



# Using the SAX API

B

## Introduction

This exercise will introduce you to the Simple API for XML (SAX) as implemented in the Java-based Oracle XML Parser.

## The Exercise

The exercise shows you how to use the SAX API to transform a simple XML document to an HTML table. The transformation requires no contextual knowledge to be maintained during processing and so this is a very appropriate task for the stream-based SAX.

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 SAX:

```
C:\> mkdir SAX
C:\> cd SAX
```

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

## Developing the Java Source

A typical SAX application such as this consists of a number of classes: the main class (from which the application begins executing) and one or more handler classes.

Create the following main class in the file *Converter.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 java.io.PrintWriter;

import org.xml.sax.Parser;
import org.xml.sax.DocumentHandler;
import org.xml.sax.helpers.ParserFactory;

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

    public static void main (final String args [])
        throws Exception
    {
        final Parser
            parser = ParserFactory.makeParser ("oracle.xml.parser.v2.SAXParser");
        final DocumentHandler
            handler = new MusicianHandler (out);
        parser.setDocumentHandler (handler);
        parser.parse (args [0]);
    }
}
```



The actual work of processing the XML document is performed by the MusicianHandler class instantiated by the Converter 'driver' class you typed in above. Create the file *MusicianHandler.java* with contents as follows:

B

```
import org.xml.sax.HandlerBase;
import org.xml.sax.AttributeList;

import java.io.PrintWriter;
import java.io.IOException;

public class MusicianHandler extends HandlerBase
{
    private PrintWriter fout;

    public MusicianHandler (final PrintWriter fout)
        throws IOException
    {
        this.fout = fout;
    }

    public void startDocument ()
    {
        fout.println("<HTML>");
        fout.println("<HEAD><TITLE>Members of Yes Circa 1973</TITLE></HEAD>");
        fout.println("<BODY>");
    }

    public void endDocument ()
    {
        fout.println("</BODY></HTML>");
        fout.close ();
    }

    public void startElement (String name, AttributeList atts)
    {
        if (name.equals ("musicians"))
        {
            fout.println ("<TABLE BORDER='1' CELLPADDING='5'>");
            fout.println ("<TR ALIGN='left'>");
            fout.println ("<TH>");
            fout.println ("Name");
            fout.println ("</TH>");
            fout.println ("<TH>");
            fout.println ("Instrument");
            fout.println ("</TH>");
            fout.println ("<TH>");
            fout.println ("Number of Recordings");
            fout.println ("</TH>");
            fout.println ("</TR>");
        }
        else if (name.equals ("musician"))
            fout.println ("<TR>");
        else
            fout.println ("<TD>");
    }

    public void endElement (String name)
    {
        if (name.equals ("musicians"))
            fout.println ("</TABLE>");
        else if (name.equals ("musician"))
            fout.println ("</TR>");
        else
            fout.println ("</TD>");
    }

    public void characters (char ch [], int start, int length)
    {
        for (int i = start; i < start + length; i++)
            fout.print (ch [i]);
    }
}
```

### Creating the XML Data Source

Create the following in a file called *Musicians.xml*:



B

```
<musicians>
  <musician>
    <name>Bill Bruford</name>
    <instrument>Drums</instrument>
    <NrOfRecordings>35</NrOfRecordings>
  </musician>
  <musician>
    <name>Steve Howe</name>
    <instrument>Guitar</instrument>
    <NrOfRecordings>24</NrOfRecordings>
  </musician>
  <musician>
    <name>Jon Anderson</name>
    <instrument>Vocals</instrument>
    <NrOfRecordings>14</NrOfRecordings>
  </musician>
  <musician>
    <name>Chris Squire</name>
    <instrument>Bass Guitar</instrument>
    <NrOfRecordings>7</NrOfRecordings>
  </musician>
  <musician>
    <name>Rick Wakeman</name>
    <instrument>Keyboards</instrument>
    <NrOfRecordings>22</NrOfRecordings>
  </musician>
</musicians>
```

Note: my long-time favourite group is a band called “Yes.” This data source contains information regarding a few of the members of that band—feel free to substitute your own favourites here...

### 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 *.java
```

This will create two java class files: *Converter.class* and *MusicHandler.class*.

### Executing the Application

Enter the following command:

```
java Converter file:./Musicians.xml > yes.html
```

You should now be able to double-click on the newly created file *yes.html* to see the resulting HTML file in IE.

### Making the Generated HTML “Beautiful”

The generated output is very basic. You can try changing the characteristics of the generated HTML table: BGCOLOR, BORDER, WIDTH, CAPTION, etc.