CS 683 Emerging Technologies
Spring Semester, 2003
Doc 10 SAX, DOM, XSLT

Contents

References

Learning XML, Erik Ray, O'Reilly, 2001

http://www.saxproject.org/

http://www.w3.org/DOM/

Using XML

Creating XML Applications

Transform XML to display it

Distributed Computing with XML

Creating XML Applications

Computer program that uses XML

Sample XML applications

- uPortal Channels
- Microsoft Word

The application usually must

- Parse XML
- Generate XML

XML Parsers

A number of XML parsers are available for use in applications

Sun's Java based XML Parser (JAXP)

http://java.sun.com/xml/xml_jaxp.html

Included in JDK 1.4

Xerces from Apache

http://xml.apache.org/

Parsers by Jim Clark

- Expat
  - Open source
  - Fast parser written in C

  http://www.jclark.com/xml/expat.html
- XP
  - Java based parser

  http://www.jclark.com/xml/xp/index.html

VisualWorks Smalltalk includes an XML Parser

Example of Using Sun's Java XML Parser

File sample.xml

```
<?xml version="1.0" ?>
<greetings>
        <from>
                <name>Roger</name>
        </from>
        <to>
                <name>World</name>
        </to>
        <message>
                Hi
        </message>
</greetings>
```

XMLParseExample.java

```
import javax.xml.parsers.*;
import org.xml.sax.*;
import java.io.*;
import org.w3c.dom.*;

public class Test {
   public static void main(String argv[]) throws Exception {
   DocumentBuilderFactory factory;
   factory = DocumentBuilderFactory.newInstance();

   DocumentBuilder builder = factory.newDocumentBuilder();

   Document document;
   document = builder.parse( new File("sample.xml") );

   NodeList names = document.getElementsByTagName("name");
   printNodes( names);
   NodeList message = document.getElementsByTagName("message");
   printNodes( message);
   }

   public static void printNodes(NodeList list) {
      for (int k = 0; k< list.getLength(); k++ ) {
         Node tag = list.item(k);
         Node child = tag.getFirstChild();
         System.out.print(tag.getNodeName() + "\t");
         System.out.println( child.getNodeValue());
      }
   }
}
```

Output

```
name Roger
name World
message
      Hi
```

Document Object Model (DOM)

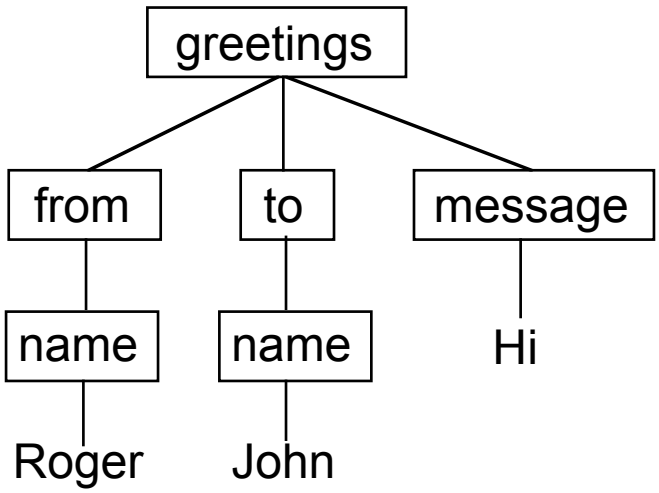
The statement:

```
builder.parse( new File("sample.xml") );
```

Converts:

```
<greetings>
    <from>
        <name>Roger</name>
    </from>
    <to>
        <name>World</name>
    </to>
    <message>
        Hi
    </message>
</greetings>
```

into the tree structure



Document Object Model (DOM)

DOM is a W3C standard

The DOM is a platform- and language-neutral interface

Allows programs to

- Access XML documents
- Modify XML documents
- Create XML documents

Simple API for XML (SAX)

For some applications DOM may be overly complex

SAX is a simpler event driven API for parsing XML

SAX is common in Java XML parsers,

SAX version 1 & 2 exists

Not a W3C standard

Basic SAX Operation

Your Sax handler class either

- Extends org.xml.sax.helpers.DefaultHandler
- Implements org.xml.sax.ContentHandler

DefaultHandler

- Implements interface org.xml.sax.ContentHandler
- Methods implement a default behavior

