

Distributed Systems

Lab 5 – XML

The java_xml eclipse project contains the source files and xml files for the lab. Extract the zip file and open the project in eclipse.

Parsing XML files with a DOM Parser

1. The DOMParserDemo.java class parses the student.xml file and prints some of its contents. Deliberately make syntax errors in the student.xml file to see whether the parsing happens correctly. Edit the file to print the student grades and marks as well.

Note: The given file uses a DOM Parser. Since, DOM parsers load the entire xml document to the memory, they may not be good for handling large XML files, for which you can use a SAX Parser. SAX Parsers load only a section of an XML file to the memory that it needs to query.

XML Schema validation

2. Open a command line and go into the src sub-directory. And type the command *schemagen Employee.java Address.java*. Schemagen is a java tool to generate the xml schemas based on java classes. This should generate a schema at the src directory, based on the Employee class and the Address class. Copy the schema back to the project root directory. Rename the schema to Employee.xsd.

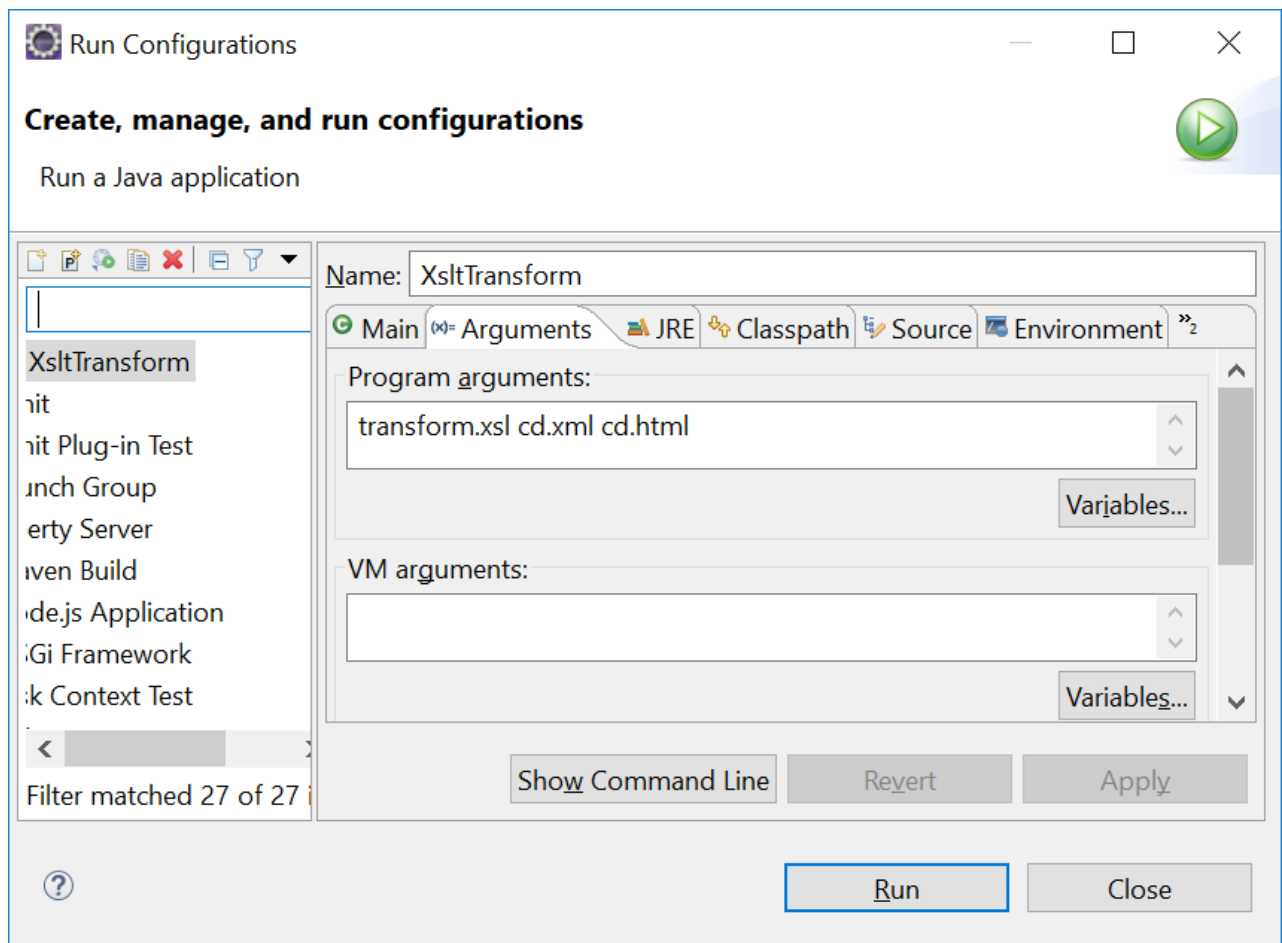
Note: For this you need to have JDK installed in your system. If the *schemagen* command doesn't work, check whether the <<Jdk installation directory>>/bin is added to your PATH system variable.

3. Run the XMLValidation.java file to see whether the given employee.xml file is valid against the generated Employee.xsd schema. Modify the employee.xml file to change its element names and data types to see whether the validation fails.

Xslt Transformations

4. Set the input parameters of the XsltTransform.java class to transform.xsl cd.xml cd.html. Run the java file to see whether the html file generates correctly. Open the cd.html file in the project directory using a browser to view it.

Note: To add input parameters, right click the XsltTransform.java file, *run as -> run configurations -> Arguments* and copy *transform.xsl cd.xml cd.html* in the program arguments window. Make sure XsltTransform is input in the *Name* field. Then click *Run*.



Edit the transformation file to print the cd price as well in the table.

Hint: See the *to do* comments in the *transform.xml* file. You can edit the xml file using any text editor (or using an XML editor tool such as XML copy editor).

<https://xml-copy-editor.sourceforge.io/>

1. Set the input parameters of the XsltTransform.java class to *stocks.xml investments.xml stocks.xml*. Run the transformation to see the output. Modify the *stocks.xml* transformation file to include the stock price as well, in the output *stocks.xml* file. (**Hint:** view the to do comments in the *stocks.xml* file)

XML Serialization

2. The SerializeToXML and DeserializeFromXML classes would serialize a movie list and deserialize a movie list in the DVD class. Run the SerializeToXML class to see the output of the dvd.xml file. Open the xml file using a text editor or an XML editor to view its contents. Modify the DVD.java class to print the movie release year as well, after deserializing it.

Hint: Check the to do comment in the DVD.java file toString() method.

Submission:

Upload the modified code as a single zip file to the courseweb link. The name of the file should be your registration number.