.

Some links:

- [How does Betwixt compare to technologies like JAXB and Castor?](#)
- [Entity Beans with Betwixt](#)

How it works

from the javadoc

Betwixt transformer marshals an object from the Sitemap, Session, Request or the Context into a series of SAX events.

Configuration: The betwixt transformer can be configured to not output element reference ids. The default setting is to output reference IDs.

```
<map:transformer name="betwixt" src="org.apache.cocoon.transformation.BetwixtTransformer">
  <ref-ids>true</ref-ids>
</map:transformer>
```

[BR](#)

Sample:

```
<root xmlns:betwixt="http://apache.org/cocoon/betwixt/1.0">
  <betwixt:include name="invoice"/>
  <betwixt:include name="product" scope="sitemap"/>
  <betwixt:include name="product2" element="other-product"/>
</root>
```

The `BetwixtTransformer` support only one Element `betwixt:include`. This element is replaced with the marshalled object. The Object given through the attribute `name` will be searched in the `request`, `session`, `context` and at least in `sitemap`. If the `scope` is explicitly given, the object will be located only there. The attribute `element` can be given to specify an alternative root element for the object. Collections are marshalled by marshalling each object it contains.

How to use

make sure you have the following files in your classpath (or newer version) [Commons: BIN](#) | [SOURCE](#):

- commons-beanutils-1.5.jar
- commons-betwixt-1.0-beta-1.jar
- commons-collections-2.1.jar
- commons-digester-1.3.jar
- commons-logging-1.0.3.jar
- xerces.jar and xml-apis.jar

put the following in your sitemap:

```

<!-- under map:transformers -->
<map:transformer name="betwixt" src="org.apache.cocoon.transformation.BetwixtTransformer"/>

<!-- In your pipeline -->
<map:match pattern="test-betwixt.xml">
  <map:generate type="serverpages" src="test-betwixt.xsp"/>

  <map:transform type="betwixt"/>
  <!--<map:transform type="xslt" src="sometemplate.xsl"/>-->
  <map:serialize type="xml"/>
</map:match>

```

and a file named test-betwixt.xsp:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">
  <page>
    <xsp:logic>
      <!-- Add something to Request, that should be written as XML -->
      java.util.Map testBean = new java.util.Hashtable();
      testBean.put("entry1", "String1");
      testBean.put("entry2", new Integer(22));
      request.setAttribute("testB", testBean);
    </xsp:logic>
    <example xmlns:betwixt="http://apache.org/cocoon/betwixt/1.0">
      <betwixt:include name="testB" scope="request"/>
    </example>
  </page>
</xsp:page>

```

The output should somehow like:

```

<page>
  <example>
    <testB>
      <entry><key>entry1</key><value>String1</value></entry>
      <entry><key>entry2</key><value>22</value></entry>
    </testB>
  </example>
</page>

```

Attachment: [BetwixtTransformer.java](#)