

Using XSL to generate HTML

Author: Mihai Rotaru

Date: 1 Dec 2011

The Extensible Stylesheet Language Family (XSL) can be used to transform XML documents into other types of XML documents (such as XHTML), or into completely different formats - such as PDF, or simple plain text.

One of the most common applications of XSL is to present documents stored inside XML files as XHTML, for consumption by browsers. The principle behind XSL is simple: take a well-formed XML document, find the needed elements/nodes/text (using XPath), and place them in a template. Assuming a valid XSL template and a well-formed XML file were used, the resulting document will also be a valid XHTML document.

Taking the document used in the previous blog post, which contains information about a few movies, I will assume the goal is to generate the XHTML code which will present the information inside the file.

To accomplish this, an XSL Style Sheet must be created. The name is slightly misleading, because XSL can be used not only to configure and customize how an XML document is presented, but transformations can be applied as well. This will not, of course, alter the XML document, but might not represent all the data inside it - some information can be filtered out, new information might be added by the XSL template.

I have come up with the following XSL file:

```
01 <?xml version="1.0"?>
02 <xsl:stylesheet
03 version="1.0"
04 xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
05 <xsl:template match="/">
06 <!--XHTML generation begins here-->
07 <html>
08 <body>
09 <h2>A list of movies</h2>
10 <p>This list comprises several randomly selected movies</p>
11 <table>
12
13 <!--Generate table headings-->
14 <tr>
15 <th>Title</th>
16 <th>Year</th>
17 <th>Rating</th>
18 </tr>
19
20 <!--Generate table rows-->
21 <xsl:for-each select="movies/movie">
22 <tr>
23 <td><xsl:value-of select="title"/></td>
24 <td><xsl:value-of select="year"/></td>
```

```
25 <td><xsl:value-of select="rating"/></td>
26 </tr>
27 </xsl:for-each>
28 </table>
29 </body>
30 </html>
31 <!--XHTML generation finishes here-->
32 </xsl:template>
33 </xsl:stylesheet>
```

A few things to note:

I intentionally omitted the price information, to underline how XSL might create the wrong impression of the information contained in the XML file. This information is available inside the XML, but the XSLT omits it.

XPath is used on L05, to define the elements to which the template should be applied. In this case, the '/' means that it should be applied to the whole document. Again, very similar to paths in Unix - '/' represents the root of the filesystem.

XPath is used again on L21; the statements contained by the for-each element will be executed for each node matching the XPath expression "movies/movie".

Another important XSL element, the `xsl:value-of` is used on lines 23-25; it relies on the concept of the "current element". In the context of a `for-each`, this means that each element matching `for-each`'s XPath (in this case, "movies/movie") will become the "current element", in sequence. We have five movie elements in our XML file, so each of these will become, in turn, the current element. `xsl:value-of` will extract the value of certain child elements - title, year and rating, and place these values instead of itself.

One more step is needed - we need to tell the XML file that we made an XSL stylesheet for it - otherwise, when opened in a browser, it will simply be displayed as before. To do this, a line needs to be added to the XML file:

```
2 <?xml-stylesheet type="text/xsl" href="movies-style.xml"?>
```

When opened in an XSL-compliant browser (all modern browsers are compatible with XSL), the XML file will be processed using our XSL file (movies-style.xml), and the result of this process is XHTML - which will then be displayed by the browser.