Constructing a Class List

8

Introduction

This exercise will introduce you to a few XML-savvy tools:

- XML Notepad
- Oracle's Java-based XML parser
- Internet Explorer 5
- The Microsoft *Validate.htm* file

You will also learn how to create a small XML document, how to add a DTD to the document and how to add an external Schema to the document.

The Exercise

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

Installing Software

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

These should all be installed before doing this exercise.

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

Setting Up

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

```
C:\> mkdir ClassList
C:\> cd ClassList
```

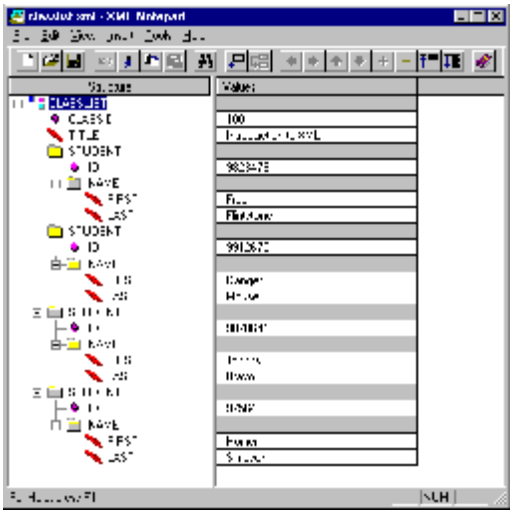
All the files that you subsequently create as you do this exercise should be contained in this directory.

Create the *ClassList.xml* Data File

The XML Notepad tool is a simple editor for XML-structured data. You should use it to create the file *ClassList.xml*.

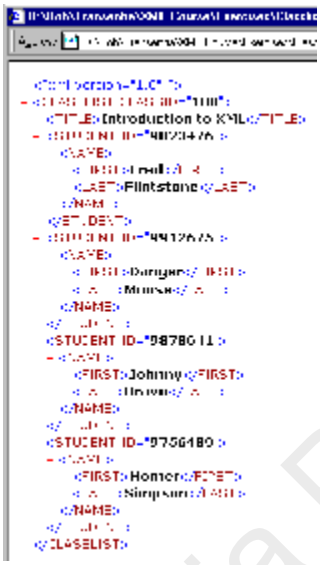
Unlike a 'normal' text editor, the XML Notepad 'understands' XML and lets you enter data in a way that conforms to the central ideas of XML: elements, attributes, etc.

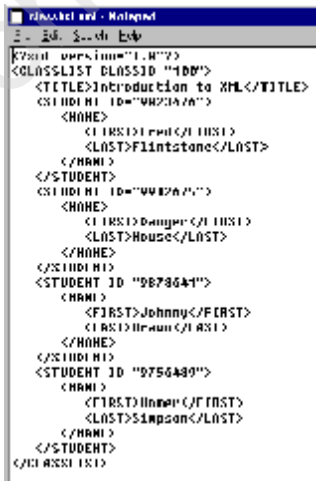
The file you create should have the following appearance:



Notice how the tree-structure of the document is very clearly defined.

If you save this file and then double-click on it, IE5 will display the created XML file, as shown at left. The pure text file is shown at right.





You should experiment with IE5's ability to collapse and expand XML elements.

Check the File for Well-Formedness

To ensure that your file is correct, you should check it to ensure that it is well-formed.

You can do this using Oracle's Java-based parser.

Ensure that your Java CLASSPATH is set appropriately (*the example given here assumes that Oracle's parser is installed into C:\xmlparser_v2_0_2_7. you should 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%
```

To check your XML file, enter the following command into an MS-DOS window:

```
C:\ClassList> C:\xmlparser_v2_0_2_7\bin\nt\oraxml.bat ClassList.xml
```

You will be told if there are errors in the file that make it not well-formed.

You should 'play' with your XML file: insert various errors into the file (missing angle brackets, missing tags, missing attribute, etc.) and see how the checking process responds.

Adding an Internal DTD Subset

8

Although it is possible to ensure that your file is well formed and hence *structurally* sound (XML Notepad is a useful—but simple—tool to ensure this), it is still not possible to ensure that the *data* represented by file is correct. To allow for this, a DTD is needed.

Modify the *ClassList.xml* document to add an internal DTD at the head of the file, as shown below:

