

Java XML XSLT Overview

Java XML XSLT Overview

Instructor: Rick Palmer, SCWCD

rick@online-ettraining.com

Topics Covered

- ❑ XML Parsers
- ❑ DOM Processing with JAXP
- ❑ XSLT Transformations with Java

XML Parsers

- ❑ “Wrap” XML documents with programmatic interface.
- ❑ Turn XML tags into objects in memory.
- ❑ Ensure XML documents “follow the rules”.
 - ⇒ Well-formed (required)
 - ⇒ Valid (adheres to DTD or XML Schema spec)
 - ❑ Optional setting
 - ❑ Validation is turned off by default
- ❑ Perform XSLT transformations.

Flavors of Java XML Parsers

□ DOM

- ⇒ Tree-based API (nodes, elements, attributes)
- ⇒ Represents the document's logical structure
- ⇒ Allows accessing, traversing, modifying, and generating XML documents
- ⇒ Loads entire XML document into memory

□ SAX

- ⇒ Event-based callback API
- ⇒ Does not load entire XML document into memory
- ⇒ Can be much faster for specific element lookups

Main Java XML Interfaces

- ❑ org.w3c.dom package
 - ⇒ Node (Superinterface)
 - ❑ Document (Subinterfaces)
 - ❑ Element
 - ❑ Attr
 - ⇒ NodeList
 - ❑ Access Nodes by collection index
 - ⇒ NamedNodeMap
 - ❑ Access Nodes by name

Java XML Parser Classes (JAXP)

- ❑ Parser classes – `javax.xml.parsers.*`
 - ⇒ `DocumentBuilderFactory`
 - ⇒ `DocumentBuilder`
- ❑ Transformer classes – `javax.xml.transform.*`
 - ⇒ `TransformerFactory`
 - ⇒ `Transformer`
- ❑ Transformation classes
 - ⇒ `DOMSource`, `DOMResult` – `javax.xml.transform.dom.*`
 - ⇒ `SAXSource`, `SAXResult` – `javax.xml.transform.sax.*`
 - ⇒ `StreamSource`, `StreamResult` – `javax.xml.transform.stream.*`

Obtain a DOM parser object

```
import javax.xml.parsers.*;



---



// Create a DOM factory and builder
DocumentBuilderFactory dbfFactory = null;
DocumentBuilder domBuilder = null;

try {
    // Create a DocumentBuilder (DOM Parser)
    dbfFactory = DocumentBuilderFactory.newInstance();
    domBuilder = dbfFactory.newDocumentBuilder();
}
catch (ParserConfigurationException pce) {
    pce.printStackTrace();
}
```

Load XML from a File or String

```
import javax.xml.parsers.*;
import org.w3c.dom.Document;



---



// Load XML from a File
Document docXMLFile = domBuilder.parse("C:\\Java\\employees.xml");

// Load XML from a String variable
String strXML = "<person><name>Jeff</name></person>";
Document docXMLString =
    domBuilder.parse(new StringBufferInputStream(strXML));
```

Create DOM Elements/Attributes

- ❑ Create element; then append to parent

```
// Create empty XML Document
Document docXMLDoc = domBuilder.newDocument();

// Create a person element
Element elmPerson = docXMLDoc.createElement("person");

// Create name attribute and set its value to "Jeff"
elmPerson.setAttribute("name", "Jeff");

// Attach person element to the XML Doc
docXMLDoc.appendChild(elmPerson );
```

Resulting XML:

```
<person name="Jeff" />
```

Search for XML Elements (XPath)

```
import org.apache.xpath.*;
import javax.xml.transform.*;



---



Node nodeCompany = null;
Element elmCompany = null;

try {
    // Use XPath expression to find a specific XML node/element
    nodeCompany = XPathAPI.selectSingleNode(docXMLDoc, "//company");
    elmCompany = (Element) nodeCompany;

    // Set/Create an attribute called name (<company name="PCC">)
    elmCompany.setAttribute("name", "PCC");
}
catch (TransformerException tfe) { tfe.printStackTrace(); }
```

Set the Text Node Value

```
// Get the child nodes (text, attributes, other elements, etc)
NodeList ndlChildren = xmlNode.getChildNodes();

// Iterate to find the Text node
Node nodeTemp = null;
Node nodeText = null;

int iMaxNodes = ndlChildren.getLength();
for (int i = 0; i < iMaxNodes; i++) {
    nodeTemp = ndlChildren.item(i);
    if (nodeTemp.getNodeType() == Node.TEXT_NODE) {
        nodeText = nodeTemp;
        break;
    }
}

// Set the text node value (<company>Oracle</company>)
nodeText.setNodeValue("Oracle");
```

Transform XML with Java and XSLT

```
try {  
  
    // Load StreamSource objects with XML and XSLT files  
    StreamSource xmlSource =  
        new StreamSource( new File("input.xml") );  
    StreamSource xsltSource =  
        new StreamSource( new File("format.xslt") );  
  
    // Create a StreamResult pointing to the output file  
    StreamResult fileResult =  
        new StreamResult(new FileOutputStream("output.xml"));  
  
    // Load a Transformer object and perform the transformation  
    TransformerFactory tfFactory =  
        TransformerFactory.newInstance();  
    Transformer tf = tfFactory.newTransformer(xsltSource);  
    tf.transform(xmlSource, fileResult);  
}  
catch (Exception e) { e.printStackTrace(); }
```

Get an XML String from a DOM

```
import javax.xml.transform.*;
import javax.xml.transform.dom.*;
import javax.xml.transform.stream.*;

public String getDocumentXML(Document doc) {

    //Create a Transformer object
    TransformerFactory tfFactory = TransformerFactory.newInstance();
    Transformer tf = tfFactory.newTransformer();
    tf.setOutputProperty(OutputKeys.INDENT, "yes"); //keep lines and tabs

    //Create a DOMSource, pointing to the source XML Document
    DOMSource domSource = new DOMSource(doc);

    //Serialize DOMSource to a String
    StringWriter strWriter = new StringWriter();
    tf.transform(domSource, new StreamResult(strWriter));

    //Get and return the XML String
    return strWriter.toString();
}
```

Save an XML String to a File

- Same process as serializing a DOM to a String
 - ⇒ Use a `FileWriter` class instead of a `StringWriter`

```
// Create a Transformer object without specifying a stylesheet
TransformerFactory tfFactory = TransformerFactory.newInstance();
Transformer tf = tfFactory.newTransformer();

// Create a DOMSource, pointing to the source XML Document
DOMSource domSource = new DOMSource(doc);

// Create a file writer, pointing to an output file
FileWriter fileWriter = new FileWriter("output.xml");

// Perform the transformation
tf.transform(domSource, new StreamResult(fileWriter));
```

Resources

❑ Java Web Services Tutorial:

<http://java.sun.com/webservices/docs/1.6/tutorial/doc/>

❑ Apache XML Project: <http://xml.apache.org/>

❑ Java Guru: <http://www.jguru.com/>

❑ Java World: <http://www.javaworld.com/>

❑ MSDN XSLT Reference:

<http://msdn.microsoft.com/library/en-us/xmlsdk/html/xmrefxsltrefer>