

Persistence

Saving and retrieving objects to/from XML files

Produced Mairead Meagher
by: Dr. Siobhán Drohan



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
<http://www.wit.ie/>

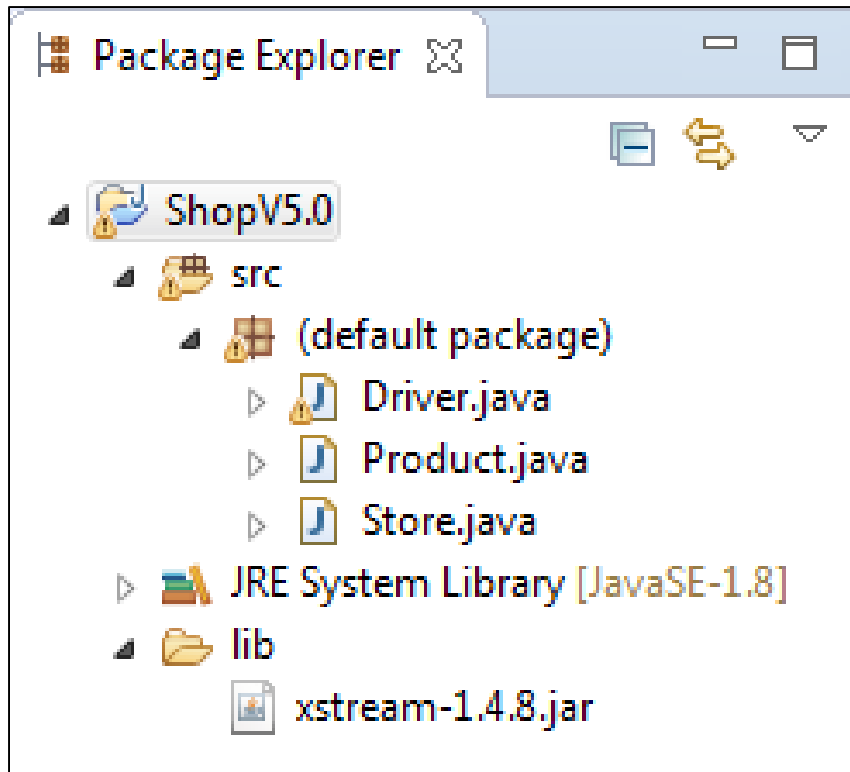
Shop V4.0 has implemented the CRUD process, but when we close down our application, all the entered data is lost.

Shop V5.0 - We will use XML to make our data persistent. We will use a component to do this for us.

Shop V5.0 (using XML)

1. Download the **xstream-1.4.8.jar** component and add it to your Shop project.
2. Store Class – write the load(), save() methods.
3. Driver Class - include extra load and save functionality to the menu.

Adding a component to the lib folder

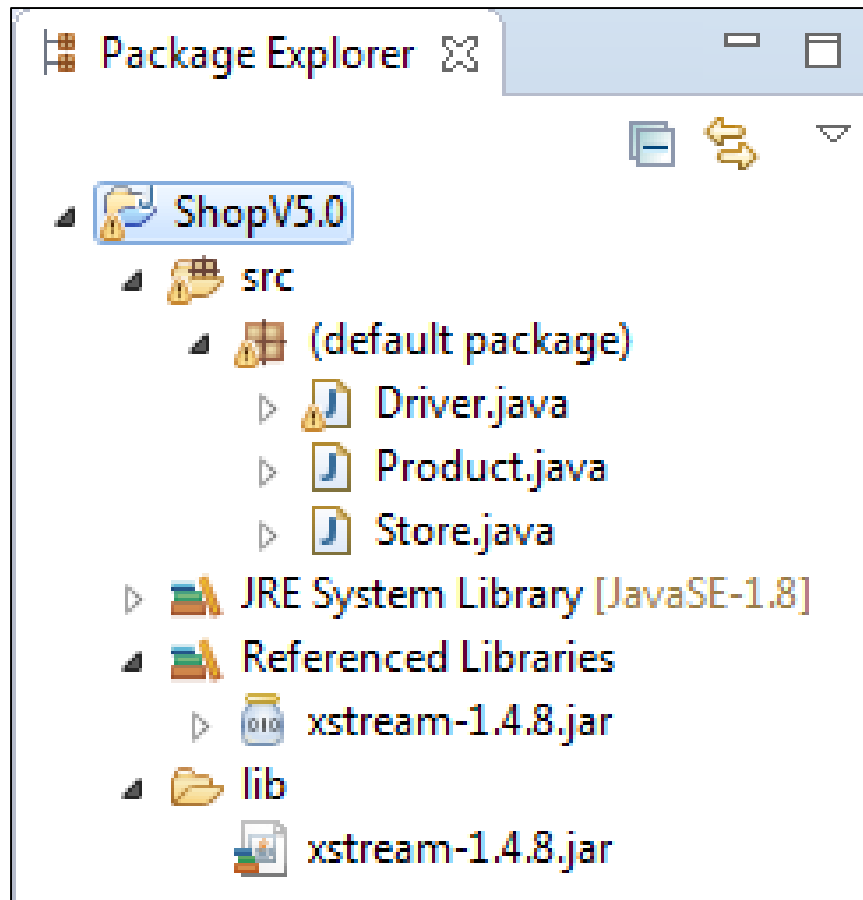


Download the xstream-1.4.8.jar component.

