

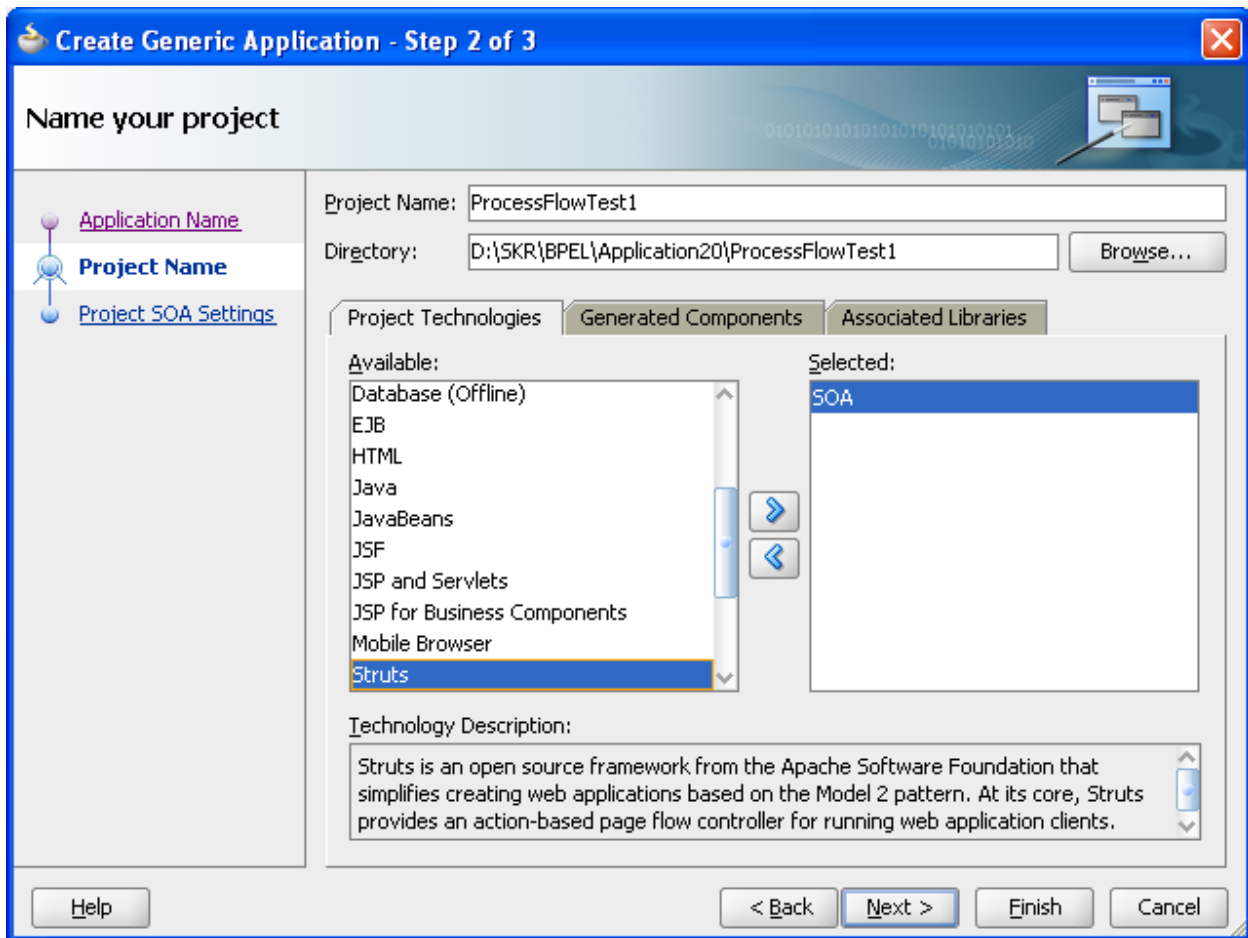
BPEL_FCJ_Dev Guide
Oracle FLEXCUBE Universal Banking
Release 14.2.0.0.0
[December] [2018]



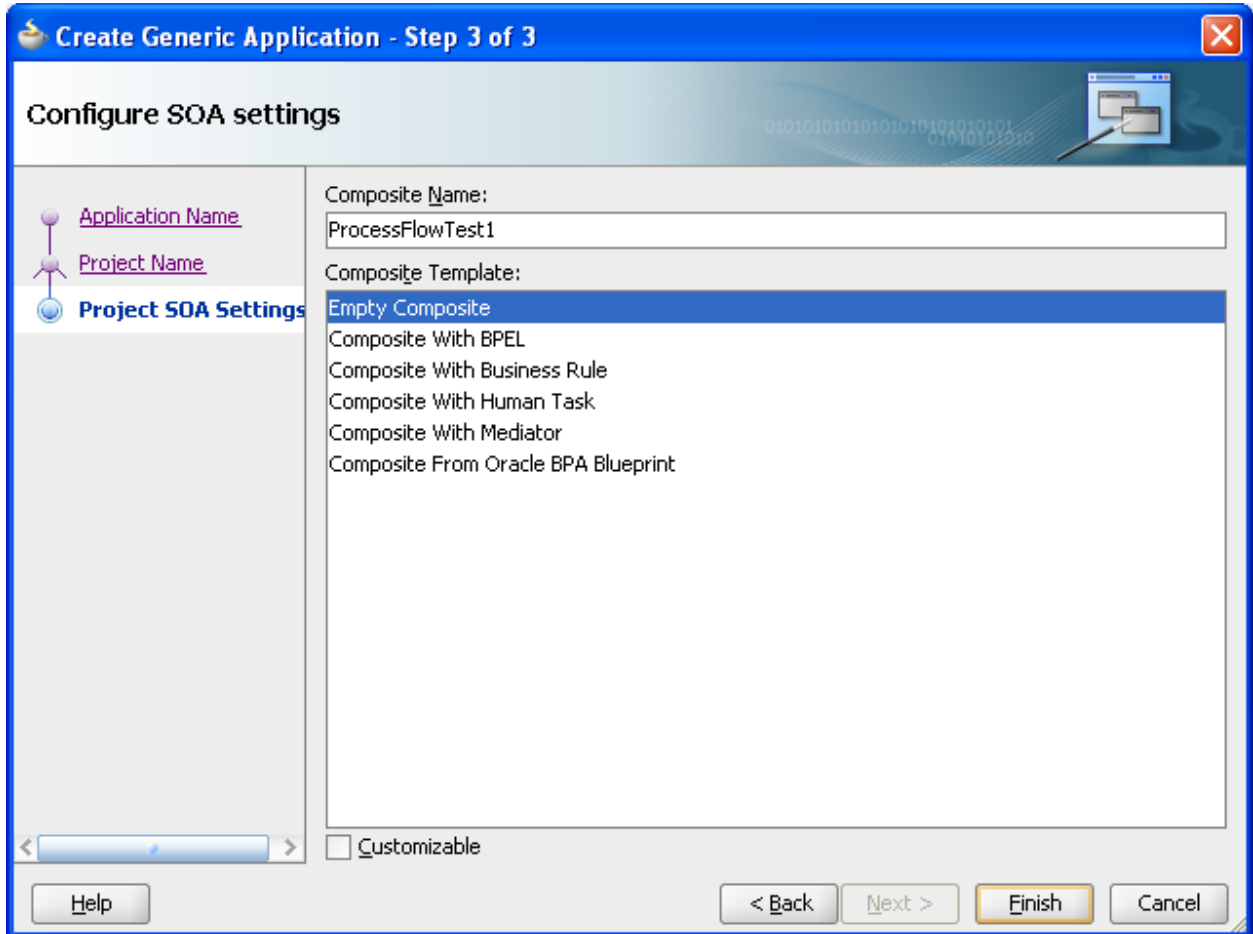
This Document is intended to explain as to how to develop simple process flow in FCUBS using BPEL, SOA Suite and JDeveloper

CREATING PROCESS FLOW

1. Create Generic application -> create SOA project.



2. Choose 'Empty Composite'



3. Name BPEL Process **same as the project name in Step -1**

Create BPEL Process

BPEL Process

A BPEL process is a service orchestration, used to describe/execute a business process (or large grained service), which is implemented as a stateful service.

Name:

Namespace:

Template:

Service Name:

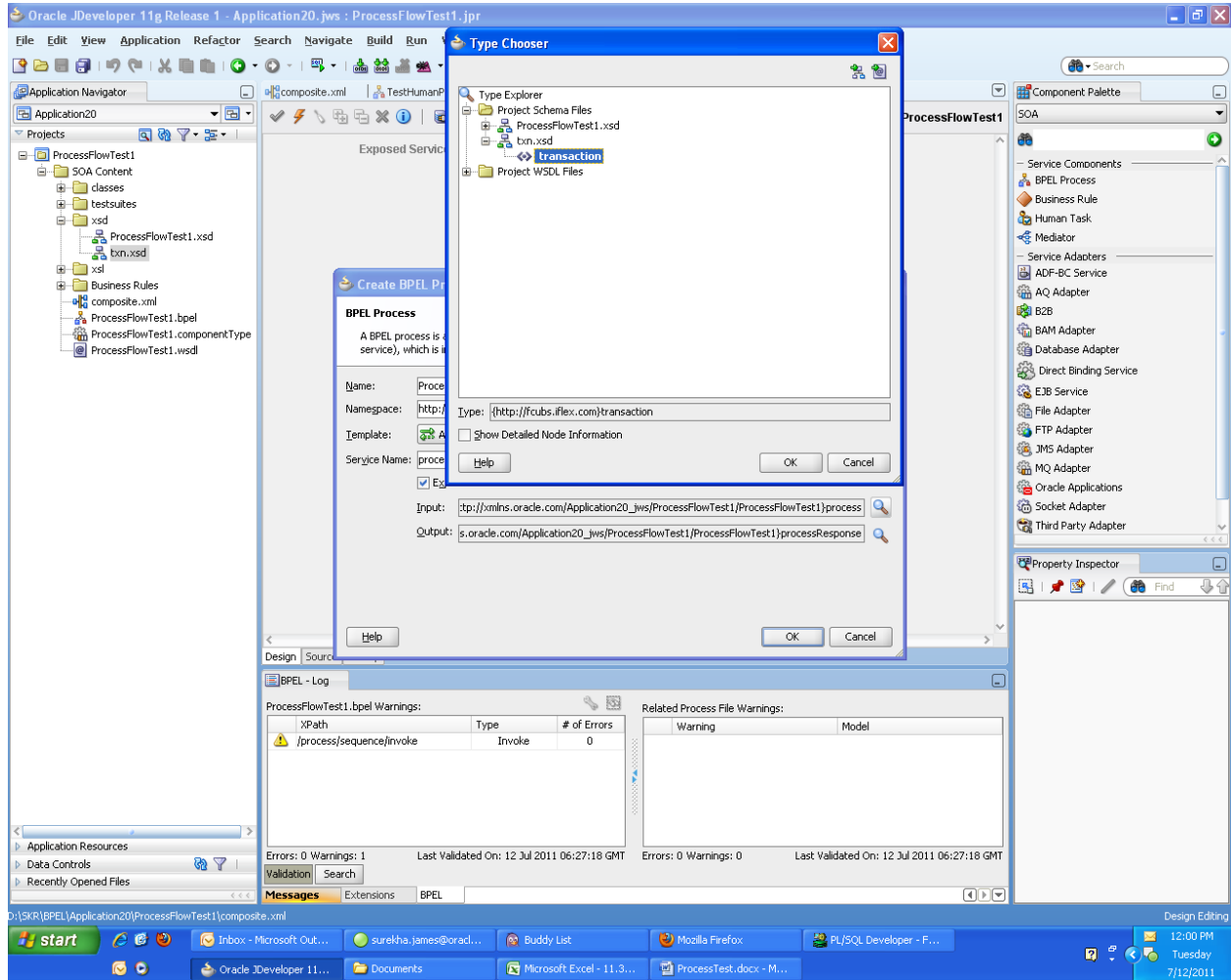
Expose as a SOAP service

Input:

Output:

Buttons: Help, OK, Cancel

4. Change the input/output of .xsd to type txn.xsd which is the standard followed for FCUBS



5. Click OK.

Create BPEL Process

BPEL Process

A BPEL process is a service orchestration, used to describe/execute a business process (or large grained service), which is implemented as a stateful service.

Name:

Namespace:

Template:

Service Name:

Expose as a SOAP service

Input:

Output:

Buttons: Help, OK, Cancel

6. Process is created.

The screenshot displays the Oracle JDeveloper 11g Release 1 interface for creating a BPEL process. The main workspace shows a diagram with two components: 'processflowte...' in the 'Exposed Services' section and 'ProcessFlow...' in the 'Components' section. The 'Component Palette' on the right lists various SOA components, including BPEL Process, Business Rule, Human Task, Mediator, and several Adapters. The 'Properties' window on the right shows the following details for the selected component:

Name:	ProcessFlowTest1
Revision:	1.0
Label:	2011-07-12_11-55-2
Home:	
Mode:	active
State:	on
Validate Schema:	Unspecified

The 'BPEL - Log' window at the bottom shows a table for 'Related Process File Warnings' with columns for XPath, Type, and # of Error. The status bar at the bottom indicates the current file is 'D:\SKR\BPEL\Application20\ProcessFlowTest1\composite.xml' in 'Design Editing' mode.

