

JAXP and JAXB

CSE 486/586
Sept 14, 2004

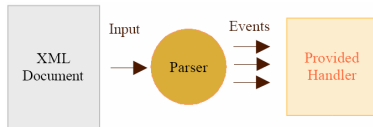
References:
ftp://ftp.oreilly.com/pub/conference/java2001/Hunter_et_al_jaxp.pdf
<http://java.sun.com/developer/technicalArticles/WebServices/jaxb/>

JAXP: Java API for XML Processing

- Accessing XML documents from code
- Generally more lightweight
- JAXP using:
 - Callback driven: SAX
 - Tree-based: DOM
- Requires complete knowledge of XML schema

SAX: Simple API for XML

- Very lightweight
- Difficult to program
- Supports only serial event-based access.
- Hence, no in-memory manipulation of data provided for.
- Package: `org.xml.sax`

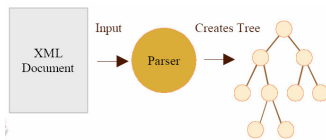


SAX: Simple API for XML

- `ContentHandler` is called when "events" occur
- The following "callback" methods are implemented by `ContentHandler` object:
 - `startDocument()` throws `SAXException`
Executed when any starting tag is encountered
 - `endDocument()` throws `SAXException`
Executed when any ending tag is encountered
 - `characters(char ch[], int start, int end)` throws `SAXException`
Executed when any "text" is encountered
- Try out example `RSSReader`. Will post link in the newsgroup.

DOM: Document Object Model

- Package: `org.w3c.dom`
- The document must be built and populated in the memory.
- The document is represented as a tree of objects.



DOM: Document Object Model

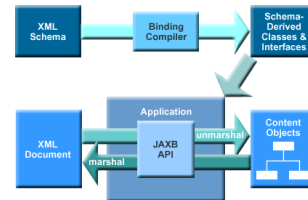
- So your code must first create a `org.w3c.dom.Document` object.
- Then it must create an `org.w3c.dom.Element` object for every element in the hierarchy of the XML document.
- Each element is then added to the object by using the `Element.appendChild(Element)` method.
- Text in the XML documents are represented as `Text` objects. They are appended to the elements that contain them using the `Element.appendChild(Text)` method.
- Once you have built the tree, you can use DOM methods to navigate the tree.

DOM: Document Object Model

- You can transform this tree to and from XML files using the `java.xml.transform.Transformer` Class
- DOM provides for non-sequential access, although the tree has to be navigated through.
- Try out example UsingDOM. Will post link in the newsgroup
- Java 1.4.2 also has option to verify the XML file using a schema (.xsd) file.

JAXB: Java Architecture for XML Binding

- JAXB allows access to XML documents without having to know their schema and XML parsing in general.



JAXB: Java Architecture for XML Binding

- Steps involved:
 - Binding
 - The first step is to bind the schema (.xsd) file into a set of Java classes that represent the schema.
 - This is done by using a binding compiler (xjc):
`xjc.sh -p test books.xsd`
 - The compiler generates a set of interfaces and a set of classes that implement these interfaces.
 - Compile these classes
- Now you are ready to use JAXB API in your program

JAXB: Java Architecture for XML Binding

- Unmarshalling:
 - Unmarshalling a document means creating a tree of content objects that represent the content and organization of the document.
 - These content objects are instances of classes provided by the binding compiler.
 - Methods in these objects are used to get and set data for each element and attribute in the schema.
- Marshalling:
 - Marshalling a document means converting the content objects into an XML document.