In Eclipse Package Explorer, create a new folder "lib" (select "File->New->Folder")

Drag the xstream-1.4.8.jar file into the lib folder (choose the option to copy the file).

Adding the component to your build path



Right click on the
xstream-1.4.8.jar file
and select:

"Build Path->Add to
Build Path".

Shop V5.0 (using XML)

1. Download the **xstream-1.4.8.jar** component and add it to your Shop project.
2. Store Class – write the load(), save() methods.
3. Driver Class - include extra load and save functionality to the menu.

Store.java

- To use this code in another project, change:
- the **type** of object stored in the ArrayList.
 - the name of the xml file
 - The name of the ArrayList object.

```
@SuppressWarnings("unchecked")
public void load() throws Exception
{
    XStream xstream = new XStream(new DomDriver());
    ObjectInputStream is = xstream.createObjectInputStream
        (new FileReader("products.xml"));
    products = (ArrayList<Product>) is.readObject();
    is.close();
}

public void save() throws Exception
{
    XStream xstream = new XStream(new DomDriver());
    ObjectOutputStream out = xstream.createObjectOutputStream
        (new FileWriter("products.xml"));
    out.writeObject(products);
    out.close();
}
```

Store.java

```
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.ObjectInputStream;  
import java.io.ObjectOutputStream;  
  
import com.thoughtworks.xstream.XStream;  
import com.thoughtworks.xstream.io.xml.DomDriver;
```

Note: you will need to import these additional packages.

Shop V5.0 (using XML)

1. Download the **xstream-1.4.8.jar** component and add it to your Shop project.
2. Store Class – write the load(), save() methods.
3. Driver Class - include extra load and save functionality to the menu.

Driver class code

```
private int mainMenu()  
{  
    System.out.println("Shop Menu");  
    System.out.println("-----");  
    System.out.println("  1) Add a Product");  
    System.out.println("  2) List the Products");  
    System.out.println("  3) Update a Product");  
    System.out.println("  4) Delete a Product");  
    System.out.println("-----");  
    System.out.println("  5) List the cheapest product");  
    System.out.println("  6) List the products in our current product line");  
    System.out.println("  7) Display average product unit cost");  
    System.out.println("  8) List products that are more expensive than a given price");  
    System.out.println("-----");  
    System.out.println("  9) Save Products to product.xml");  
    System.out.println(" 10) Load Products from product.xml");  
    System.out.println("-----");  
    System.out.println("  0) Exit");  
    System.out.print("==>> ");  
    int option = input.nextInt();  
    return option;  
}
```

Add the **Save** and **Load**
functionality to
the menu system.

```
Shop Menu  
-----  
  1) Add a Product  
  2) List the Products  
  3) Update a Product  
  4) Delete a Product  
-----  
  5) List the cheapest product  
  6) List the products in our current product line  
  7) Display average product unit cost  
  8) List products that are more expensive than a given price  
-----  
  9) Save Products to product.xml  
 10) Load Products from product.xml  
-----  
  0) Exit  
==>>
```

Driver class code

```
case 9:    try{
            store.save();
        }
        catch (Exception e){
            System.out.println("Error writing to file: " + e);
        }
        break;
case 10:   try{
            store.load();
        }
        catch (Exception e)
        {
            System.out.println("Error reading from file: " + e);
        }
        break;
```

Add the **Save** and **Load** functionality to the menu system.

Inside a try/catch block, call the save method for option 9, and the load method for option 10.



```
<object-stream>
  <list>
    <Product>
      <productName>24 inch monitor</productName>
      <productCode>3423</productCode>
      <unitCost>129.99</unitCost>
      <inCurrentProductLine>true</inCurrentProductLine>
    </Product>
    <Product>
      <productName>14 inch monitor</productName>
      <productCode>2322</productCode>
      <unitCost>109.99</unitCost>
      <inCurrentProductLine>true</inCurrentProductLine>
    </Product>
  </list>
</object-stream>
```

The XML file that is created when the save option is selected from the menu; it can be found in your project directory.

**Any
Questions?**





Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
<http://www.wit.ie/>