



# Simple DOM Programming with Java

## Introduction

This exercise will introduce you to the W3C's core Document Object Model (DOM) as implemented in the Java-based Oracle XML Parser.

## The Exercise

The exercise shows you how to use the DOM API to manipulate a document's object tree. In this case, you will be manipulating the contents of an element and also the value of an attribute associated with an element.

There are a number of tasks required in this exercise. Each task is detailed below.

## Installing Software

Your instructor will provide you with the *Oracle XML Parser* software and give you instructions regarding how these should be installed in your system.

This should be installed before you do this exercise.

You should also ensure that a Java Development Kit is installed on your system before undertaking this exercise.

## Setting Up

Make a new directory for this exercise. Call this directory *DOM*:

```
C:\> mkdir DOM
C:\> cd DOM
```

All the files that you subsequently create as you do this exercise should be contained in this directory.

## Developing the Java Source

Create the following main class in the file *ConvertCurrency.java* (N.B.: case is important in Java. You should ensure that everything you do follows the examples and commands shown in this script.):

```
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.Text;

import oracle.xml.parser.v2.DOMParser;
import oracle.xml.parser.v2.XMLDocument;

import java.io.PrintWriter;
import java.io.FileReader;

public class ConvertCurrency
{
    private static final PrintWriter
        out = new PrintWriter (System.out, true),
        err = new PrintWriter (System.err, true);

    public static void main (String args [])
    {
        try
        {
            final String
                currency = args [1];
            final double
                conversionRate = new Double (args [2]).doubleValue ();
            final DOMParser parser = new DOMParser();
            parser.setErrorStream (err);
            parser.setValidationMode (false);
            parser.showWarnings (true);
```



C

```
parser.parse (new FileReader (args [0]));
final XMLDocument
    doc = ((DOMParser) parser).getDocument();
final NodeList prices = doc.getElementsByTagName ("price");
for (int i = 0; i < prices.getLength (); i ++){
    {
        final Node
            node = prices.item (i),
            firstChild = node.getFirstChild ();
        final double
            oldValue =
                new Double (firstChild.getNodeValue ().doubleValue ());
        node.getAttributes ().getNamedItem ("unit").
            setNodeValue (currency);
        final Text
            t = doc.createTextNode (" " + oldValue * conversionRate);
        node.replaceChild (t, firstChild);
    }
    doc.print (out);
}
catch (final ArrayIndexOutOfBoundsException aiobe)
{
    err.println ("usage: ConvertCurrency xmlfile currency rate");
    System.exit (1);
}
catch (Exception e)
{
    e.printStackTrace();
    System.exit (2);
}
System.exit (0);
}
```

### Creating the Simple Data Source

The above application is written to manipulate a very simple data file. Enter the following into a file called *values.xml*:

```
<?xml version="1.0"?>
<root>
  <thing>
    <price unit="USD">100.99</price>
  </thing>
  <thing>
    <price unit="USD">12.25</price>
  </thing>
  <thing>
    <price unit="USD">33.45</price>
  </thing>
</root>
```

### Establishing the Java CLASSPATH

Ensure that your Java CLASSPATH is set appropriately (the CLASSPATH tells Java where to find components of an application. *The example given here assumes that Oracle's parser is installed into C:\xmlparser\_v2\_0\_2\_7. Check your local system for the location of the Oracle parser and use the 'real' value in the command below*):

```
set CLASSPATH=.;C:\xmlparser_v2_0_2_7\lib\xmlparserv2.jar;%CLASSPATH%
```

### Compiling the Application

Use the following command:

```
javac ConvertCurrency.java
```

This will create the java class file *ConvertCurrency.class*.



## Executing the Application

Enter the following command:

```
| java ConvertCurrency values.xml MOP 10.00
```

C

You will be able to see how the simple application that you have written has transformed the input XML.

Transentia Pty. Ltd. DonationWare