You register you handler object with the parser

Parse calls back to your object with information about the xml document

Some Basic Methods in DefaultHandler

void **startDocument** ()
Receive notification of the beginning of the document.

void **endDocument** ()
Receive notification of the end of the document.

void **startElement** (String uri, String localName, String qName, Attributes attributes)
Receive notification of the start of an element.

void **endElement** (String uri, String localName, String qName)
Receive notification of the end of an element.

void characters (char[] ch, int start, int length)
Receive notification of character data inside an element.

void **ignorableWhitespace** (char[] ch, int start, int length)
Receive notification of ignorable whitespace in element content.

void **error** (SAXParseException e)
Receive notification of a recoverable parser error.

void **fatalError** (SAXParseException e)
Report a fatal XML parsing error.

void **warning** (SAXParseException e)
Receive notification of a parser warning.

SAX 1 Example

```
import org.xml.sax.helpers.DefaultHandler;
import org.xml.sax.Attributes;
import javax.xml.parsers.SAXParserFactory;
import javax.xml.parsers.SAXParser;
import java.util.Vector;
import java.io.File;

public class SAXDriverExample extends DefaultHandler
   {
   private Vector root;
   String currentTag;

   public static void main(String argv[])
      {
      SAXDriverExample handler = new SAXDriverExample();

      // Use the default (non-validating) parser
      SAXParserFactory factory = SAXParserFactory.newInstance();
      try
         {
         SAXParser saxParser = factory.newSAXParser();
         saxParser.parse( new File("sample"), handler );
         }
      catch (Throwable t)
         {
         t.printStackTrace();
         }
      System.out.println( handler.root());
      }
```

```
   public void startDocument()
      {
      root = new Vector();
      }

   public void characters(char[] ch, int start, int length)
      {
      root.addElement(currentTag + "(" +  new String(ch,start,+length) +
          ")[" + start+":"+ length+"]\n");
      }

   public void startElement(String  uri, String  localName, String qName,
         Attributes attributes)
      {
      currentTag = qName;
      }

   public void endElement(String  uri, String  localName, String qName
)
      {
      currentTag = "None";
      }

   public Vector root()
      {
      return root;
      }
   }
```

Sample Run

Sample File

```
<?xml version="1.0" ?>
<Sample>
	<a>hi</a>
	<b>mom</b>
	<b>how are</b>
	<a>you</a>
</Sample>
```

Output

```
[Sample(
	)[9:2]
, a(hi)[14:2]
, None(
	)[20:2]
, b(mom)[25:3]
, None(
	)[32:2]
, b(how are)[37:7]
, None(
	)[48:2]
, a(you)[53:3]
, None(
)[60:1]
]
```

Validating Example

Modification to Main

```
public static void main(String argv[])
        {
        SAXDriverExample handler = new SAXDriverExample();

        SAXParserFactory factory =
SAXParserFactory.newInstance();
        factory.setValidating(true);
        try
                {
                SAXParser saxParser = factory.newSAXParser();
                saxParser.parse( new File("sample"), handler );
                }
        catch (Throwable t)
                {
                t.printStackTrace();
                }
        System.out.println( handler.root());
        }
```

Sample

```
<?xml version="1.0" ?>
<!DOCTYPE Sample [
<!ELEMENT Sample (  ( a* , b*)*)>
<!ELEMENT a (#PCDATA)>
<!ELEMENT b (#PCDATA)> ]>

<Sample>
        <a>hi</a>
        <b>mom</b>
        <b>how are</b>
        <a>you</a>
</Sample>
```

Output

```
[a(hi)[117:2]
, b(mom)[128:3]
, b(how are)[140:7]
, a(you)[156:3]
]
```

SAX 2 Example

```
import org.xml.sax.helpers.DefaultHandler;
import org.xml.sax.helpers.XMLReaderFactory;
import org.xml.sax.InputSource;
import org.xml.sax.XMLReader;
import org.xml.sax.Attributes;

import java.util.Vector;
import java.io.FileReader;

public class SAXDriverExample extends DefaultHandler
   {
   private Vector root;
   String currentTag;

   public static void main(String argv[])
      {
      SAXDriverExample handler = new SAXDriverExample();
      try
         {
         XMLReader saxParser =
XMLReaderFactory.createXMLReader();
         saxParser.setContentHandler( handler);
         saxParser.parse( new InputSource(new FileReader("sample")) );
         }
      catch (Throwable t)
         {
         t.printStackTrace();
         }
      System.out.println( handler.root());
      }
```

