9.7: XML Creation using StAX

There are multiple ways to create an XML document in Java.

One way is to simply print XML strings to a file using standard Java libraries. However, printing strings to a file can be error prone and the chance of creating malformed XML documents is high.

The preferred way is to use XMLStreamWriter. XMLStreamWriter works just like other Writer classes we have already seen. It provides methods that can help you create well formed XML documents. Methods include writeStartDocument(), writeEndDocument(), writeStartElement(), writeEndElement() and more.

The example below demonstrates using the XMLStreamWriter to create a collection with one DVD entry:

```
XMLStreamWriter xw =  null;
XMLOutputFactory xof = XMLOutputFactory.newInstance();
try {
    xw =  xof.createXMLStreamWriter(new FileWriter(fn));
    xw.writeStartDocument("1.0");
    xw.writeDTD("<!DOCTYPE  DVDS SYSTEM \"dvd.dtd\">");
    xw.writeStartElement("DVDS");
    xw.writeStartElement("DVD");
    xw.writeAttribute("id",  "4");

    xw.writeStartElement("title");
    xw.writeCharacters("Star  Wars: Attack of the Clones");
    xw.writeEndElement();

    String []  performers = {
        "Ewan  McGregor",
        "Natalie  Portman",
        "Hayden  Christensen"
    };
    xw.writeStartElement("performers");
    for (String  performer : performers) {
        xw.writeStartElement("performer");
        xw.writeCharacters(performer);
        xw.writeEndElement();
    }
    xw.writeEndElement();
    xw.writeStartElement("discs");
    xw.writeCharacters("2");
    xw.writeEndElement();
    xw.writeStartElement("price");
    xw.writeCharacters("14.99");
    xw.writeEndElement();
    xw.writeEndElement();  // close the DVD tag
    xw.writeEndElement();  // close the DVDS tag
    xw.writeEndDocument();
    xw.flush();
    xw.close();
```

```
} catch  (XMLStreamException e) {
      e.printStackTrace();
} catch (IOException e)  {
      e.printStackTrace();
}
```

The following are the steps to creating an XML document:

1. Include the following libraries:

```
import  javax.xml.stream.*;
import java.io.*;
```

2. Get a new instance of the XMLOutputFactory.
3. With the new instance of the XMLOutputFactory, create an XMLStreamWriter and pass the output writer object.
4. Start the document by calling the XMLStreamWriter's writeStartDocument( ).
5. Write the DTD/prolog line using writeDTD( ).
6. Create individual elements in the following order (with the start and end tags mirroring the nesting of the document):
   a. start tag - writeStartElement( )
   b. attributes – writeAttribute( )
   c. element text – writeCharacters( )
   d. end tag - writeEndElement( )
7. Invoke writeEndDocument( ).
8. Close the XMLStreamWriter.

As you can see, XMLStreamWriter provides a convenient and straightforward way to write XML documents.