7. Open the .bpel

The screenshot displays the Oracle JDeveloper 11g IDE with a BPEL process diagram open. The main editor shows a process flow starting with a 'receiveInput' activity, followed by a 'callbackClient' activity. The interface includes a Project Explorer on the left, a Component Palette on the right, and a Log/Warnings panel at the bottom.

Log/Warnings Panel:

ProcessFlowTest1.bpel Warnings:			Related Process File Warnings:	
XPath	Type	# of Errors	Warning	Model
/process/sequence/invoke	Invoke	0		

Errors: 0 Warnings: 1 Last Validated On: 12 Jul 2011 06:27:18 GMT

Errors: 0 Warnings: 0 Last Validated On: 12 Jul 2011 06:27:18 GMT

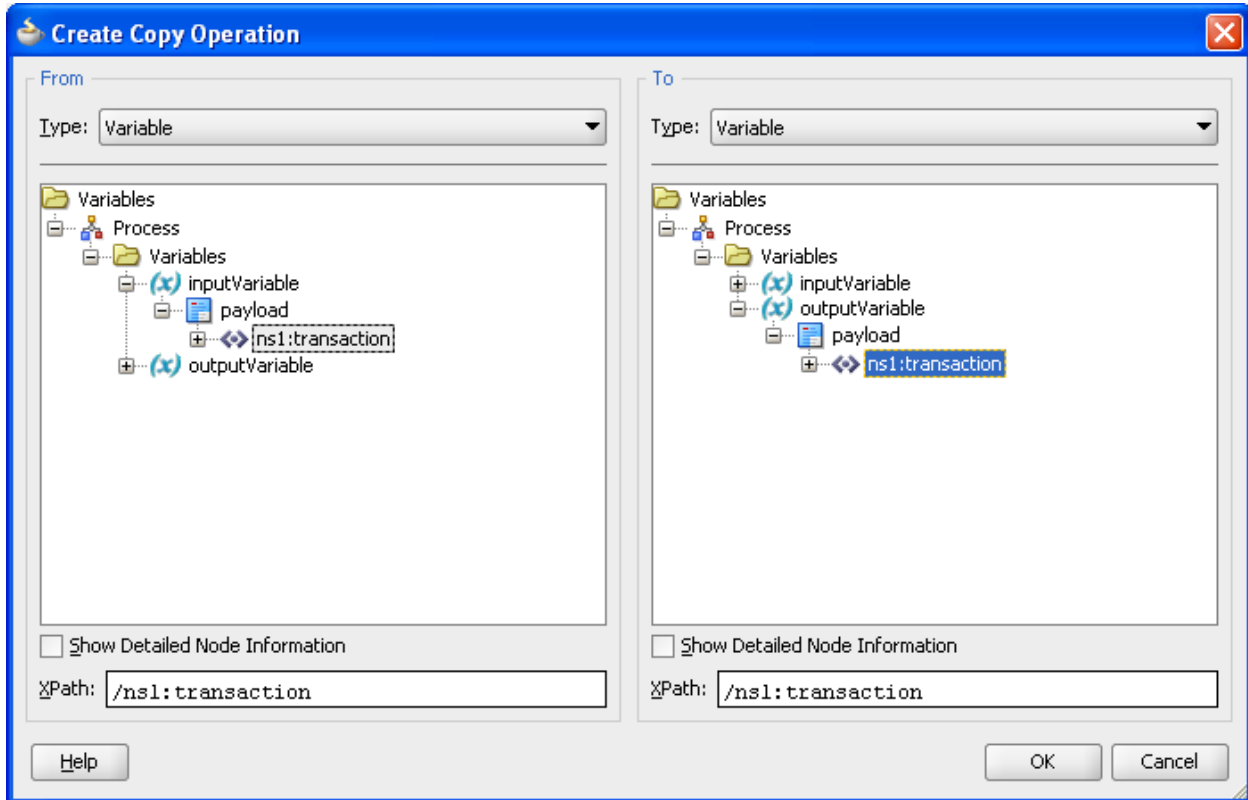
8. Drag and drop 'Assign' activity

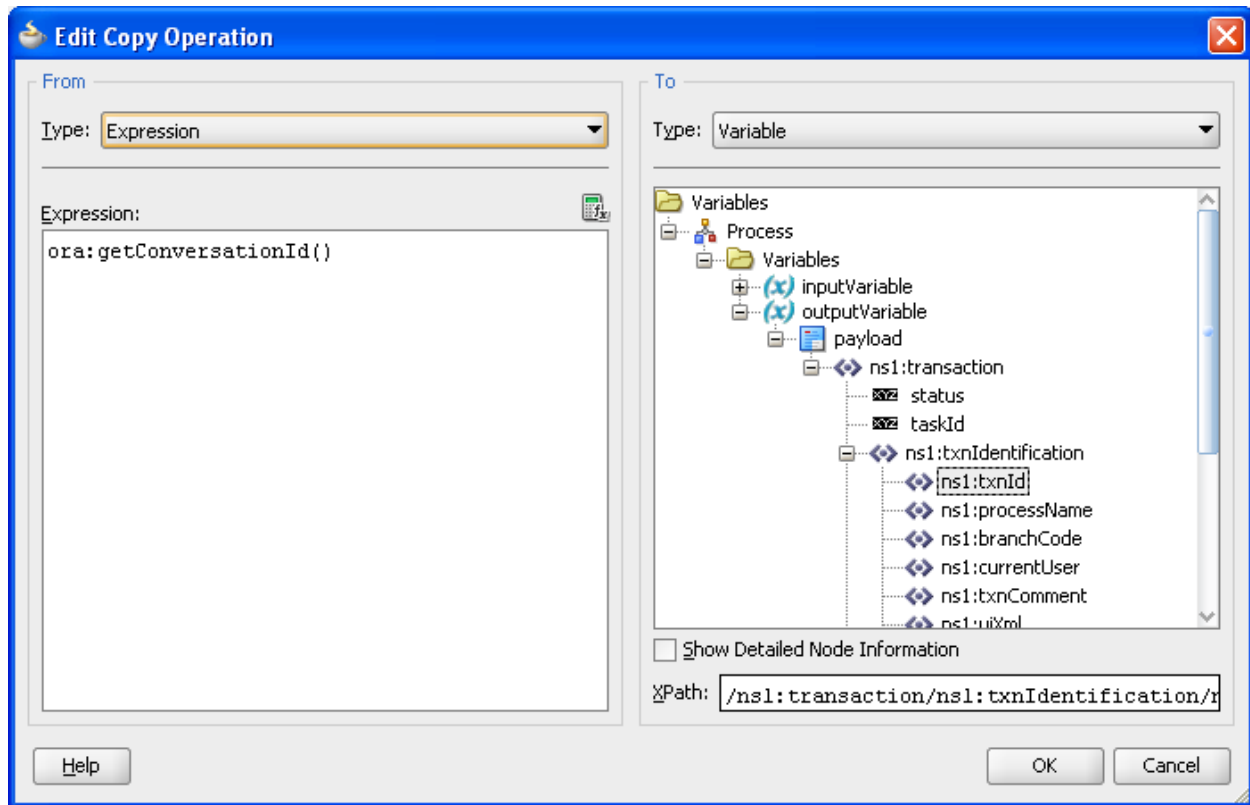
The screenshot displays the Oracle JDeveloper 11g IDE interface for editing a BPEL process. The main workspace shows a process flow starting with a 'receiveInput' activity, followed by an 'Assign_I' activity, and then a 'callbackClient' activity. A blue arrow indicates the 'Assign' activity being dragged from the Component Palette to the flow between 'receiveInput' and 'callbackClient'. The Component Palette on the right lists various BPEL activities, including 'Assign'. The bottom of the IDE features a 'BPPEL - Log' window with two tables: 'ProcessFlowTest1.bpel Warnings' and 'Related Process File Warnings'. The status bar at the bottom indicates 'Errors: 0 Warnings: 1' and 'Last Validated On: 12 Jul 2011 06:27:18 GMT'.

