OSM Orchestration Exercise

Lab 8-1 - Order Item Specification

In this Lab we will continue with our configuration of the Central Order Management cartridge by configuring the '**Order item Specification**' and mapping this to the product class file.

The **order item specification** that you configure in Design Studio contributes to the **control data** that OSM adds to the created order. The control data provides the information OSM requires to execute the **orchestration plan** and includes information about the **order items** and the **order components** in the orchestration plan.

Before we create the Order item Specification, we need to create a '**Data Provider**' in Studio that points to the XML file '**productClassMapping.xml**'.

1. Right-click on your cartridge and create a new '**Data Provider**'.
   - Name : **ProductClassMappingFile**
   - Folder Name : **DATAPROVIDER**

2. Under 'Settings' set the '**Provider Type**' to 'XML File'.

3. Switch to the '**Interface**' tab and add a new parameter.

   - Current param : **oms:url** (no quotes)
   - Default Value : paste the following path to the location of the XML file.

```
'file:///u01/app/oracle/user_projects/domains/osm/productClassMapping.xml'
```

** **Note:** The quotes **must** be present around the entry.
** **Note:** This location is present in the linux VM and is the 'osm' domain root directory.
** **Note:** Currently the mapping file is not in this location yet. Please create it directly in VirtualBox.

4. Click the 'Save' button to save all changes.

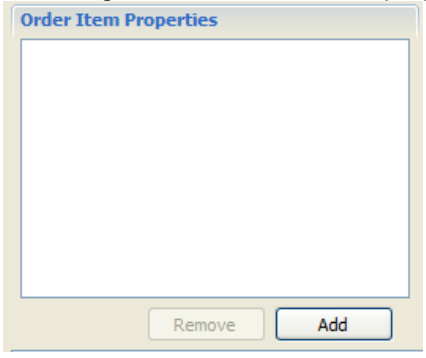
   - This 'Data Provider' is only a connection to the XML file and will be used by a '**Data Instance**' behaviour shortly.

Now we can resume with the creation of the '**Order Item Specification**' (the missing component from the Decomposition Rules).

5. Right click your cartridge and create a new '**Order Item Specification.**

   - Namespace: 'osm7Order1.com'
   - Folder Name: **LINE**
   - Spec name **OrderItemSpecification**.

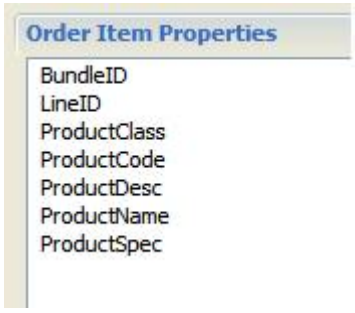
6. In the open editor you now need to add the following elements in the 'Order item Properties' window using the 'Add' button. The properties will hold key values extracted from the incoming order lines.



Add the following

- ProductCode
- ProductName
- ProductDesc
- ProductClass
- ProductSpec
- LineID
- BundleID

Assuming that the above steps went well, you should have the following configuration.



7. Now we need to inform OSM which of the above properties will hold the 'Product Spec' and which will be used to name the Order Item. Use the drop-downs on the lower part of the window to select the following values.

8. Now for each of the '**Order Item Properties**' you need to enter the XQuery statement that will extract the required value from the XML order data.

For 'LineID' the following command needs to be used.

```
declare namespace osm7dd="http://www.example.org/osm7Order1";
fn:normalize-space(osm7dd:lineID/text())
```

** **Note:** The Namespace declared is the Namespace used in the XML data.

** **Note:** The XQuery function 'fn:normalize-space' removes leading and training spaces and ensures only once space is left between words.

** **Note:** Finally, '`osm7dd:lineID/text()`' selects the value held in the XML file between the 'lineID' tags, for example if the file contained:

```
<osm7dd:lineID>03</osm7dd:lineID>
```

The value '**03**' would be selected.

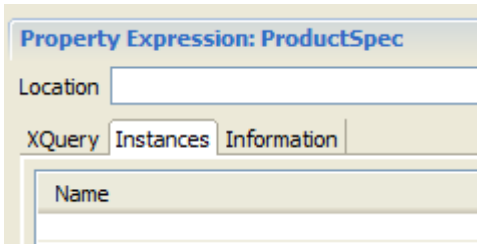
9. Using the information contained in the input order '**OSM7-DD-Lab2.xml**', you can complete the XQuery entries for all the other property values with the exception of 'ProductSpec'. If you are unsure what to look for, here are the entries.

```
fn:normalize-space(osm7dd:lineID/text())
```

```
fn:normalize-space(osm7dd:BundleID/text())
fn:normalize-space(osm7dd:class/text())
fn:normalize-space(osm7dd:productCode/text())
fn:normalize-space(osm7dd:productDesc/text())
fn:normalize-space(osm7dd:productName/text())
```

Producing the value for '**ProductSpec**' is slightly more complicated as only the '**Class**' exists in the incomming order data. What we need to do is create a lookup to the ProductClassMapping file to find what the product spec name is for the class in the order. To do this we use a standard function of OSM called a '**Data Instance**'.

