

## Creating transformation using text editor

In this practice we will create an xslt file for transforming an xml into a new required format. You can find many products that natively support editing xslt files as this is relatively common task. Visual Studio 2008 has very good support for XSLT (though you need to activate advanced features from the registry), NetBeans only offers basic support (as it does for XML files generally). Many lightweight text editors (like Notepad++) also offer limited support for editing XSLT files.

Information on XSLT and XPath (used for addressing in XSLT files) can be found at <http://www.w3.org/Style/XSL/> . There are examples of XSLT available at w3schools and many other Internet locations.

### Warm up

Original file: [plants.xml](#)

Stylesheet: [table.css](#)

- **From XML to HTML**

1. Create a java project in Netbeans.
2. Add a new XSL Stylesheet.
3. Create a parser to transform the XML original file into a table in HTML:
4. To test your xslt, right click in your xslt file and select “XSL Transformation”, select the file to transform and specify the name of the output file.
  - a. Look at the examples in:  
<http://infohost.nmt.edu/tcc/help/pubs/xslt/index.html>

From:				
<pre>&lt;CATALOG name="Title"&gt;   &lt;PLANT&gt;     &lt;COMMON&gt;Bloodroot&lt;/COMMON&gt;     &lt;BOTANICAL&gt;Sanguinaria canadensis&lt;/BOTANICAL&gt;     &lt;ZONE&gt;4&lt;/ZONE&gt;     &lt;LIGHT&gt;Mostly Shady&lt;/LIGHT&gt;     &lt;PRICE&gt;\$2.44&lt;/PRICE&gt;     &lt;AVAILABILITY&gt;031599&lt;/AVAILABILITY&gt;   &lt;/PLANT&gt;   . . . &lt;/CATALOG&gt;</pre>				
To:				
Title				
<b>Name</b>	<b>Botanical</b>	<b>Zone</b>	<b>Light</b>	<b>Price</b>
Bloodroot	Sanguinaria canadensis	4	Mostly Shady	\$2.44
. . .				

Add a stylesheet to change the properties of the table. To add the css in the HTML, add the next line into the head section:

```
<link rel="stylesheet" type="text/css" href="fileName.css" title="Style"/>
```

- **From XML to XML**

It is also possible to transform an XML document into a new XML. Create a XSLT to obtain the following file:

```
To:
<PLANTS name="Title">
  <ITEM name="Bloodroot">
    <PRICE>$2.44</PRICE>
    <BOTANICAL>Sanguinaria
canadensis</BOTANICAL>
  </ITEM>
  . . .
</PLANTS>
```

## Purchase Order Mediation

This exercise introduces you to the development of simple transformations of XML documents. The tutorial exercise is based on a SWS Challenge scenario : Purchase Order Mediation ([http://sws-challenge.org/wiki/index.php/Scenario:\\_Purchase\\_Order\\_Mediation](http://sws-challenge.org/wiki/index.php/Scenario:_Purchase_Order_Mediation)). The transformations to implement are:

- From RosettaNet Purchase Order (PO) to Moon Add Line Item operation message (with parameter showing LineItem index).
- From Moon Order Confirmation to RosettaNet Purchase Order Confirmation (POC). Example documents and Schemas are available from the scenario web site. You are welcome to complete the whole challenge scenario as well.

## Debugging

- **Netbeans 6.5 - 6.9:** <http://plugins.netbeans.org/plugin/17747/xslt-debugger>. An easy way to start is following the next tutorial: <http://netbeans-xslt-debugger.java.net/>
- **Visual Studio:** <http://www.codeproject.com/Articles/41992/Debugging-XML-Transforms-XSLT-with>
- **Eclipse XSLT:** <http://x-assist.sourceforge.net/>

## Graphical XSLT Mapper

XSLT mapper lets you visually work with any XSLT file, mapping data from one or more input files to any output format. This is a nice thing to have and you would really use in an Enterprise scale project. Some examples that you can try are:

[http://www.stylusstudio.com/xslt\\_mapper.html](http://www.stylusstudio.com/xslt_mapper.html)  
<http://www.axizon.com/>  
<http://xsl-easy.com/eclipse/>  
<http://soa.netbeans.org/> // Netbeans 6.5 - 6.7