XPath	Type	# of Errors
/process/sequence/invoke	Invoke	0

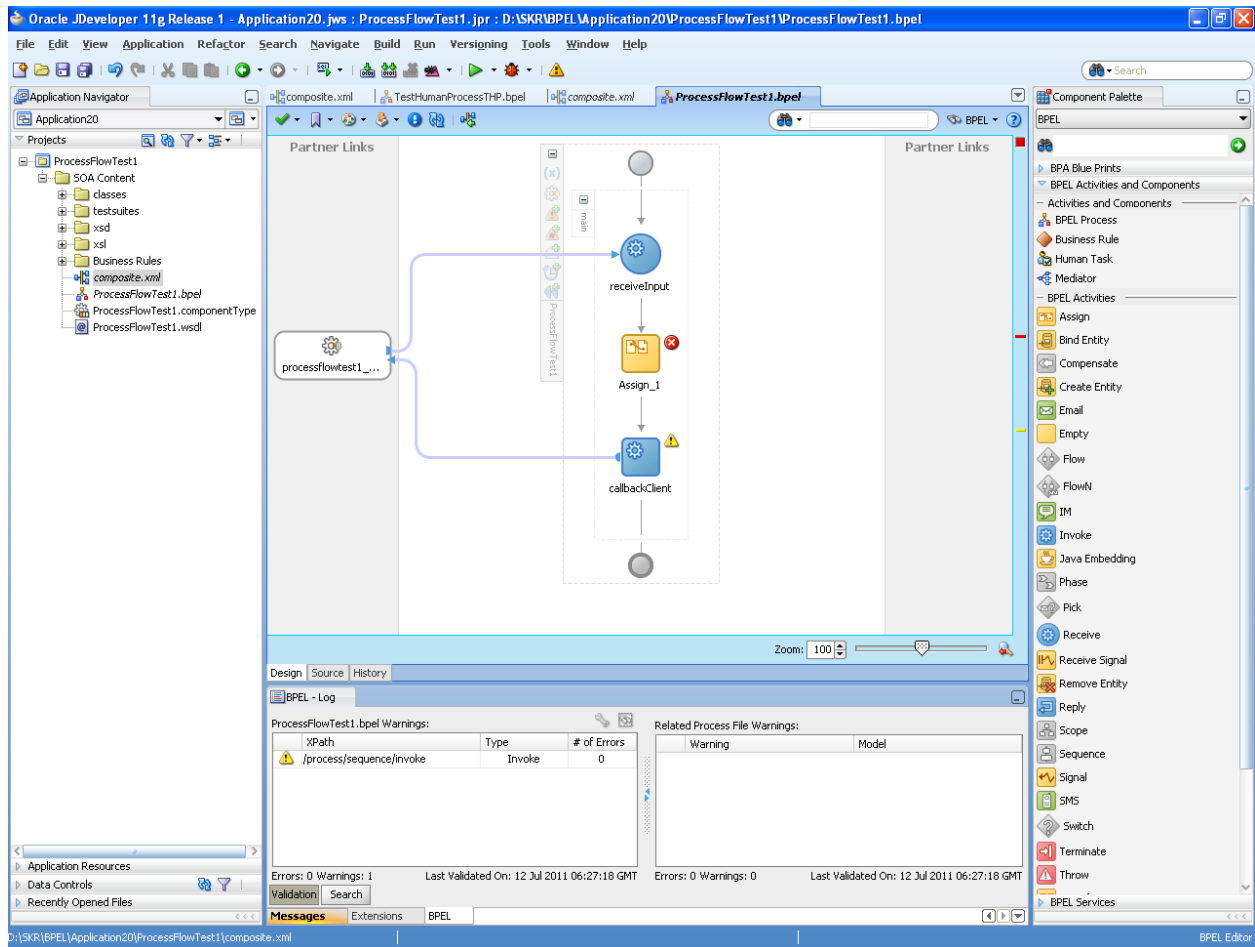
Warning	Model
---------	-------

9. Give the 'Copy Operation' in 'Assign' activity as follows.

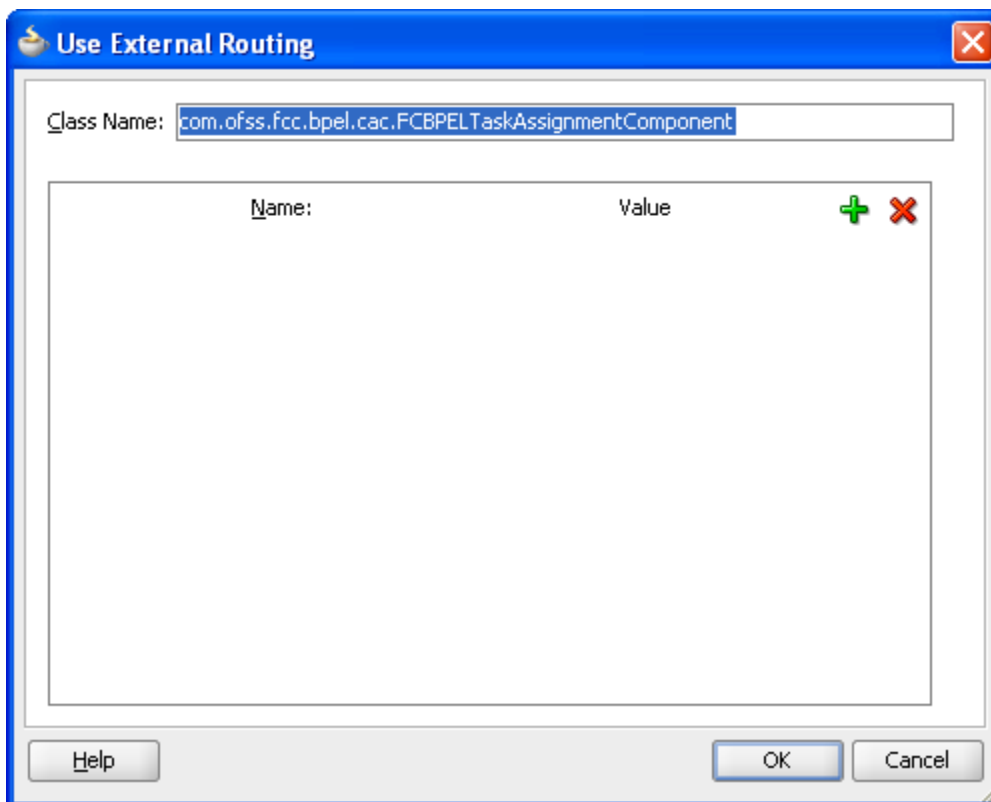
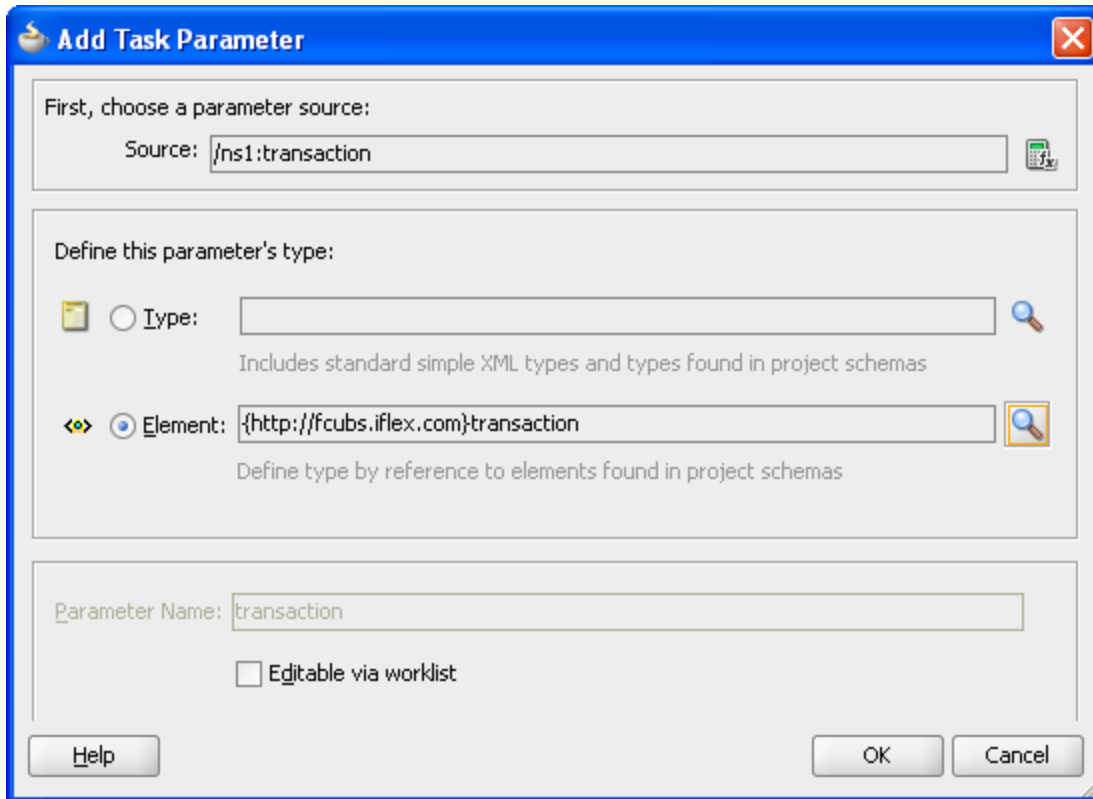




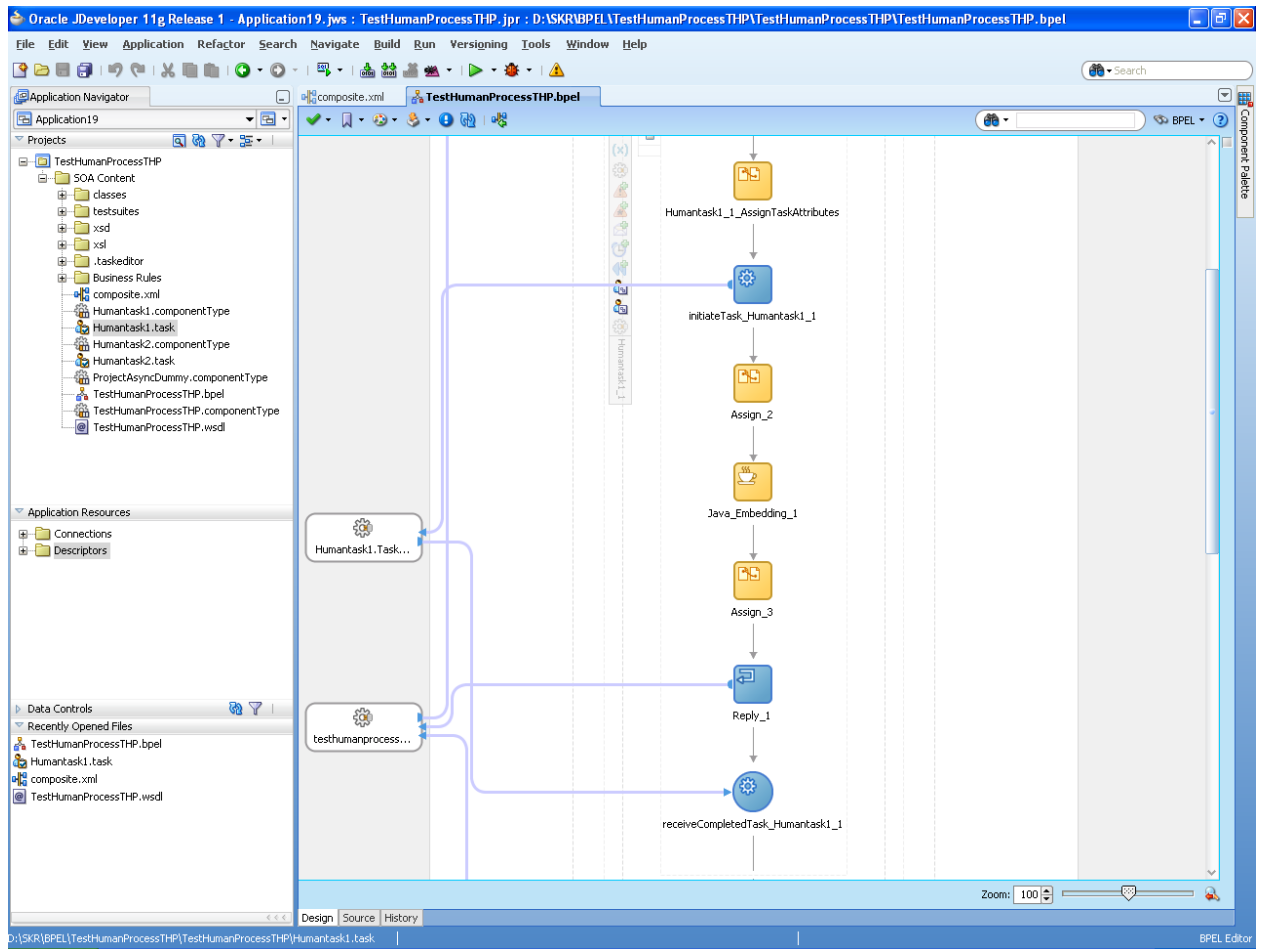
10. Drag and drop the Human task from the component Palette (BPEL activities and components) after the assign activity and navigate to .task from the Application Navigator on the left.



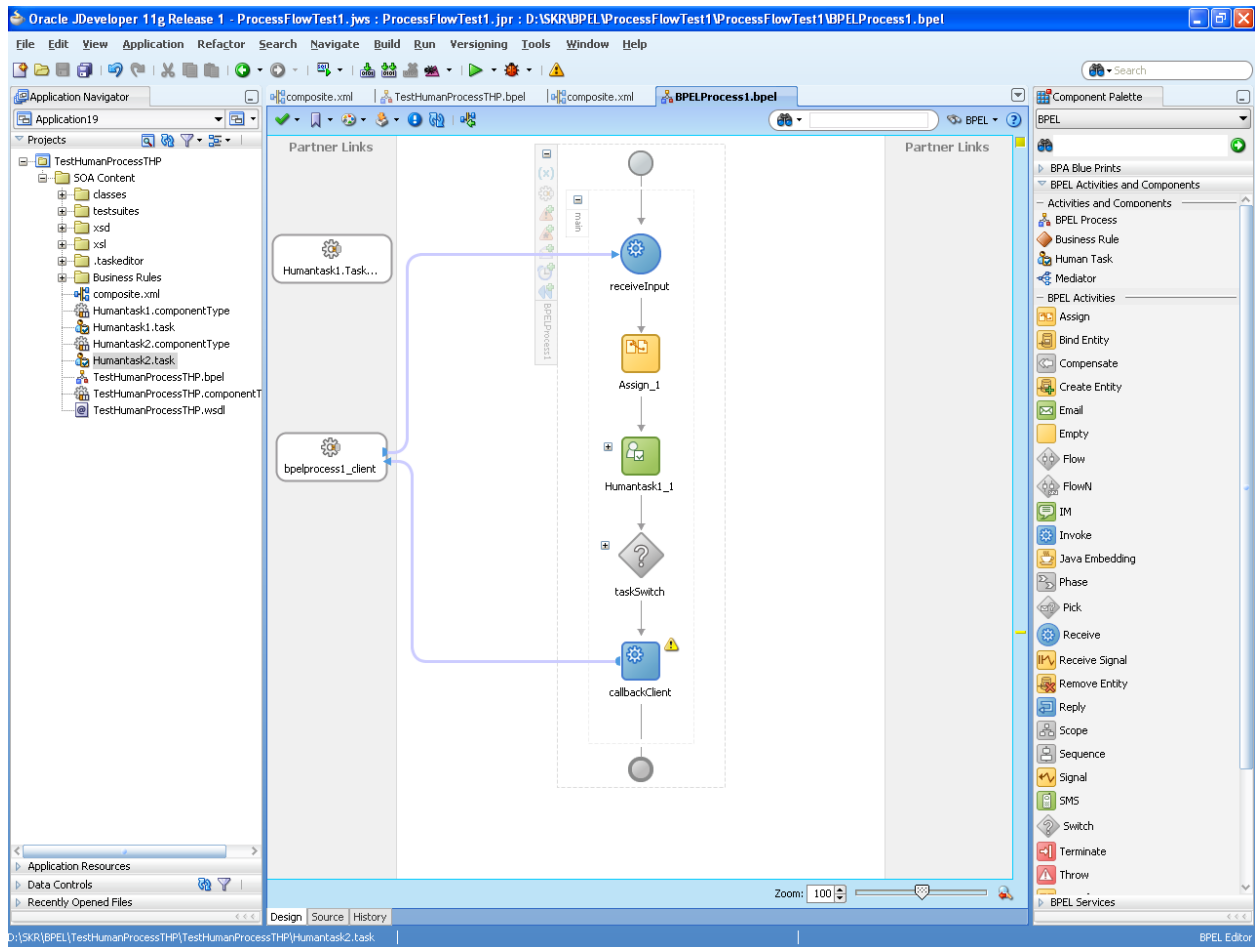
11. Give the 'source' same as output variable and 'Element' as transaction type in the Human task.



