Oracle BPEL Tutorial

Installation

1. Install **VirtualBox**

https://www.virtualbox.org/wiki/Downloads

2. Install **Oracle VM VirtualBox Extension Pack**

http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html

Important: Take into consideration that the version of virtual box has to be compatible with the version of the extension pack

3. Download and install **Pre-built Virtual Machine for SOA Suite and BPM Suite**

http://www.oracle.com/technetwork/middleware/soasuite/learnmore/vmsoa-172279.html

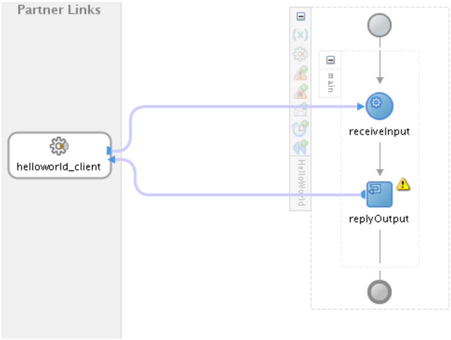
Hello World

In this section of the practice, you have to construct a process to receive as input a language and a name, and then display a greeting in the specified language. You will need Internet connection for the second part of the practice, then before starting, make sure your virtual machine is well configured. In Virtualbox, under Settings/Network/Advanced select for Adapter type: Intel PRO / 1000 MT Desktop.

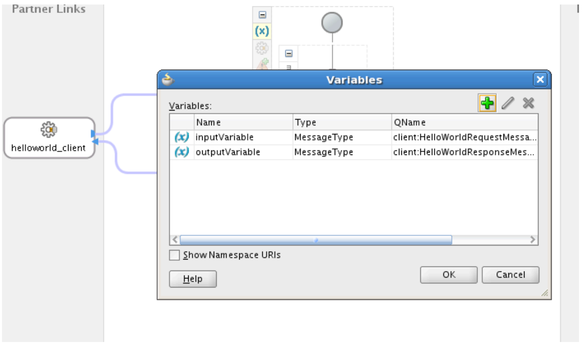
1. Open JDeveloper and create a new SOA project, specify a name and, in the Composite Template, select Empty Composite

2. Drag a BPEL Process component from the Component Palette
   a. Change the name of the process
   b. The template shall be a Synchronous BPEL Process

3. Double click in the created component to see the process. There are already a start and an end event, in addition to a receive input and a reply input



4. If you want to see the existing variables then click the '(x)' symbol



Variables:

	Name	Type	QName
(x)	inputVariable	MessageType	client:HelloWorldRequestMessa...
(x)	outputVariable	MessageType	client:HelloWorldResponseMes...

□ Show Namespace URIs

5. Open the file under your_project_name/SOA Content/xsd/file.xsd. Add a new element in the section of the input variable of type **string**. For easier understanding, rename the two variables; one has to be **language** and the other **name**

6. Now we are going to define the decision points. Drag a Switch component, click in the condition part. Add the following code:

   xp20:compare(string(bps:getVariableData('inputVariable','payload','/client:process/client:language')),'spanish') = 0

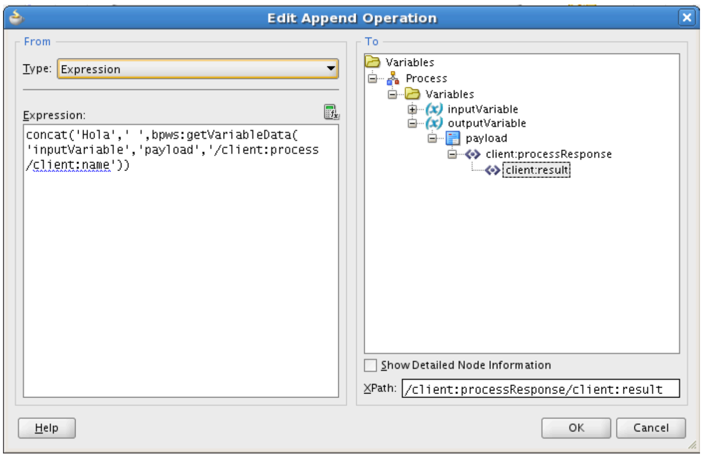
7. In the branch of the condition, drag an **Assign** activity from the **Component Palette. Assign** activity is used to copy values from one variable to another variable

8. In the **Assign**, we are going to define the action after detecting that the language entered was 'spanish'. Right click in the Assign activity and select the option Edit
   a. In the tab Assign Operation, select the green plus symbol and select Append Operation
   b. In the left side part (section for From) select Type Expression. And add the following code.

      code:concat('Hola,' ',bpws:getVariableData('inputVariable','payload','/client:process/client:name'))

      Note: It is possible when you deploy the application some errors appear because of the name of some variables. In that case, replace "*bpws:getVariableData('inputVariable','payload','/client:process/client:name')*" with the correct value.

   c. In the right side (section for To), select client:result under the outputVariable
   d. And select OK. With this, we append the word 'Hola' before the name entered, and assign the new string to outputVariable



Type: Expression

Expression:

```
concat('Hola',' ',bpws:getVariableData(
'inputVariable','payload','/client:process
/client:name'))
```

XPath: /client:processResponse/client:result

9. In the other branch of the switch ('otherwise'), add another Switch and define greetings in two more languages
10. Validate your diagram and select deploy. If you get no errors, then go to the console of the server and test your application

