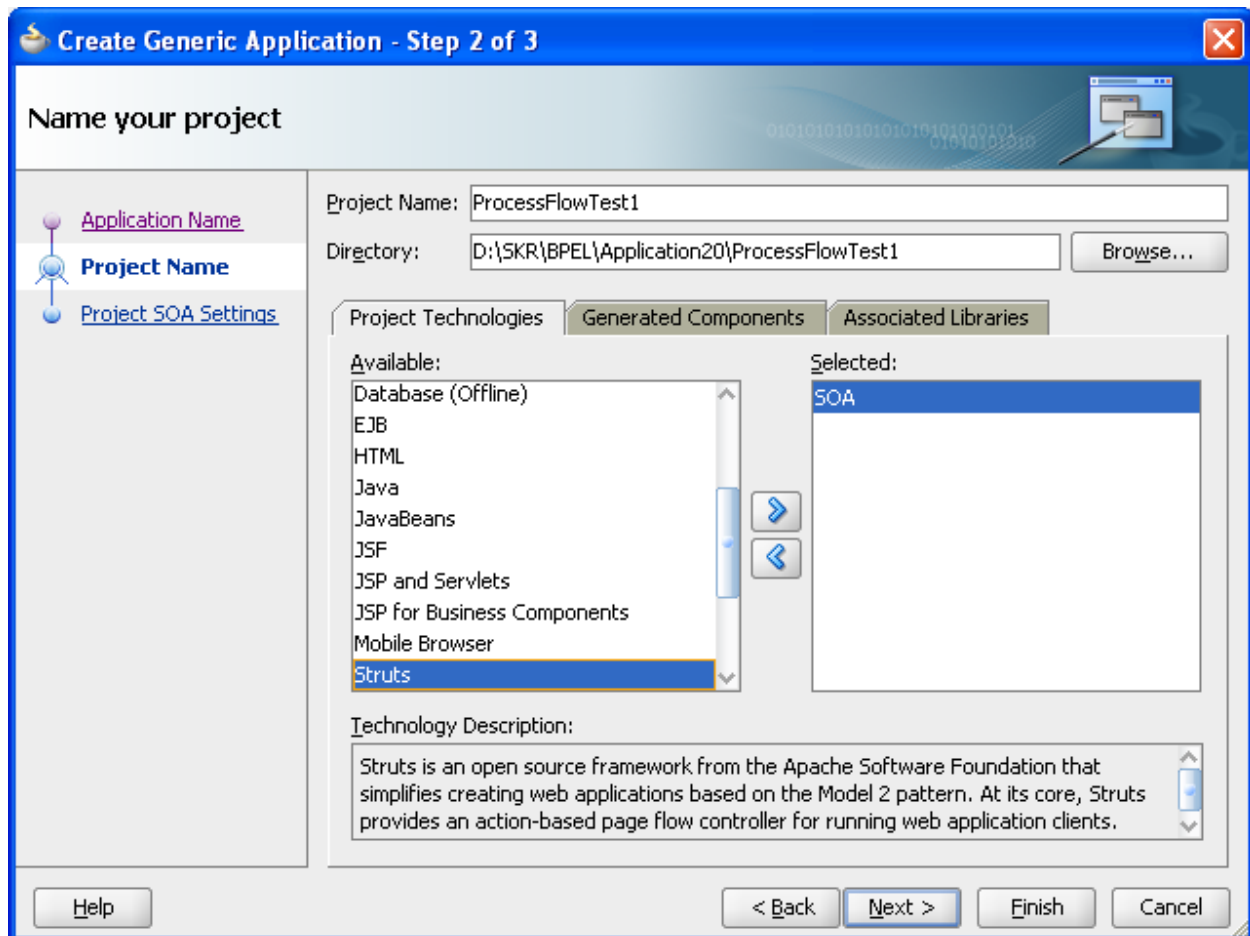


This Document is intended to explain as to how to develop simple process flow in FCUBS using BPEL, SOA Suite and JDeveloper