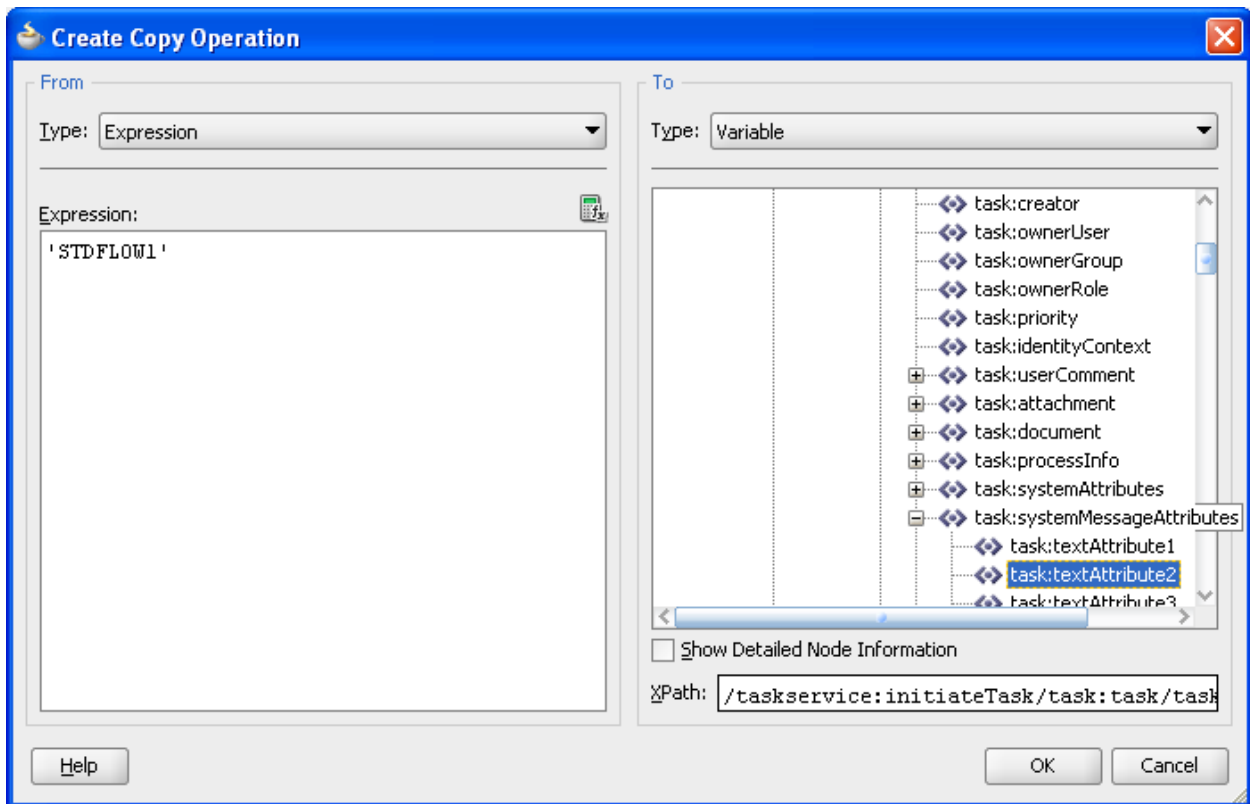
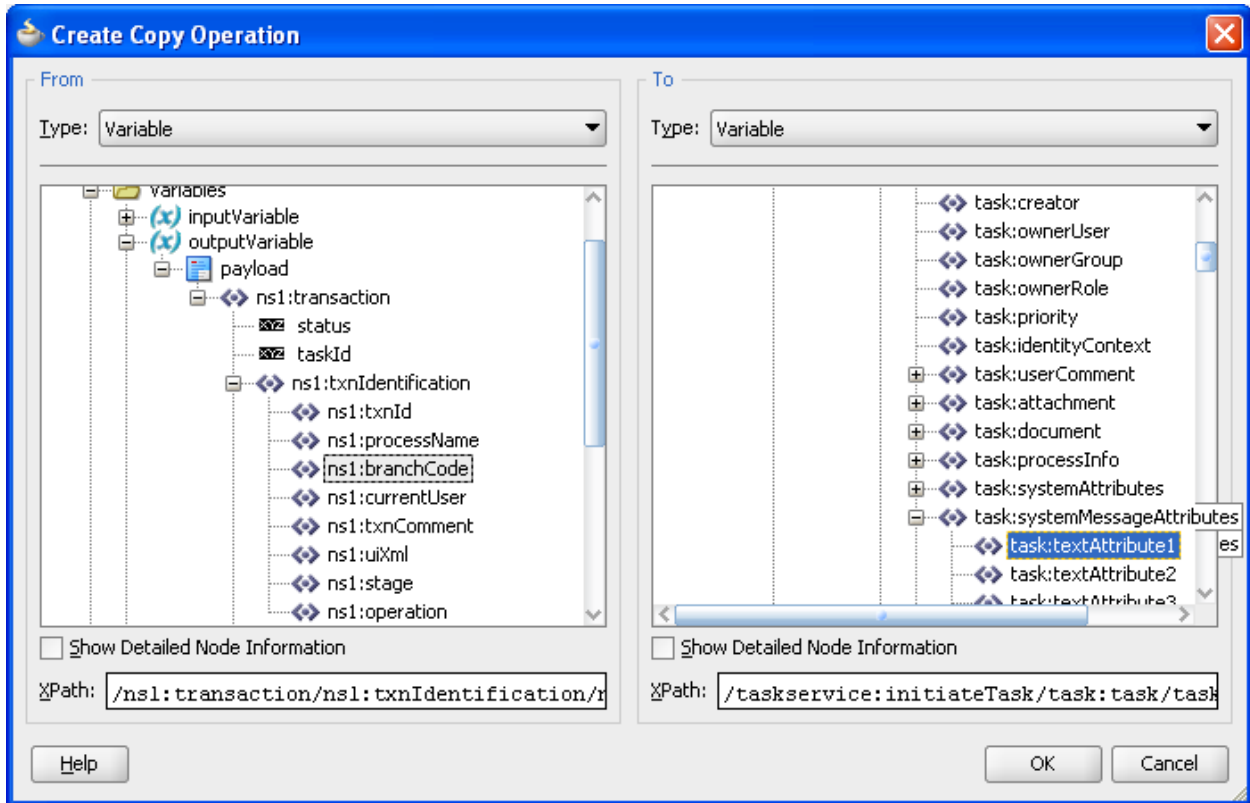
12. Expanded Humantask shown below.



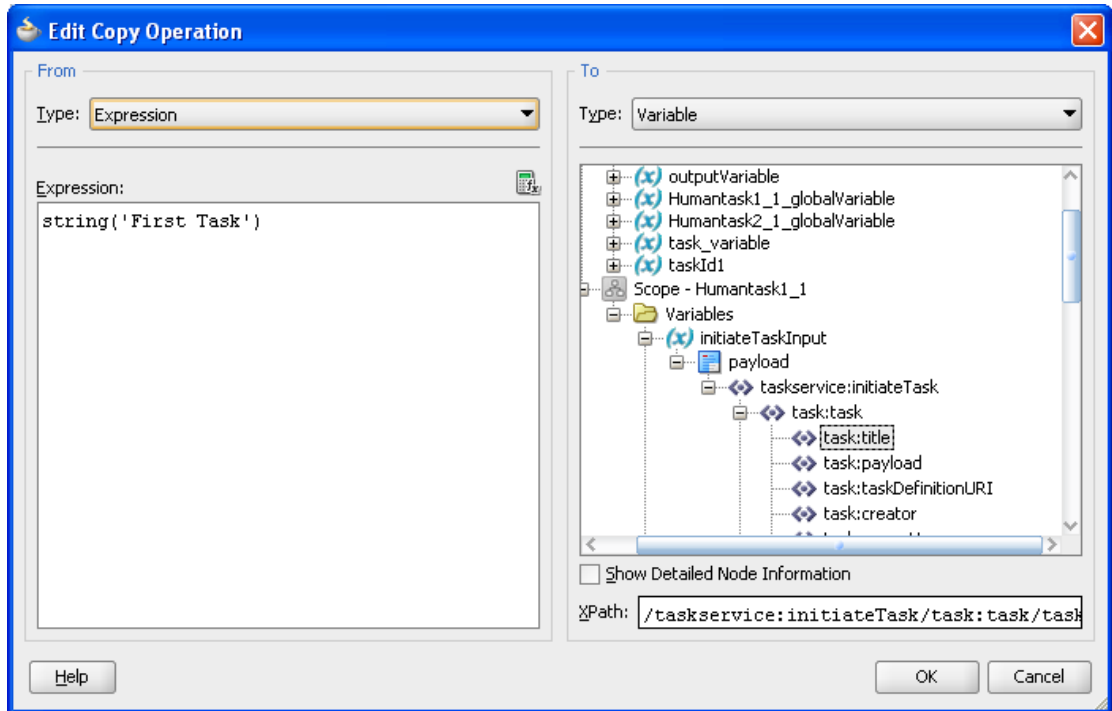
13. Human task is created.



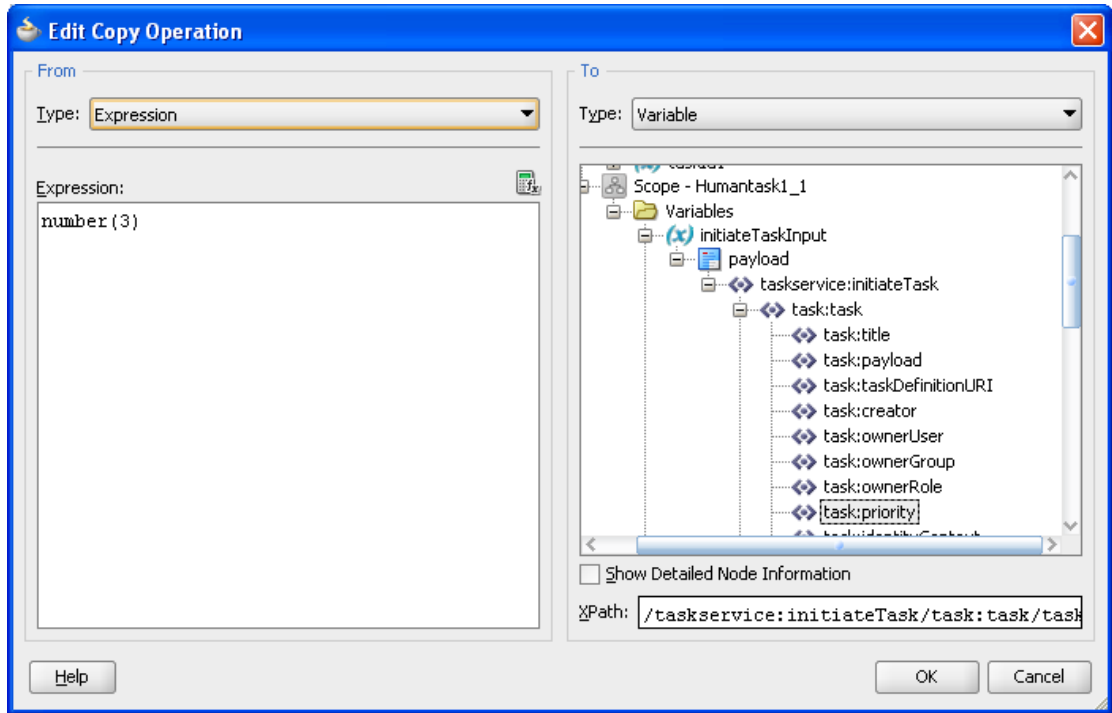
14. Create 'Copy Operation' for 'Assign' activity in Human task as follows.