9. Select the '**ProductSpec**' property and go to the 'Instances' tab.



10. Click 'Add' and you should see a new instance appear called '**DataInstance**'. Once created,right-click on the instance and rename it to **ClassMappingFileDataInstance**.

11. Select the new '**ClassMappingFileDataInstance**' data instance and click the 'Properties' button.

    - At the bottom of the screen in the '**Properties Field**' use the 'Select' button to point to the 'Data Provider' you previously created called '**ProductClassMappingFile**'.
    - Click the 'Save' button and you should see the file reference in the greyed out 'Expression' box.

This completes the creation of the '**Data Instance**', we now need to add an XQuery statement to query the instance for the product spec.

12. Under the '**productSpec**' property, go back to the 'XQuery' tab and select the 'Edit' button. Then add or paste the following statement.

```
declare namespace osm7dd="http://www.example.org/osm7Order1";
      let $productClassMap := vf:instance('ClassMappingFileDataInstance')
      let $productClassName  :=   fn:normalize-space(osm7dd:class/text())
      return
            fn:normalize-space($productClassMap/productClass[fn:lower-
case(@name)=fn:lower-case($productClassName)]/productSpec/text())
```

** **Note:** you will get a warning when you paste the XQuery command into the edit window.

```
Error 1 of 1: line 2, column 25: {err}FONS0003: "vf": prefix not defined in
static context
```

You can ignore this as the 'vf' is and internal OSM 'View Framework' functions and therefore does not need defining.

** **Note:** You have have an error in design studio – this can be ignored for now.

```
Invalid Order Template - The Order Item Specification OrderItemSpecification is
not associated with an Order.  Please ensure that the following relationship path
```

OSM Orchestration Exercise

Lab 8-1 - Order Item Specification

exists Order --> Orchestration Process --> Orchestration Sequence --> Order Item Specification.

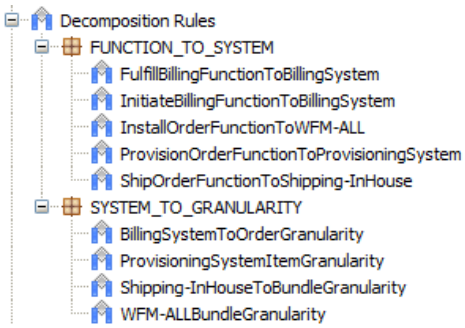
XQuery Explanation

XQuery Code	Description
`declare namespace osm7dd="http://www.example.org/osm7Order1";`	This defines the namespace in the incoming XML order.
`let $productClassMap := vf:instance('ClassMappingFileDataInstance')`	This creates a variable called 'productClassMap' containing the whole of the XML structure in the Data Instance i.e. the contents of the class to product spec mapping file. In this case the XML is: `<productClass name="MOBILE" cartridge="L3-DD-XXX"><productSpec>Mobile.Product.Spec</productSpec></productClass>`
`let $productClassName := fn:normalize-space(ord:class/text()) return`	Creates a variable called 'productClassname' containing the '**Class**' value in the incoming order. The function 'normalize-space' returns the argument string with the leading, trailing, and repeating white spaces stripped.
`return`	Indicates that the following line will define the return values to the calling routine.
`fn:normalize-space($productClassMap/productClass[fn:lower-case(@name)=fn:lower-case($productClassName)]/productSpec/text())`	This code '$productClassMap/productClass[fn:lower-case(@name' pulls out the attribute 'name' and converts it to lower case. In the case of the above XML mapping file this would be 'mobile' This is then compared to the product class extracted from the order using '=fn:lower-case($productClassName)' If a match is found then the value of 'productSpec' is returned by 'productSpec/text()

You have created your '**Order item Specification**' ☺

13. For each of your '**Decomposition Rules**' that are marked with a **red x**

- Edit the entity
- Use the select button next to the '**Order** Item' field to pick your '**Order Item Specification**' from the list.
- Once you have completed all rules goto 'File / Save All' on the menu bar to save all changes.
- All errors on the decomposition rules should be resolved



Once you have done all the rules you have completed this section and earned a well deserved rest.

We have now completed **Lab** 8.

END OF LAB