Rest of program is the same

Running the Example

```
java -Dorg.xml.sax.driver=org.apache.crimson.parser.XMLReaderImpl
        SAXDriverExample
```

Output

```
[a(hi)[139:2]
, b(mom)[150:3]
, b(how are)[162:7]
, a(you)[178:3]
]
```

The org.xml.sax.driver property can be set inside your program

Some XML Parsers for Java

`gnu.xml.aelfred2.SAXDriver`	Lightweight non-validating parser; Free Software
`gnu.xml.aelfred2.XmlReader`	Optionally validates; Free Software
`oracle.xml.parser.v2.SAXParser`	Optionally validates; proprietary
`org.apache.crimson.parser.XMLReaderImpl`	Optionally validates; used in JDK 1.4; Open Source
`org.apache.xerces.parsers.SAXParser`	Optionally validates; Open Source

The following has links to the above XML parsers

http://www.saxproject.org/?selected=links

Validating Example

```
  public static void main(String argv[])
    {
    SAXDriverExample handler = new SAXDriverExample();
    try
      {
      XMLReader saxParser =
XMLReaderFactory.createXMLReader();
      saxParser.setFeature("http://xml.org/sax/features/validation",
true);
      saxParser.setContentHandler( handler);
      saxParser.parse( new InputSource(new FileReader("sample")) );
      }
    catch (Throwable t)
      {
      t.printStackTrace();
      }
    System.out.println( handler.root());
    }
```

Displaying XML
Adding Presentation Layer to XML

Cascading Style Sheets (CSS)

- An early presentation layer for Web browsers
- Formats and displays XML documents

Extensible Stylesheet Language for Transformation (XSLT)

- Presentation layer for XML
- Subset of XSL
- Used uPortal
- Based on transformations rules
- A rule matches XML elements
- Each rule specifies output in XML

XHTML

- A version of HTML that is also XML

Hello World Example
File hello.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="hello.xsl"?>
<greetings>Hello World</greetings>
```

File hello.xsl

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
  <html>
  <body>
    <center><b>
    <xsl:value-of select="greetings"/>
    </b></center>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

Instructions

Place both files in the same directory
Open hello.xml with IE 5

Displayed In IE
Hello World

Hello World Example Explained
File hello.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="hello.xsl"?>
<greetings>Hello World</greetings>
```

<?xml-stylesheet type="text/xsl" href="hello.xsl"?>

A processing instruction

- Tells the application to apply a stylesheet
- Gives the type of style sheet: xsl
- Gives the location of the style

Hello World Example Explained
File hello.xsl

<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">

xmlns:xsl="http://www.w3.org/TR/WD-xsl"

Declares an XML namespace

Namespaces avoid name collisions

xsl:stylesheet

The name of the tag

The tag "stylesheet" in the xsl name space

Hello World Example Explained
File hello.xsl

```
<xsl:template match="/">
  <html>
  <body>
    <center><b>
    <xsl:value-of select="greetings"/>
    </b></center>
  </body>
  </html>
</xsl:template>
```

<xsl:template match="/">

Define a rule that matches all elements

Contents of rule is the output

<xsl:value-of select="greetings"/>

Replaced with the contents of tag "greetings"

Hello World Example Explained
Output of Transformation

```
<html>
  <body>
    <center>
      <b>Hello World</b>
    </center>
  </body>
</html>
```

CD Catalog Example
Using a Loop

File catalog.xml

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="cd.xsl"?>
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>

  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
</CATALOG>
```

CD Catalog Example

File cd.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
  <html>
  <body>
    <table border="2">
      <tr>
        <th>Title</th>
        <th>Artist</th>
      </tr>
      <xsl:for-each select="CATALOG/CD">
      <tr>
        <td><xsl:value-of select="TITLE"/></td>
        <td><xsl:value-of select="ARTIST"/></td>
      </tr>
      </xsl:for-each>
    </table>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

Result Displayed in IE 5

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler

Where to process XSLT

Examples used IE to process XSLT

But

- Not all browsers support XSLT
- IE was a bit slow in processing the XSLT

Use Web server to process XSLT

- Sun's Java API for XML can be used to process XSLT
- Web server sends HTML to browser