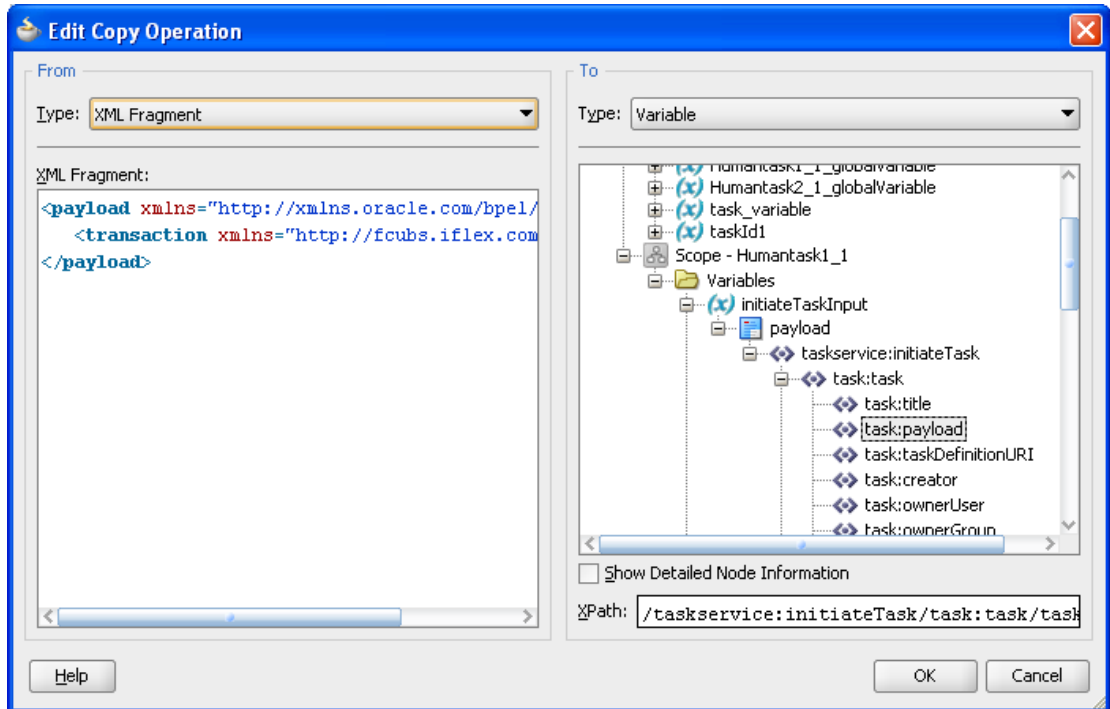
Note : Other automatic assignments which are defaulted on creation of Human Task are as follows-



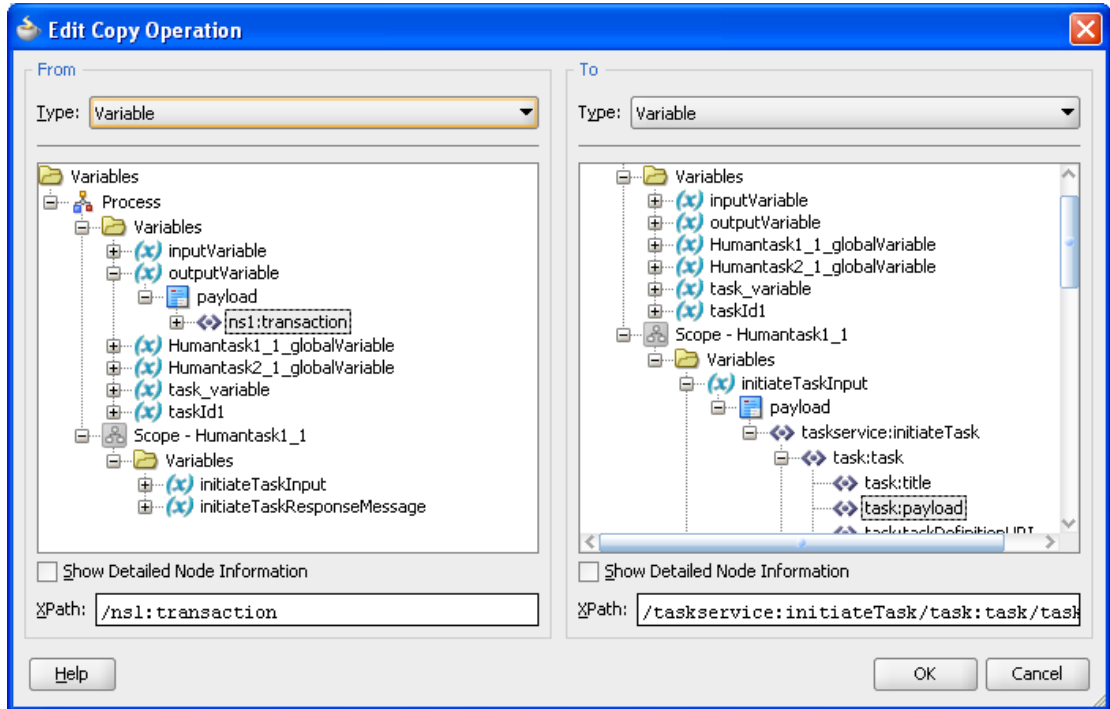
i.



ii.

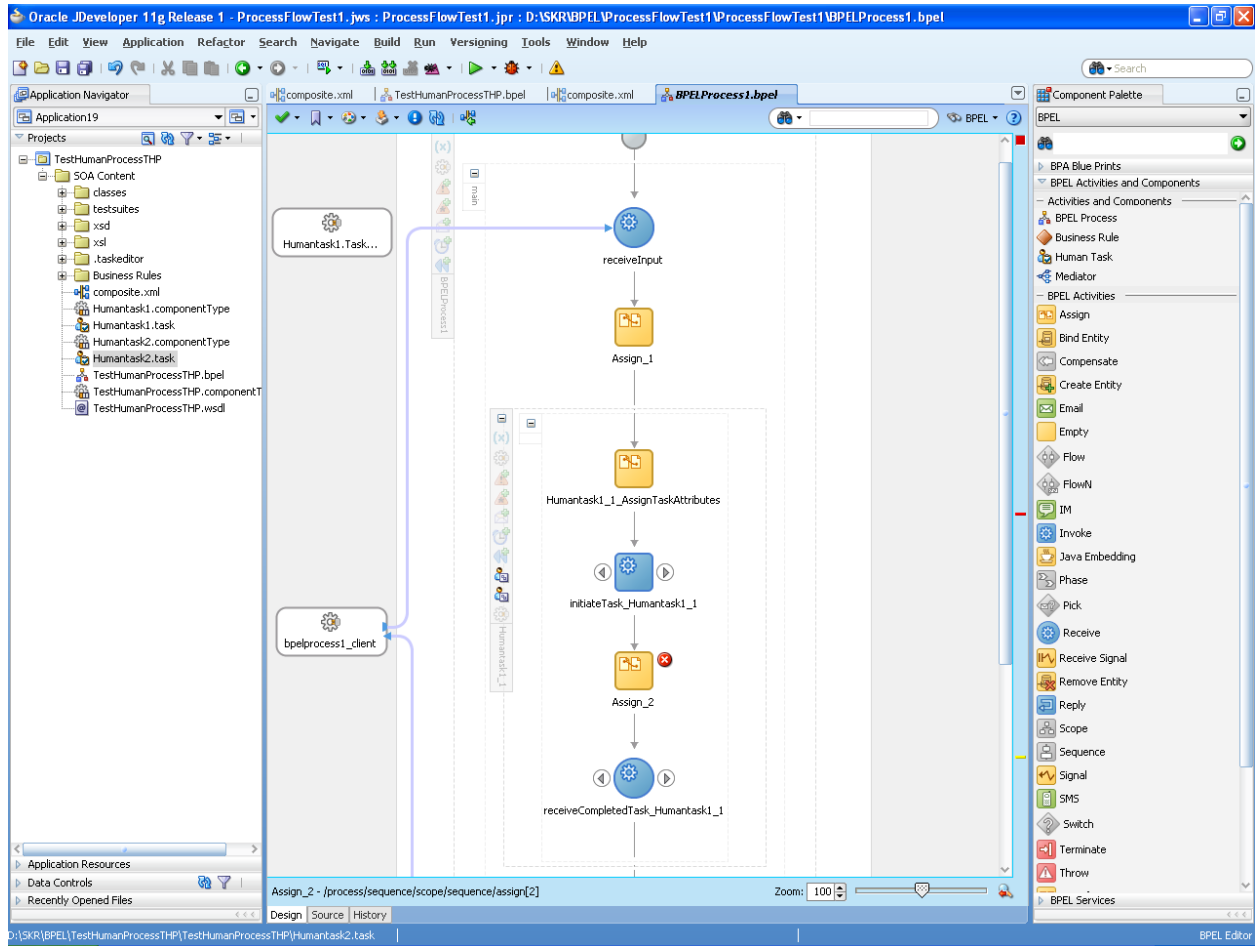


iii.

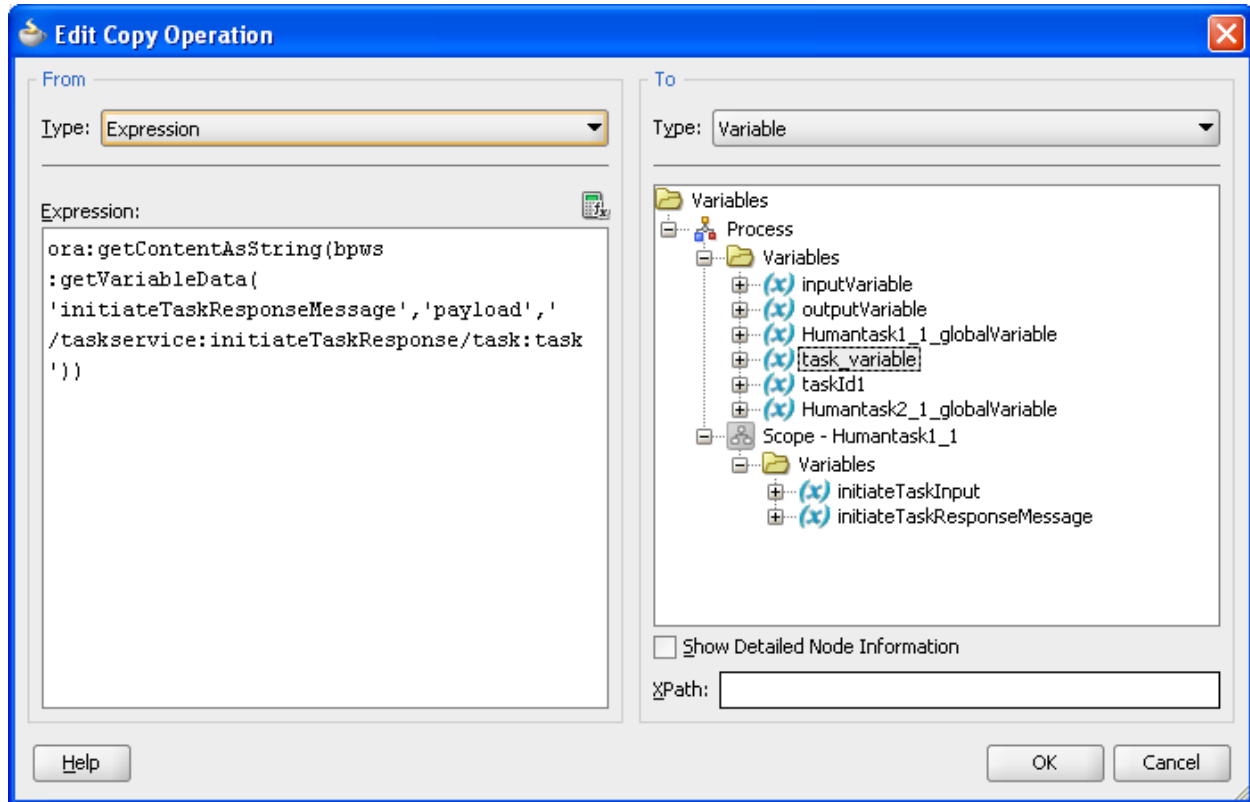


iv.

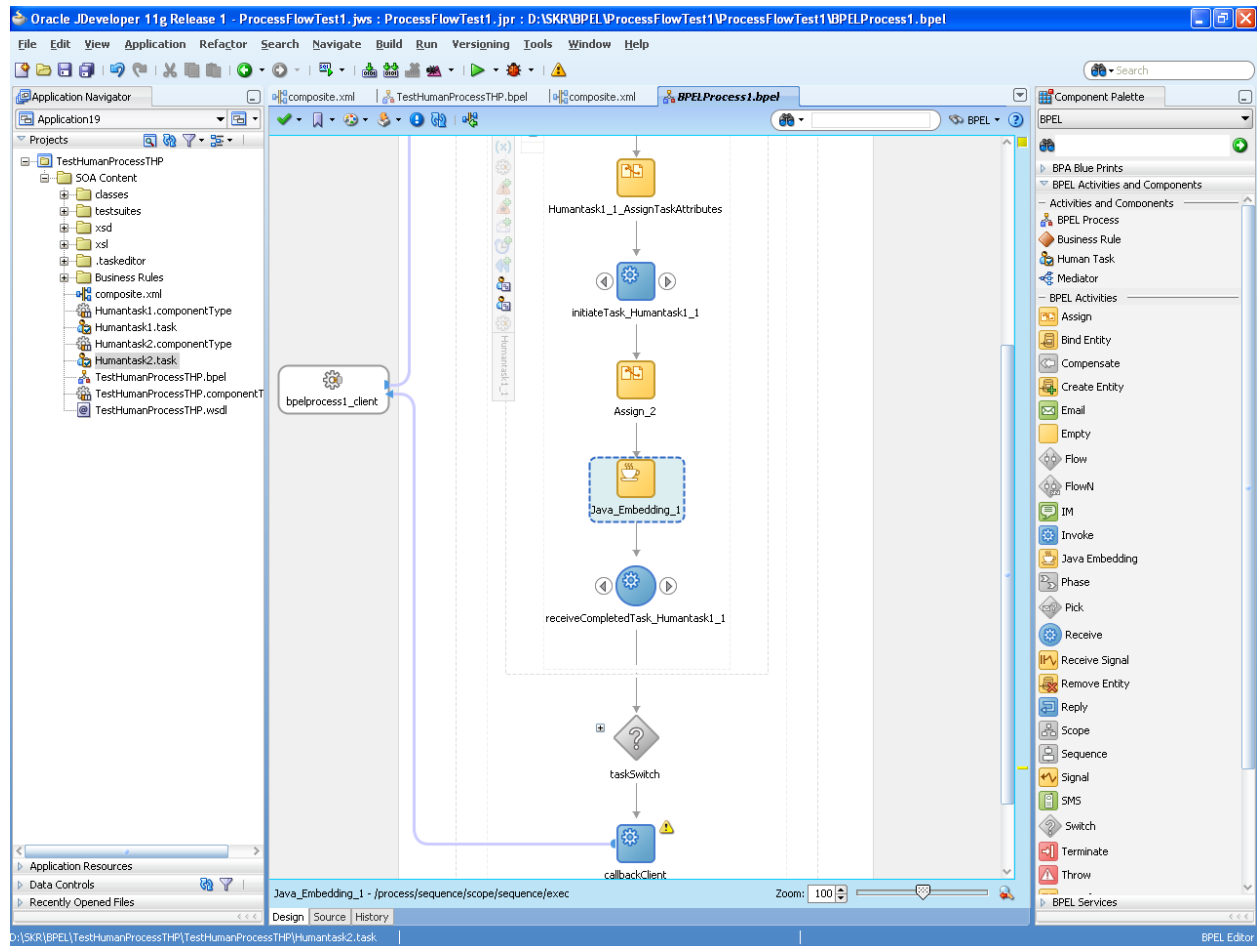
15. Put another 'Assign' activity in Human task



