

Parsing, Validating and Manipulating XML using Xerces 2.2.0

Tom Hickerson and Jun Xu

Akaza Research Developer's Seminar
Tuesday June 15th, 2004

June 15, 2004 : 1 :: 11



Presentation Roadmap

- Overview of existing packages to work with XML
 - » DOM (Document Object Model)
 - » SAX (Simple API for XML)
- How Xerces works with DOM and SAX
- Validating XML: the basics
 - » Document Type Definition
 - » XML Schema Documents
- How Xerces works with parsing and validating XML
- Other features used by XML Schemas
- Examples and Case study: PhenoNet Data Import

June 15, 2004 : 2 :: 11



Overview of packages that work with XML and Java

- Document Object Model
 - » Works with XML in the form of a tree structure
 - » Can add to and take from the XML dynamically
 - » Used for parsing, creating, modifying

- Simple API for XML
 - » It's called 'Simple' for a reason
 - » Works with XML as a stream of events
 - » Used for parsing, read-only

June 15, 2004 : 3 :: 11



What is Xerces? And how does it work with XML?

- Xerces2 Java Parser 2.6.2 is a fully conforming XML Schema version 1.0 processor, according to w3.org
 - » Implementations also exist in C++ and Perl

- Classes used with DOM and SAX are also in Xerces
 - » As an "endorsed standard", standard J2EE classes can be used with Xerces

- Xerces also supports namespaces, version 1.1

June 15, 2004 : 4 :: 11



Validating XML: the basics

- Validation can be done on one of two ways:
 - Document Type Definition
 - XML Schemas
- DTD has its own syntax
- XML Schemas are written in XML, and allow for more control

```
> <?xml version="1.0"?>
> <!DOCTYPE quotations [
> <ELEMENT quotations (quote*)>
> <ELEMENT quote (saying, attribution, era)>
> <!ATTLIST quote quoteid CDATA #REQUIRED>
> <ELEMENT saying (#PCDATA)>
> <ELEMENT attribution (#PCDATA)>
> <ELEMENT era (#PCDATA)>
> ]>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="quotations">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="quote" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="saying" type="xs:string"/>
              <xs:element name="attribution" type="xs:string"/>
              <xs:element name="era" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="quoteid" type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

June 15, 2004 : 5 :: 11



How Xerces works with Parsing XML

- Using DOM, one can just use the `javax.xml.parsers.DocumentBuilder` class
- In Xerces, one can also use the `org.apache.xerces.parsers.DOMParser` class
- In DOM, you have to keep in mind that documents are loaded into Java in a tree structure, and use the terminology `Element`, `NodeList`, and `Node`.
- Using SAX, we can use the `org.apache.xerces.parsers.SAXParser` class
 - Since SAX is parsing the entire document in a stream of events, we use Handlers to process the data
 - Code example of a Handler here

June 15, 2004 : 6 :: 11



How Xerces works with Validating XML

- Xerces uses a feature architecture to turn capabilities off and on
 - » `parser.setFeature("http://xml.org/sax/features/validation", true);`

```
HashMap valueList = new HashMap();
DOMParser parser = new DOMParser();
Document doc = null;
try {
    parser.setFeature("http://xml.org/sax/features/validation", true);
    parser.setFeature("http://apache.org/xml/features/validation/schema", true);
} catch (SAXNotRecognizedException ne) {
    logger.warning(ne.getMessage());
    System.out.println("Unrecognized feature: ");
    System.out.println("http://xml.org/sax/features/validation");
} catch (SAXNotSupportedException se) {
    logger.warning(se.getMessage());
    System.out.println("Unrecognized feature: ");
    System.out.println("http://xml.org/sax/features/validation");
}

try {
    parser.parse(dir + f.getName());
    doc = parser.getDocument();
    request = this.processDocument(doc, "GetRecord", request);
    request.setAttribute("valueList", valueList);
} catch (IOException ie) {
    System.out.println("Could not read file.");
    logger.warning(ie.getMessage());
} catch (SAXException se) {
    System.out.println("Could not create Document: ");
    logger.warning(se.getMessage());
    request.setAttribute("errorMessage", se.getMessage());
}
```

June 15, 2004 : 7 :: 11



Other Features used by XML, Xerces and Java

- Namespaces
 - » Another advantage of XML Schemas is to be able to declare one or more Namespaces, as part of the validation process
 - » In the PhenoNet project, we will be using the OAI Namespace and XML Schema:

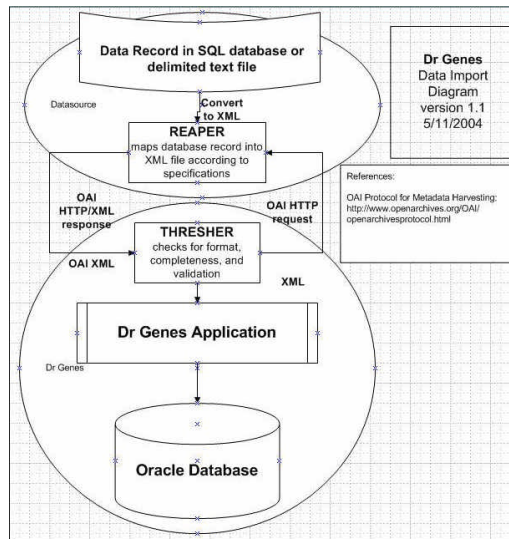
```
<?xml version="1.0" encoding="UTF-8" ?>
<ListRecords
    xmlns="http://www.openarchives.org/OAI/1.1/OAI_ListRecords"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/1.1/OAI_ListRecords
        http://www.openarchives.org/OAI/1.1/OAI_ListRecord.xsd">
```

- XSLT
 - » Using another package called Xalan for Java 2, can work together with Xerces' DOMSource and DOMResult classes to transform XML using XSL stylesheets and templates
 - » No usage yet in our current projects, but could be useful for presenting data that is only available in XML format
 - » Code samples online at <http://xml.apache.org/xalan-j/usagepatterns.html>

June 15, 2004 : 8 :: 11



Case Study: The PhenoNet Data Import



June 15, 2004 : 9 :: 11



Case Study Continued: The PhenoNet Data Import

- Reapers can take data from any source and map it to the OAI specification
- Threshers can then parse and validate the data, before beginning to insert it into the database
- If we were to use a DTD, validation would be a tricky work-around
- Since we can validate using a Namespace/XML Schema, validation can be a few lines of code in Java using Xerces

June 15, 2004 : 10 :: 11



Resources and Links

- IBM DeveloperWorks tutorials:
 - » <http://www-136.ibm.com/developerworks/xml>
- Xml.apache.org, home to Xerces:
 - » <http://xml.apache.org/>
 - » <http://xml.apache.org/xerces2-j/index.html>
 - » <http://xml.apache.org/xerces2-j/features.html>
- Xalan and Design patterns for XSLT:
 - » <http://xml.apache.org/xalan-j/usagepatterns.html#embed>
- Java and XML topics through O'Reilly:
 - » http://www.onjava.com/topics/java/java_xml
- Java API for XML Processing:
 - » <http://java.sun.com/xml/jaxp/reference/faqs/index.html>
- OAI Protocol:
 - » <http://www.openarchives.org/OAI/openarchivesprotocol.html#OAI PMHschema>