To test your application

1. In menu **View**, select **Application Server Navigation**

2. In **localhost-wls/SOA/AdminServer/default** locate your application and, in the right click menu, Test Web Service

3. Fill in the data and click in **Send Request**

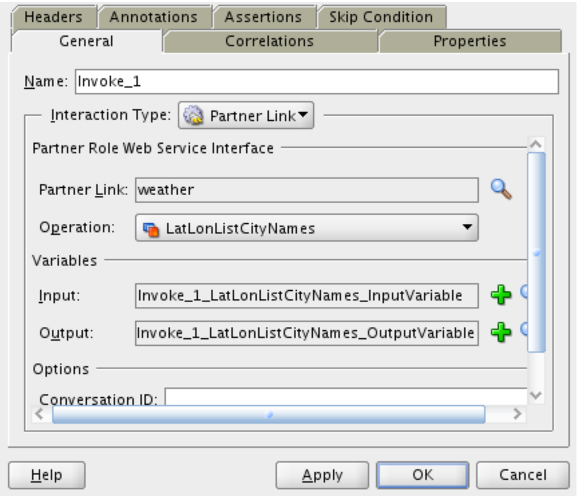
Calling a web service

1. Repeat the steps from 1 to 3 from the practice above

2. Drag to the Partner Links section a Web Service (it is under **BPEL Services** in the Component Palette)

3. In the URL add 'http://graphical.weather.gov/xml/DWMLgen/wsdl/ndfdXML.wsdl'. You can read about the available methods and see some examples of the responses in http://graphical.weather.gov/xml

4. In your process, add an Invoke activity after receiveInput. It will call a method in the web service we are making reference to. In this example, we will call the LatLonListCityNames operation. In the section variables, add a new Global variable with the plus sign in Input and Output



Name: Invoke_1

Interaction Type: Partner Link

Partner Link: weather

Operation: LatLonListCityNames

Input: Invoke_1_LatLonListCityNames_InputVariable

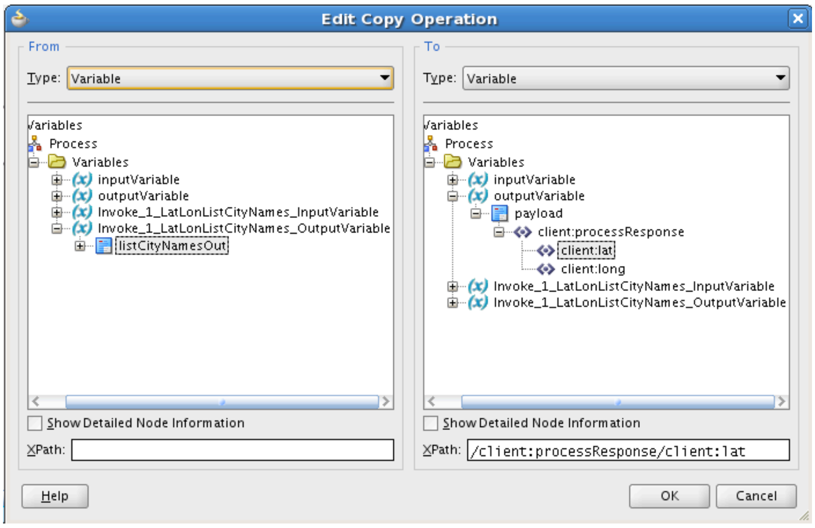
Output: Invoke_1_LatLonListCityNames_OutputVariable

Conversation ID:

5. As the selected operation receives an integer, it is necessary to transform the received variable in the Input with an Assign activity. Add the following expression in the from section and link it to the created global input variable created for the Invoke:

   number(bps:getVariableData('inputVariable','payload','/client:process/client:input'))

6. After **Invoke** element, place another Assign and put the value from the output variable in the **Invoke** to outputVariable


Edit Copy Operation
From
To
Show Detailed Node Information
Show Detailed Node Information
Help
OK
Cancel

From — Type: Variable

```
Variables
Process
  Variables
    inputVariable
    outputVariable
    Invoke_1_LatLonListCityNames_InputVariable
    Invoke_1_LatLonListCityNames_OutputVariable
      listCityNamesOut
```

XPath:

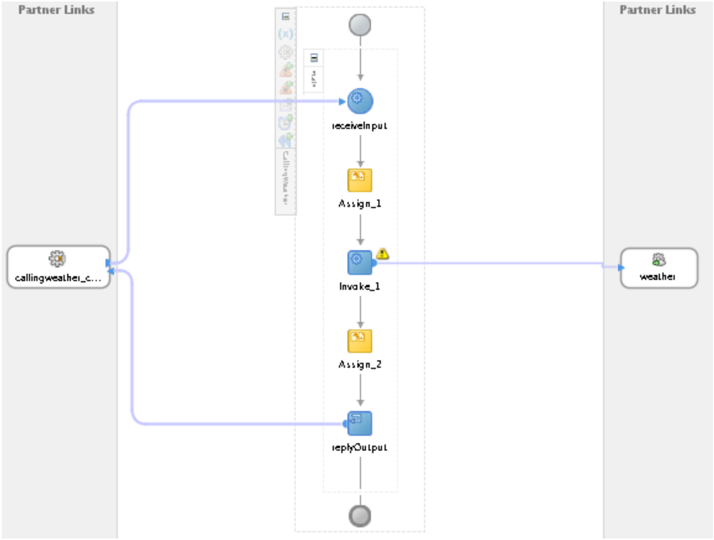
To — Type: Variable

```
Variables
Process
  Variables
    inputVariable
    outputVariable
      payload
        client:processResponse
          client:lat
          client:long
    Invoke_1_LatLonListCityNames_InputVariable
    Invoke_1_LatLonListCityNames_OutputVariable
```

XPath: /client:processResponse/client:lat

7. Save and deploy your application

8. Try other methods and if the temperature is given in Fahrenheit scale, make the conversion to Celsius with your service deployed in the ats.



Other links

In the following link you can see an example how to connect and get data from a database
http://www.youtube.com/watch?v=8vM3bkbhsyA