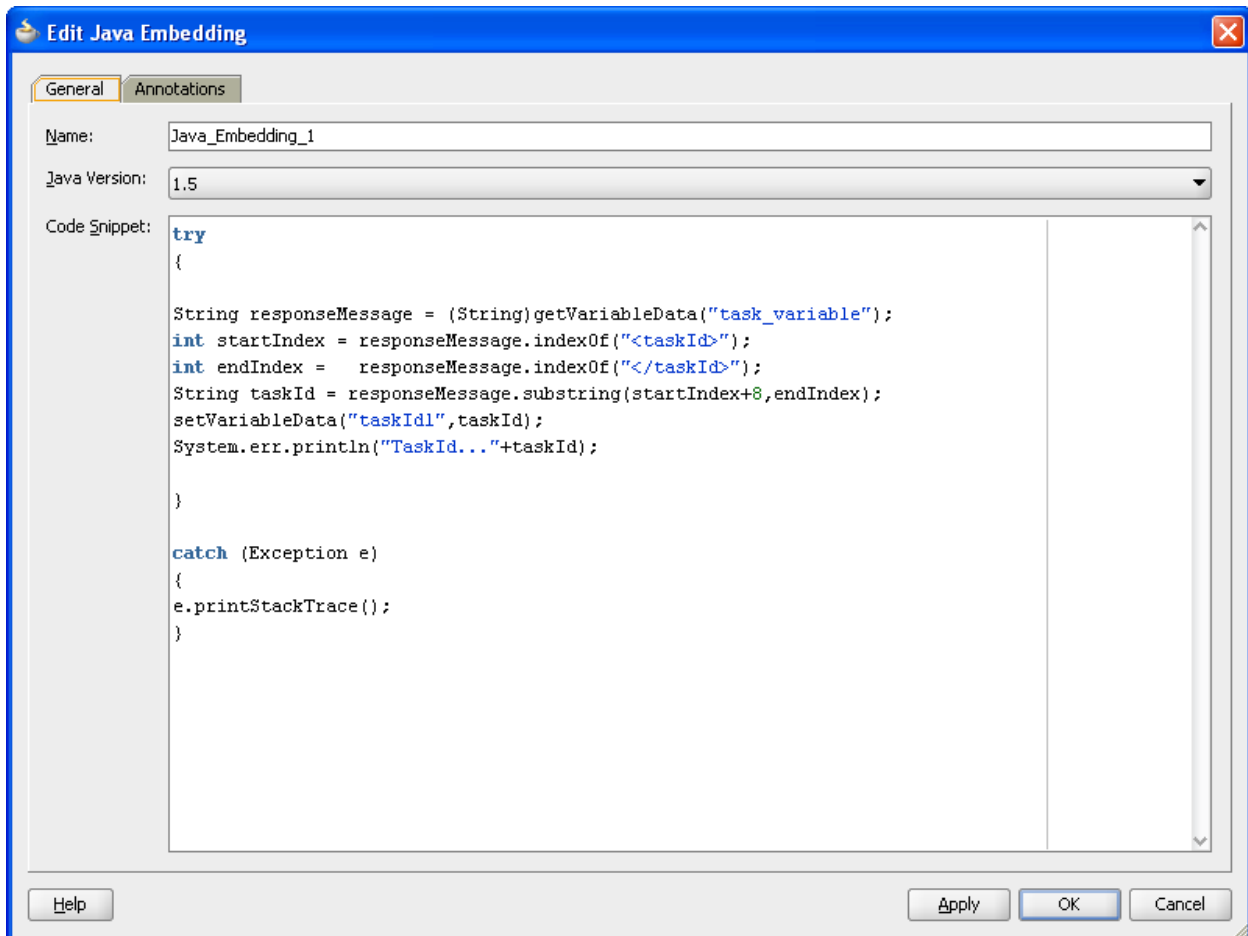
16. Give the 'Copy Operation' for 'assign' activity. Create a new variable by right clicking on 'variables' in Variables -> Process -> Variables



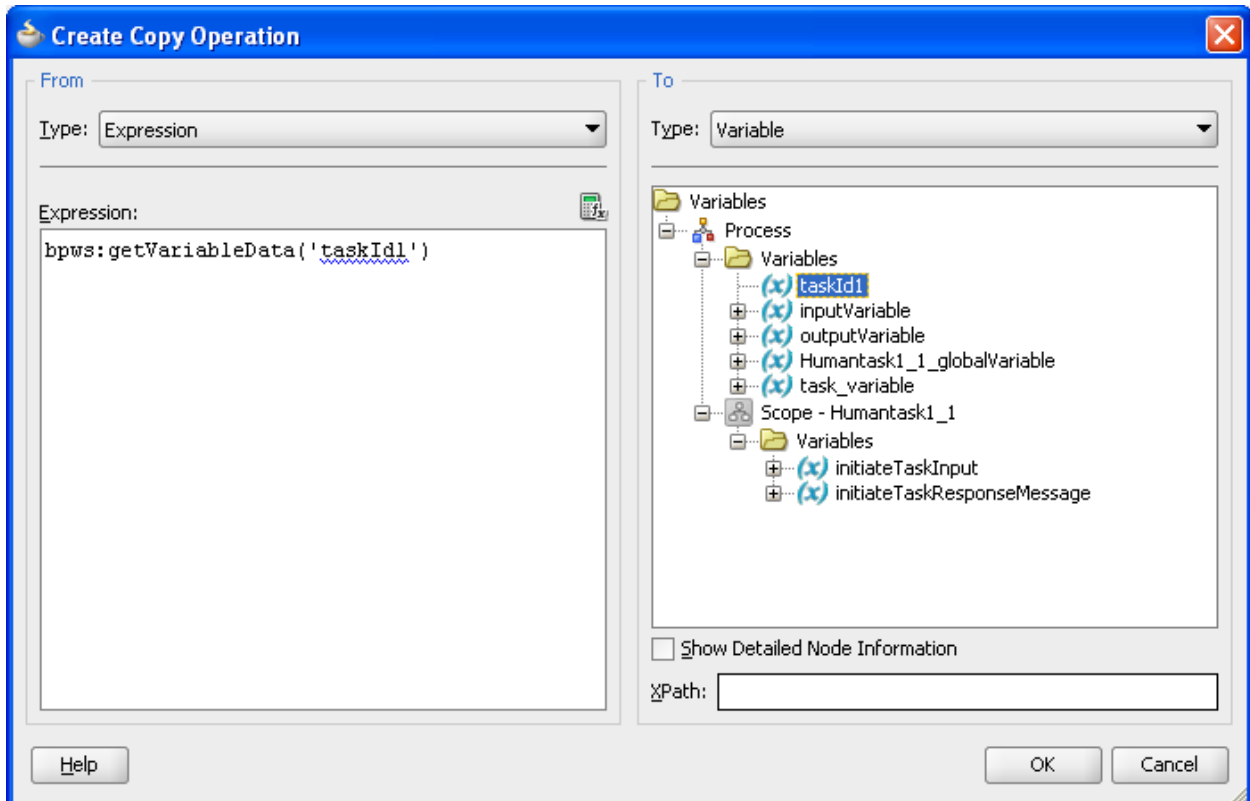
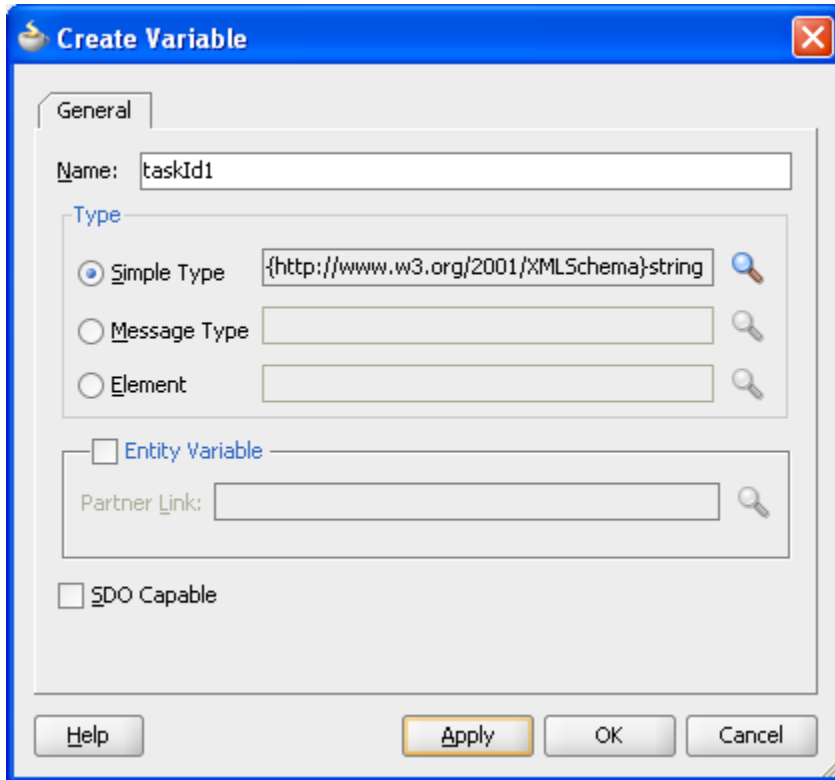
17. Put 'Java embedding' activity after the assign activity.



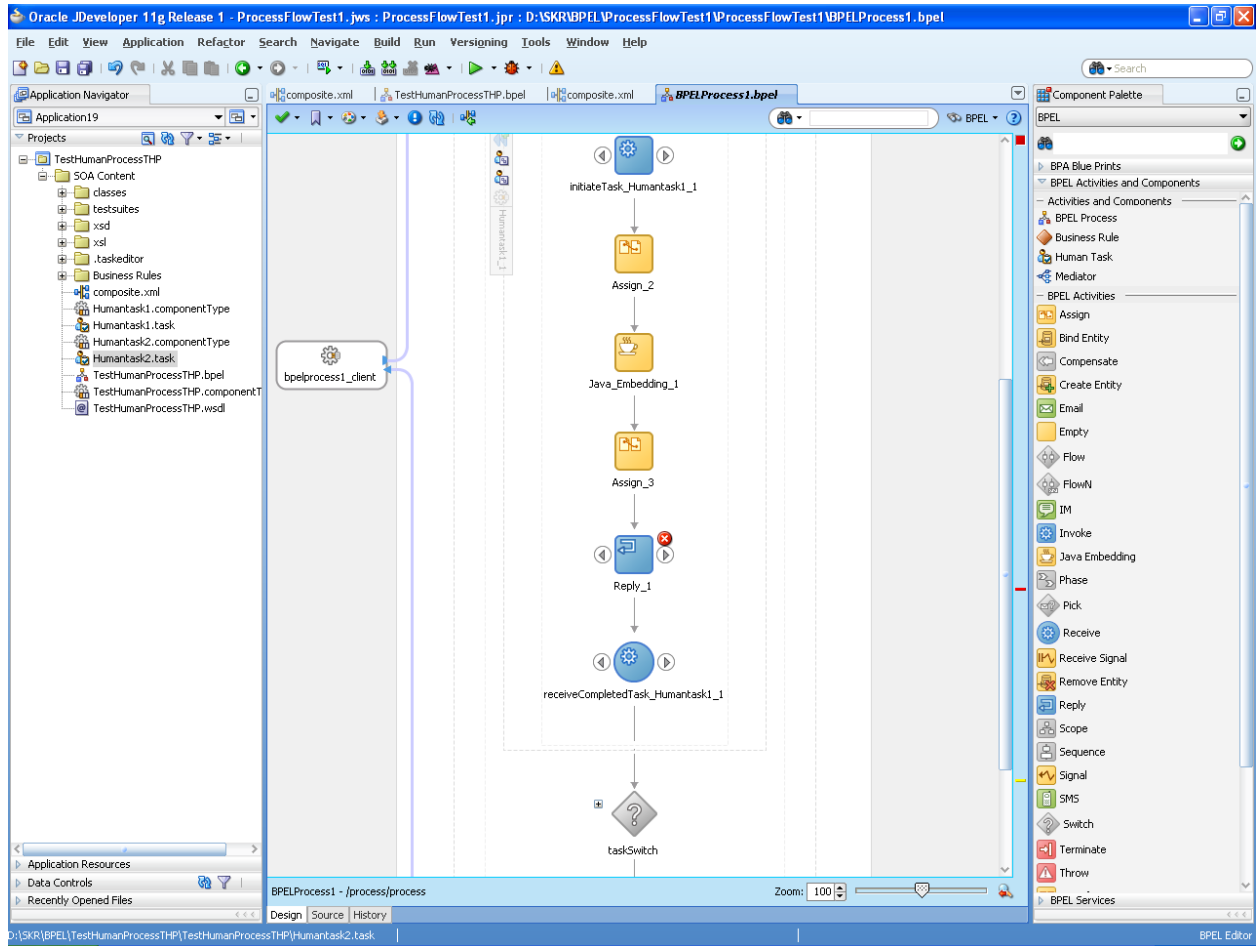
18. Put the 'Code snippet'