CREATING PROCESS FLOW

1. Create Generic application -> create SOA project.



2. Choose 'Empty Composite'

Create Generic Application - Step 3 of 3

Configure SOA settings

Application Name

Project Name

Project SOA Settings

Composite Name:
ProcessFlowTest1

Composite Template:

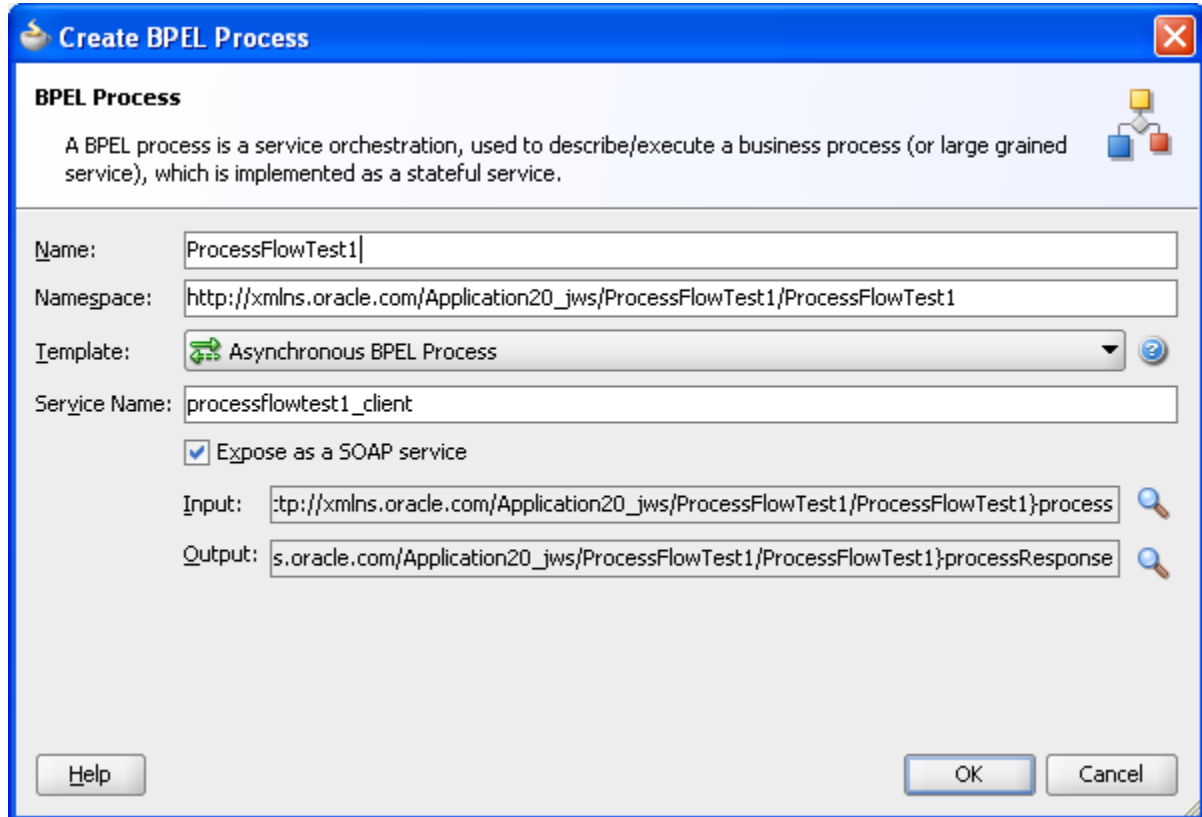
- Empty Composite
- Composite With BPEL
- Composite With Business Rule
- Composite With Human Task
- Composite With Mediator
- Composite From Oracle BPA Blueprint

☐ Customizable

Help

< Back Next > **Finish** Cancel

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



The image shows a 'Create BPEL Process' dialog box with a blue title bar. It contains a description of BPEL processes, a tree icon, and several input fields for configuring a new process. The fields are filled with specific values for a test project.



