XML PARSER- SAX Part-II

UNIT-II

SAX Validation

- SAX parsers can be validating and nonvalidating
- Validating parsers validates XML document based on a Document Type Definition (DTD) or Schema
- Apache Xerces SAX is a validating parser
- For validation, validation feature should be set to true
- If turned on for non-validating parser, SAXNotSupportedException will be thrown
- If the parser does not recognize the feature, a SAXNotRecognizedException will be thrown

Example for SAX Validation

```
public static void main(String args[]) throws Exception {
……
XMLReader parser = XMLReaderFactory.createXMLReader(
“org.apache.xerces.parsers.SAXParser”);
parser.setFeature(“http://xml.org/sax/features/validation”,
   true);
SAXValidator handler = new SAXValidator();
parser.setContentHandler(handler);
```

Example Contd…

```
parser.parse(new InputSource(new FileReader(args[0])));
if (!handler.isWellFormed()) {
System.out.println(“Document is NOT well formed.”);
}
if (!handler.isValid()) {
System.out.println(“Document is NOT valid.”);
}
if (handler.isWellFormed() && handler.isValid()) {
System.out.println(“Document is well formed and valid.”);
}
```

Handling Errors in SAX

- Locator interface gives parse position within a ContentHandler method
- Position information includes line number and column number
- Locator object should *not be used in any other methods, including*
- ErrorHandler methods
- But ErrorHandler methods also gives a SAXParseException object, that gives the position information

Example for SAX Error Handling

- public static void main(String args[]) throws Exception {
- ………
- XMLReader parser = XMLReaderFactory.createXMLReader(
- “org.apache.xerces.parsers.SAXParser”);
- parser.setFeature(“http://xml.org/sax/features/validation”, true);
- SAXErrors handler = new SAXErrors();
- parser.setContentHandler(handler);
- **parser.setErrorHandler(handler);**
- parser.parse(new InputSource(new FileReader(args[0])));
- }

Using Locator Object (Contd...)

- public void setDocumentLocator(Locator inLocator) {
- System.out.println(“***Got Locator***”);
- locator = inLocator;
- int line = locator.getLineNumber();
- int column = locator.getColumnNumber();
- System.out.println(“Line “ + line + “, column “ + column);
- }

Using SAXParseException (Contd…)

```
public void error(SAXParseException e) {
printLocation(e);
……..
}
public void printLocation(SAXParseException e) {
int line = e.getLineNumber();
int column = e.getColumnNumber();
System.out.println(“Line “ + line + “, column “ + column);
….
}
```

Resolving Entity References in SAX

- SAX parsers will resolve entity references automatically
- It also allows the developer to resolve an entity reference
- EntityResolver is implemented by DefaultHandler
- EntityResolver contains only one method, resolveEntity()
- It will be called with the system ID and public ID, depending on how the entity is defined
- It must return an InputSource pointing to where the entity is stored

Example resolving entity reference

- public static void main(String args[]) throws Exception {
- …….
- }
- XMLReader parser = XMLReaderFactory.createXMLReader(
- “org.apache.xerces.parsers.SAXParser”);
- parser.setFeature(“http://xml.org/sax/features/validation”, true);
- SAXEntity handler = new SAXEntity();
- parser.setContentHandler(handler);
- **parser.setEntityResolver(handler);**
- parser.parse(new InputSource(new FileReader(args[0])));
- }

Contd…

- public InputSource resolveEntity(String publicId, String systemId) {
- try {
- if (systemId.equals(“http://www.example.com/xml/entity.txt”)) {
- return new InputSource(
- new FileReader(“entity.txt”));
- }
- } catch (IOException e) {
- }
- return null; // for default behavior
- }

Lexical Events in SAX

- Lexical Handler allows to handle events related to DTD and CDATA
- LexicalHandler is part of the org.xml.sax.ext package
- It is supported in Xerces SAX implementation as an extension package

Example for Lexical Events in SAX

- public void startDTD(String name, String publicId,
- String systemId) throws SAXException {
- System.out.print(“*** Start DTD, name“ + name);
- }
- public void endDTD() throws SAXException {
- System.out.println(“*** End DTD ***”);
- }
- public void startCDATA() throws SAXException {
- System.out.println(“*** Start CDATA ***”);
- }
- public void endCDATA() throws SAXException {
- System.out.println(“*** End CDATA ***”);
- }