```
<?xml version="1.0"?>
<!DOCTYPE CLASSLIST [
<!ELEMENT CLASSLIST (TITLE, STUDENT*)>
<!ATTLIST CLASSLIST CLASSID CDATA #REQUIRED>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT STUDENT (NAME)>
<!ATTLIST STUDENT ID CDATA #REQUIRED>
<!ELEMENT NAME (FIRST, LAST)>
<!ELEMENT FIRST (#PCDATA)>
<!ELEMENT LAST (#PCDATA)>
]>
<CLASSLIST CLASSID="100">
   <TITLE>Introduction to XML</TITLE>
   <STUDENT ID="9823476">
   …
```

Take time to ensure that you understand how this DTD corresponds to the structure of the ClassList document…

Now that the document has an associated DTD, it is possible to determine whether the file is both well-formed and valid. You should use a similar command to that which you used before:

```
C:\ClassList> C:\xmlparser_v2_0_2_7\bin\nt\oraxml.bat -v -w ClassList.xml
```

Once again, any errors in the file will be reported to you so you can experiment by introducing the same kinds of errors that you did before (this time, you will see a somewhat different behaviour: rather less 'forgiving' of problems…)

Using a Schema

The DTD that you added in the previous section allows checking for validity as well as well-formedness. A DTD is a rather 'loose' tool: it enables an XML parser to say whether data is missing, out of sequence, etc. but it does not make it possible to detect problems such as:

```
<CLASSLIST ID="Hello, World">
```

The DTD mechanism considers `ID="Hello, World"` to be a legal instance of CDATA, even though it is clearly not a course ID.

Schemas are designed to improve upon this state of affairs.

To see the value of schemas, create the file *ClassListSchema.xml* containing the following:

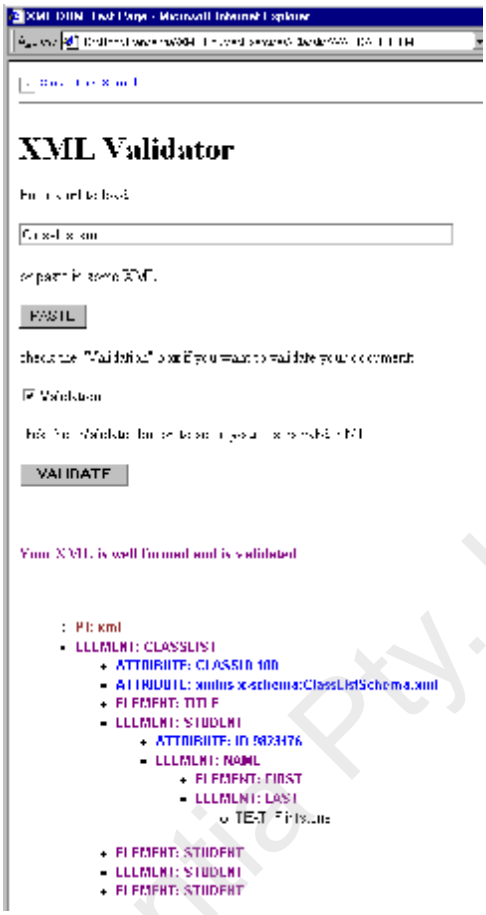
```
<?xml version="1.0"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
        xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <AttributeType name="ID" dt:type="ui4"/>
  <ElementType name="FIRST" dt:type="string"/>
  <ElementType name="LAST" dt:type="string"/>
  <ElementType name="NAME" content="eltOnly">
    <element type="FIRST" minOccurs="1" maxOccurs="1"/>
    <element type="LAST" minOccurs="1" maxOccurs="1"/>
  </ElementType>
  <ElementType name="STUDENT" content="eltOnly">
    <attribute type="ID"/>
    <element type="NAME"/>
  </ElementType>
  <ElementType name="TITLE" dt:type="string"/>
  <AttributeType name="CLASSID" dt:type="ui1"/>
```

```
    <ElementType name="CLASSLIST" content="eltOnly">
      <attribute type="CLASSID"/>
      <element type="TITLE" minOccurs="1" maxOccurs="1"/>
      <element type="STUDENT" minOccurs="0" maxOccurs="*"/>
    </ElementType>
  </Schema>
```

8 You will also need remove the DOCTYPE prologue section at the start of the *ClassList.xml* file and edit the CLASSLIST opening tag at the beginning of the file to resemble the following:

```
<?xml version="1.0"?>
<CLASSLIST CLASSID="100" xmlns="x-schema:ClassListSchema.xml">
```

You can now use the Microsoft *Validate.htm* page (open in IE5) to check your file for well-formedness and validity:



Once again, you should experiment with the file. In particular make changes to the *types* of the data contained in the attributes and rerun the validator. You should also try to remove/reorder elements.