

Processing XML with XSLT

Groovy's [XML capabilities](#) are powerful enough that you may not need to consider other alternatives such as [XSLT](#). If you wish to use XSLT though, it isn't hard to do. Here we just added xalan and serializer jars from the Apache [Xalan-J](#) project to our CLASSPATH. Depending on your JVM version and other available libraries, you may need slightly different jars (or none at all). Once our path is set up, we can run the following script to convert an XML file into XHTML:

```

// require(url='http://xml.apache.org/xalan-j/', jar='serializer.jar')
// require(url='http://xml.apache.org/xalan-j/', jar='xalan_270.jar')
import javax.xml.transform.TransformerFactory
import javax.xml.transform.stream.StreamResult
import javax.xml.transform.stream.StreamSource

def input = '''
<?xml version="1.0" ?>
<persons>
  <person username="JS1">
    <name>John</name>
    <family_name>Smith</family_name>
  </person>
  <person username="ND1">
    <name>Nancy</name>
    <family_name>Davolio</family_name>
  </person>
</persons>
'''.trim()

def xslt = '''
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/persons">
    <html xmlns="http://www.w3.org/1999/xhtml">
      <head>
        <title>Testing XML Example</title>
      </head>
      <body>
        <h1>Persons</h1>
        <ul>
          <xsl:apply-templates select="person">
            <xsl:sort select="family_name" />
          </xsl:apply-templates>
        </ul>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="person">
    <li>
      <xsl:value-of select="family_name"/>,
      <xsl:value-of select="name"/>
    </li>
  </xsl:template>
</xsl:stylesheet>
'''.trim()

def factory = TransformerFactory.newInstance()
def transformer = factory.newTransformer(new StreamSource(new StringReader(xslt)))
transformer.transform(new StreamSource(new StringReader(input)), new
StreamResult(System.out))

```

Here is the output (pretty-printed):

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Testing XML Example</title></head>
<body><h1>Persons</h1>
<ul>
  <li>Davolio, Nancy</li>
  <li>Smith, John</li>
</ul>
</body>
</html>
```

Depending on your environment (JVM, XSLT processor, operating system) you might be able to make use of XSLT directly from the command line instead of calling it from Java or Groovy. You can also call it from Ant using the `xslt` task.