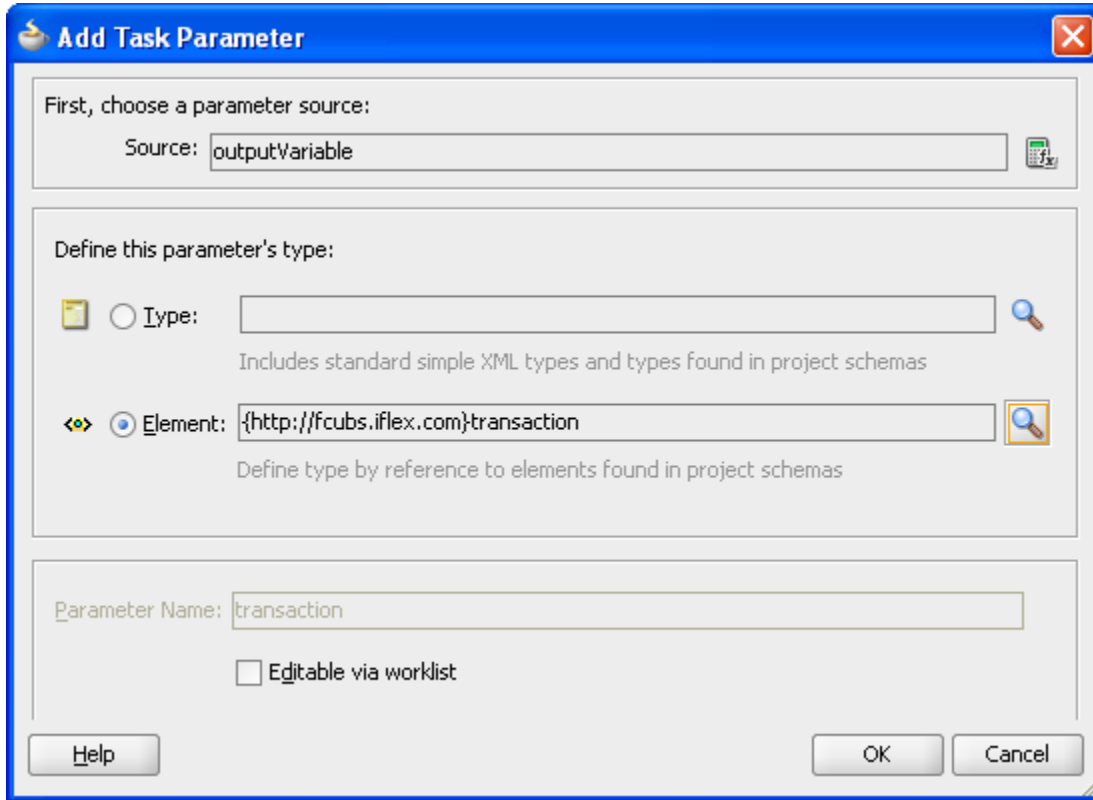
19. Put another 'Assign' activity after Java embedding and give 'Copy operation'



20. Put 'Reply' activity after the 'Assign'



22. Create another human task similarly



Create Human Task

Human Task Component

Create a Human Task Component to set up and customize workflow.






Name: Humantask2 Priority: 3 (Normal)

Namespace: http://xmlns.oracle.com/ProcessFlowTest1/ProcessFlowTest1/Humantask2

Title: TestTask2

Only static strings entered here will be included in the task definition

Outcomes: APPROVE,REJECT

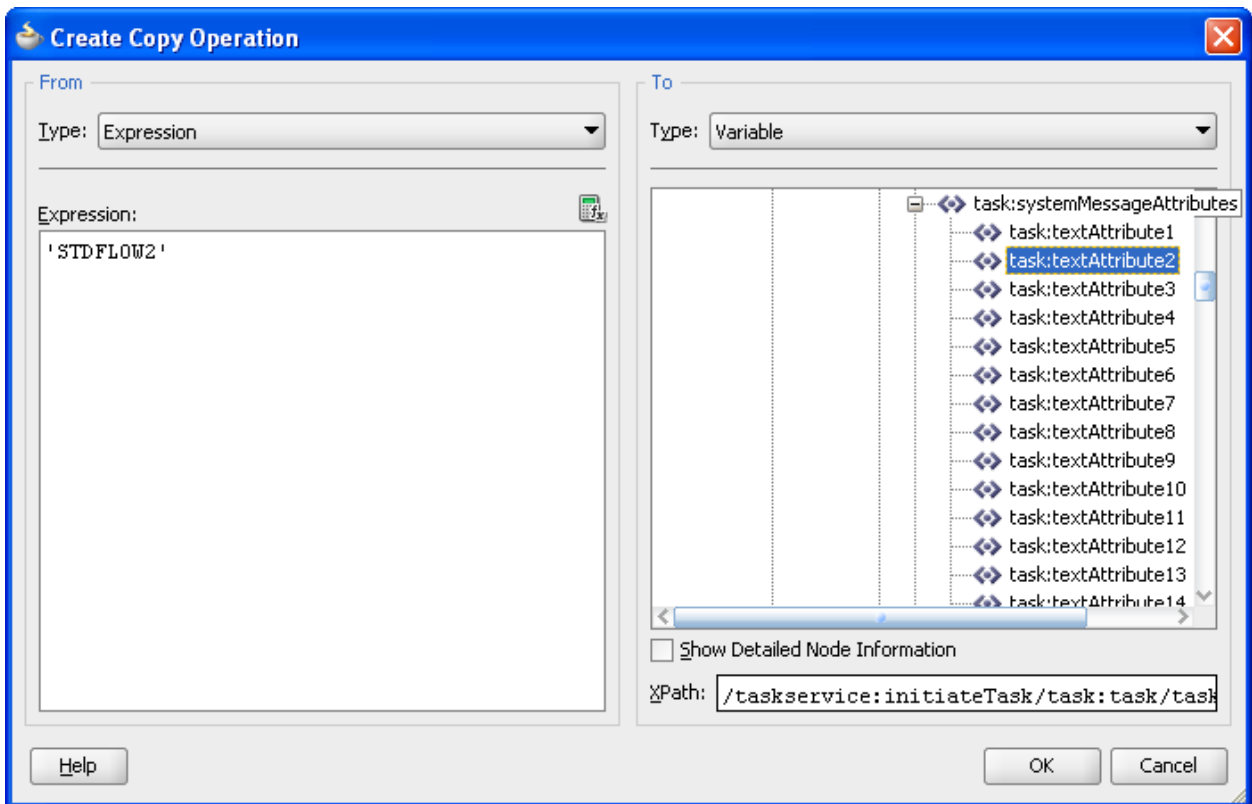
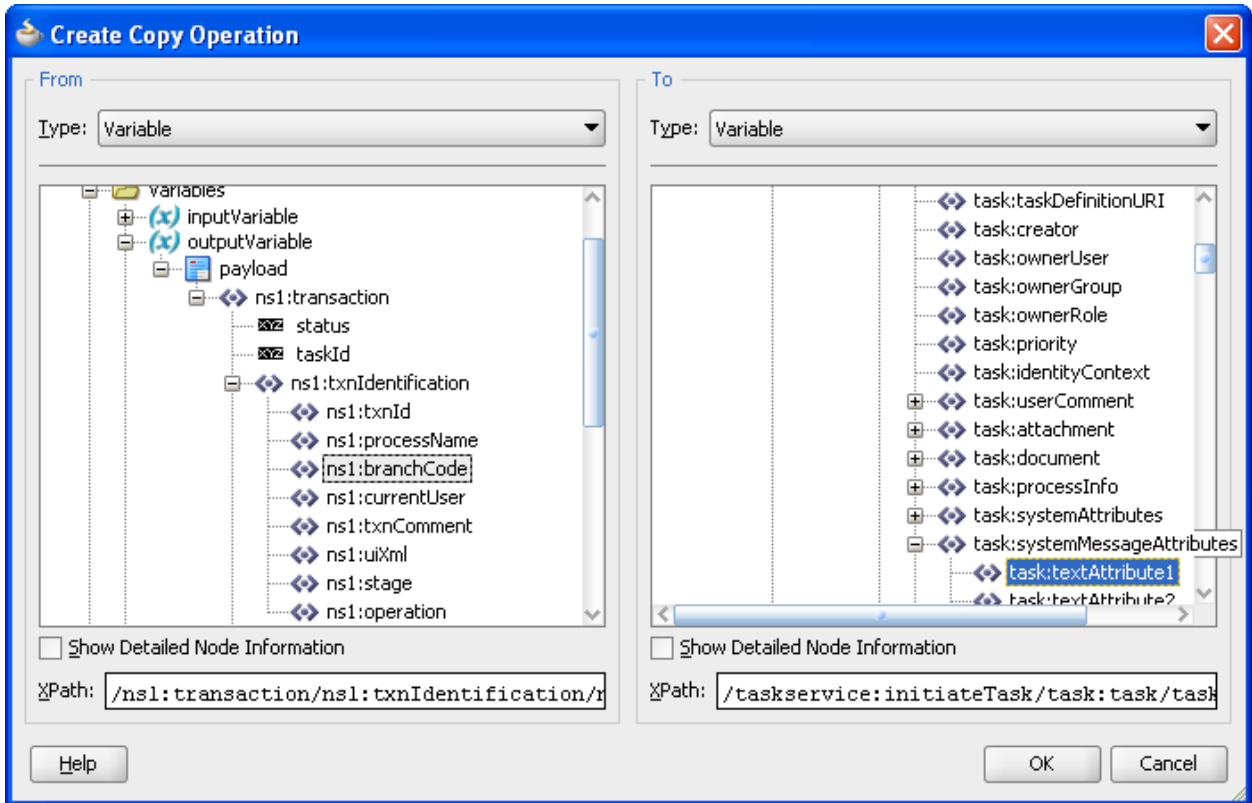
Parameters:   

Source	Payload In Task
outputVariable	transaction

Help

OK

Cancel



Oracle JDeveloper 11g Release 1 - ProcessFlowTest1_jws : ProcessFlowTest1_jpr : D:\ASKR\BPEL\ProcessFlowTest1\ProcessFlowTest1\BPELProcess1.bpel

File Edit View Application Refactor Search Navigate Build Run Versigning Tools Window Help

Application Navigator

ProcessFlowTest1

- ProcessFlowTest1
 - SOA Content
 - classes
 - testsuites
 - xsd
 - taskeditor
 - Business Rules
 - BPELProcess1.bpel
 - BPELProcess1.componentType
 - BPELProcess1.wsd
 - composite.xml
 - Humantask1.componentType
 - Humantask1.task
 - Humantask2.componentType
 - Humantask2.task

Component Palette

BPEL

- BPA Blue Prints
- BPEL Activities and Components
 - Activities and Components
 - BPEL Process
 - Business Rule
 - Human Task
 - Mediator
 - BPEL Activities
 - Assign
 - Bind Entity
 - Compensate
 - Create Entity
 - Email
 - Empty
 - Flow
 - FlowW
 - IM
 - Invoke
 - Java Embedding
 - Phase
 - Pick
 - Receive
 - Receive Signal
 - Remove Entity
 - Reply
 - Scope
 - Sequence
 - Signal
 - SMS
 - Switch
 - Terminate
 - Throw
 - BPEL Services

Design Source History

BPPEL - Log

XPath	Type	# of Error

Errors: Warnings: Last Validated On: Validation Search

Related Process File Warnings:

Warning	Model

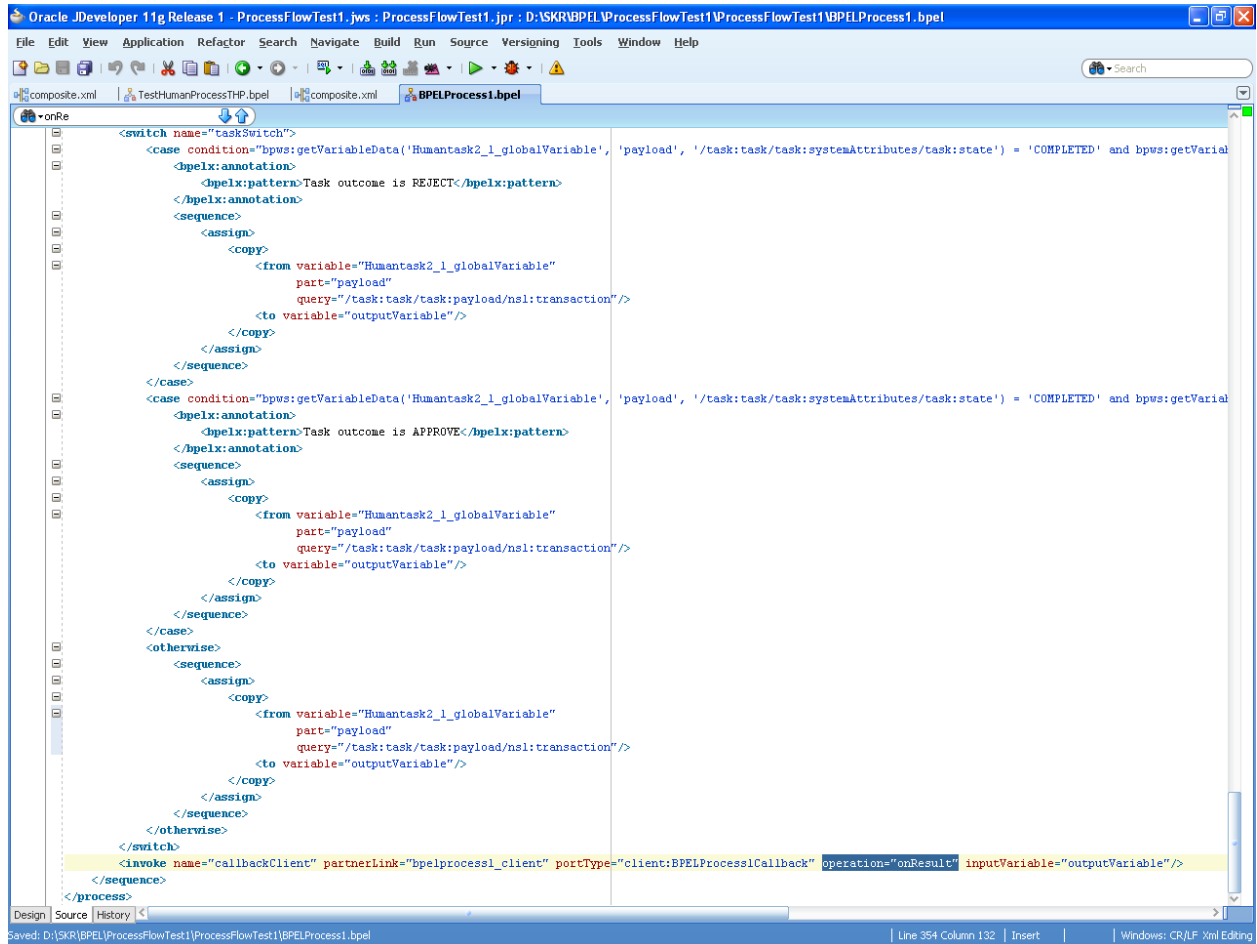
Errors: Warnings: Last Validated On:

Messages Extensions BPEL

Zoom: 100

The diagram shows a process flow starting with an external task 'Humantask2.Task...' which triggers an 'AssignTaskAttributes' activity. This is followed by an 'initiateTask_Humantask2_1' activity, and finally a 'receiveCompletedTask_Humantask2_1' activity. The flow is contained within a dashed box representing a task scope.

23. In .bpel change the operation to 'onResult' as highlighted.

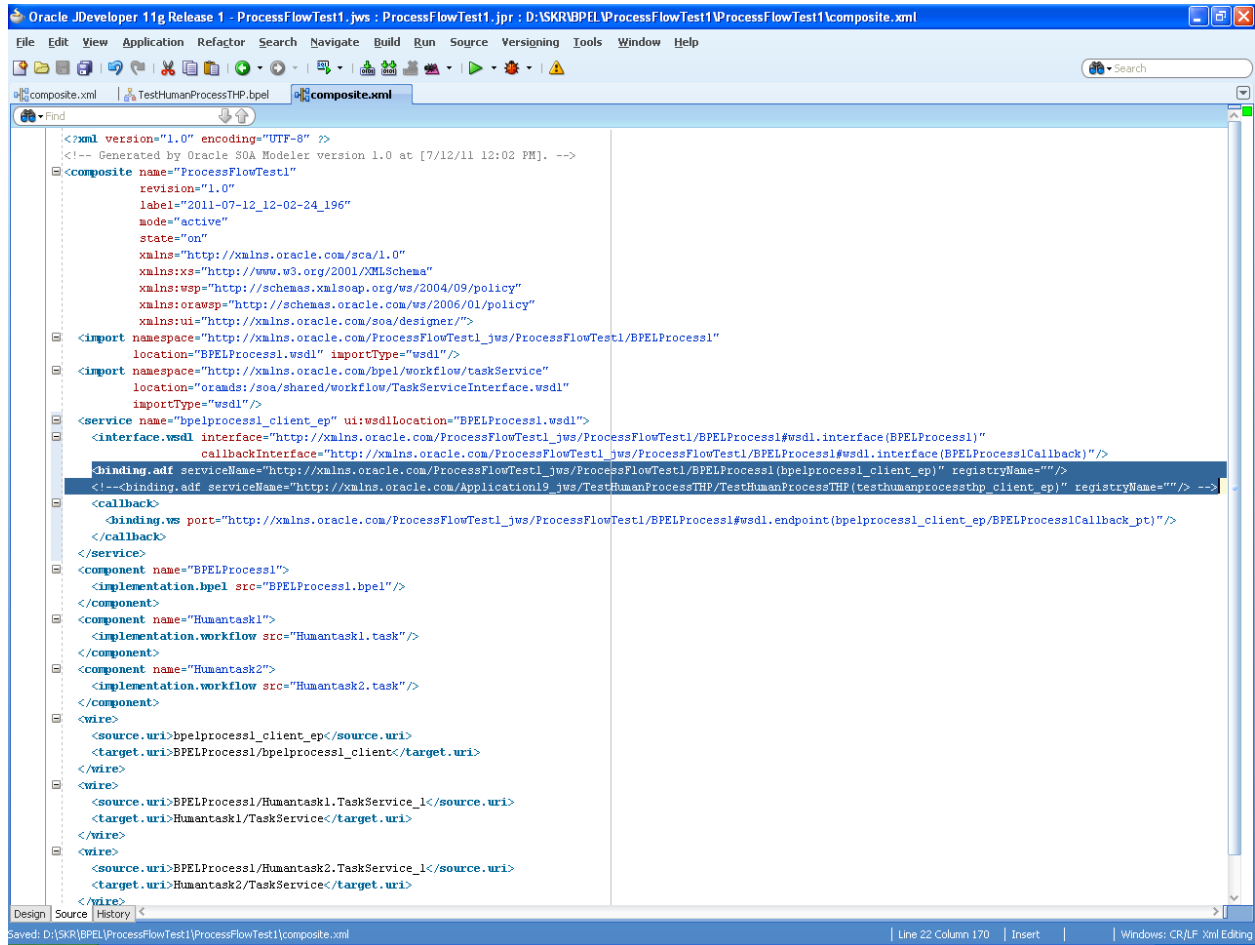


The screenshot shows the Oracle JDeveloper 11g IDE with a BPEL file named 'BPELProcess1.bpel' open. The code is displayed in a source editor with a tree view on the left. The tree view shows a sequence of activities, with the 'onRe' activity selected. The code in the editor is as follows:

```
<switch name="taskSwitch">
  <case condition="bpws:getVariableData('Humantask2_1_globalVariable', 'payload', '/task:task/task:systemAttributes/task:state') = 'COMPLETED' and bpws:getVariabl
    <bpel:annotation>
      <bpel:pattern>Task outcome is REJECT</bpel:pattern>
    </bpel:annotation>
    <sequence>
      <assign>
        <copy>
          <from variable="Humantask2_1_globalVariable"
            part="payload"
            query="/task:task/task:payload/ns1:transaction"/>
          <to variable="outputVariable"/>
        </copy>
      </assign>
    </sequence>
  </case>
  <case condition="bpws:getVariableData('Humantask2_1_globalVariable', 'payload', '/task:task/task:systemAttributes/task:state') = 'COMPLETED' and bpws:getVariabl
    <bpel:annotation>
      <bpel:pattern>Task outcome is APPROVE</bpel:pattern>
    </bpel:annotation>
    <sequence>
      <assign>
        <copy>
          <from variable="Humantask2_1_globalVariable"
            part="payload"
            query="/task:task/task:payload/ns1:transaction"/>
          <to variable="outputVariable"/>
        </copy>
      </assign>
    </sequence>
  </case>
  <otherwise>
    <sequence>
      <assign>
        <copy>
          <from variable="Humantask2_1_globalVariable"
            part="payload"
            query="/task:task/task:payload/ns1:transaction"/>
          <to variable="outputVariable"/>
        </copy>
      </assign>
    </sequence>
  </otherwise>
</switch>
<invoke name="callbackClient" partnerLink="bpelprocess1_client" portType="client:BPPELProcess1Callback" operation="onResult" inputVariable="outputVariable"/>
</sequence>
</process>
```

The 'onResult' operation in the 'invoke' element is highlighted in yellow. The status bar at the bottom indicates the file is saved at 'D:\SKR\BPEL\ProcessFlowTest1\ProcessFlowTest1\BPELProcess1.bpel', with the cursor at Line 354, Column 132.

24. In composite.xml change the 'binding.ws' as highlighted as we need .adf binding to connect to the Flexcube application.



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by Oracle SOA Modeler version 1.0 at [7/12/11 12:02 PM]. -->
<composite name="ProcessFlowTest1"
  revision="1.0"
  label="2011-07-12_12-02-24_196"
  mode="active"
  state="on"
  xmlns="http://xmlns.oracle.com/soa/1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:oraswp="http://schemas.oracle.com/ws/2006/01/policy"
  xmlns:ui="http://xmlns.oracle.com/soa/designer/">
  <import namespace="http://xmlns.oracle.com/ProcessFlowTest1_jws/ProcessFlowTest1/BPELProcess1"
    location="BPELProcess1.wsdl" importType="wsdl"/>
  <import namespace="http://xmlns.oracle.com/bpel/workflow/taskService"
    location="orasws/soa/shared/workflow/TaskServiceInterface.wsdl"
    importType="wsdl"/>
  <service name="bpelprocess1_client_ep" ui:wsdlLocation="BPELProcess1.wsdl">
    <interface wsdl interface="http://xmlns.oracle.com/ProcessFlowTest1_jws/ProcessFlowTest1/BPELProcess1/wsdl.interface(BPELProcess1)"
      callbackInterface="http://xmlns.oracle.com/ProcessFlowTest1_jws/ProcessFlowTest1/BPELProcess1/wsdl.interface(BPELProcess1Callback)"/>
    <binding adf serviceName="http://xmlns.oracle.com/Application9_jws/TestHumanProcessTHP/TestHumanProcessTHP(testhumanprocessth_client_ep)" registryName="" -->
    <callback>
      <binding.ws port="http://xmlns.oracle.com/ProcessFlowTest1_jws/ProcessFlowTest1/BPELProcess1/wsdl.endpoint(bpelprocess1_client_ep/BPELProcess1Callback_pt)"/>
    </callback>
  </service>
  <component name="BPELProcess1">
    <implementation.bpel src="BPELProcess1.bpel"/>
  </component>
  <component name="Humantask1">
    <implementation.workflow src="Humantask1.task"/>
  </component>
  <component name="Humantask2">
    <implementation.workflow src="Humantask2.task"/>
  </component>
  <wire>
    <source.uri>bpelprocess1_client_ep</source.uri>
    <target.uri>BPELProcess1/bpelprocess1_client</target.uri>
  </wire>
  <wire>
    <source.uri>BPELProcess1/Humantask1.TaskService_1</source.uri>
    <target.uri>Humantask1/TaskService</target.uri>
  </wire>
  <wire>
    <source.uri>BPELProcess1/Humantask2.TaskService_1</source.uri>
    <target.uri>Humantask2/TaskService</target.uri>
  </wire>
</composite>
```



BPEL_FCJ_Essentials
.docx

Further reference :