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:  Asynchronous BPEL Process 

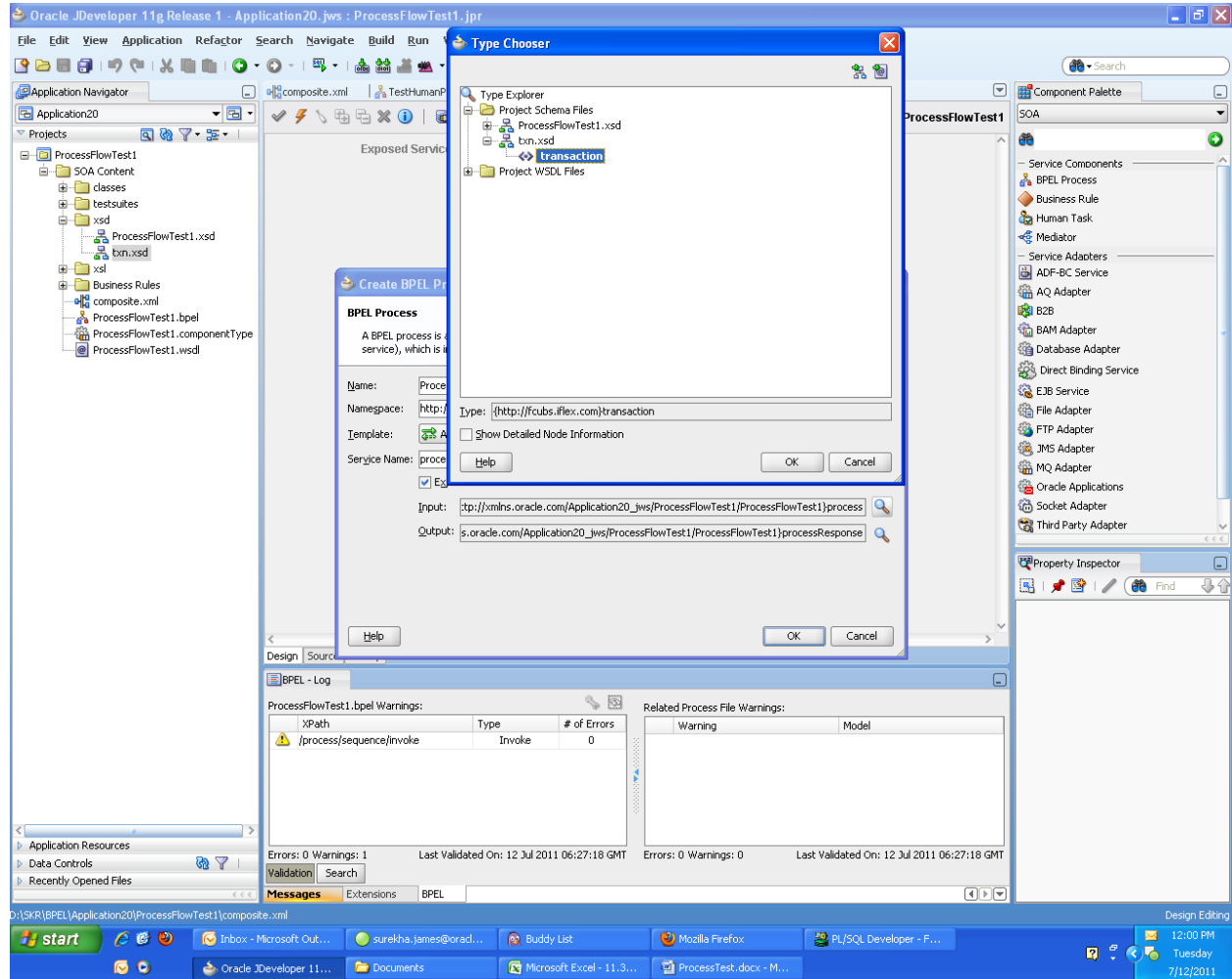
Service Name:

☒ Expose as a SOAP service

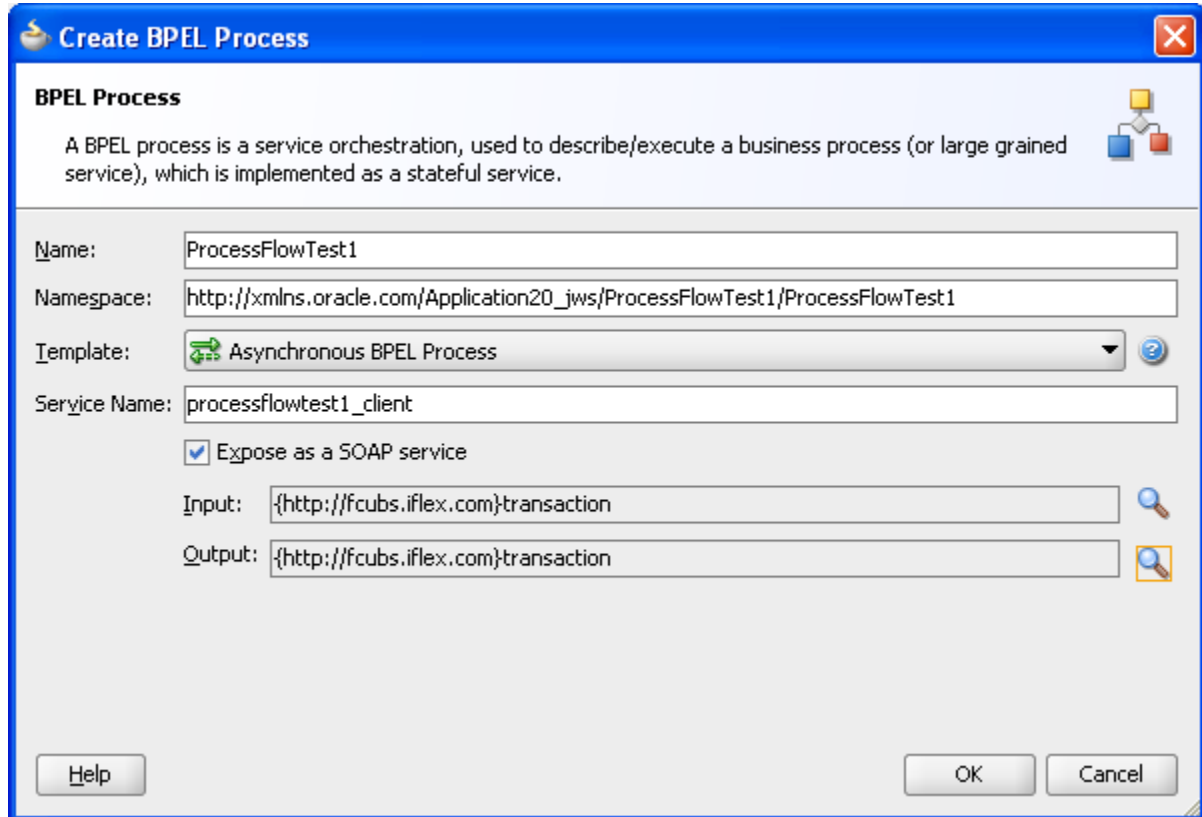
Input: 

Output: 

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



5. Click OK.



The image shows a 'Create BPEL Process' dialog box with a blue title bar and a close button. The main area has a light blue header with the title 'BPEL Process' and a small diagram icon. Below the header, a text box explains: '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.' The main form contains several fields: 'Name' with the value 'ProcessFlowTest1', 'Namespace' with the value 'http://xmlns.oracle.com/Application20_jws/ProcessFlowTest1/ProcessFlowTest1', 'Template' with a dropdown menu showing 'Asynchronous BPEL Process' and a help icon, 'Service Name' with the value 'processflowtest1_client', and a checked checkbox 'Expose as a SOAP service'. Below these are 'Input' and 'Output' fields, both containing the value '{http://fcubs.iflex.com}transaction' and a search icon. At the bottom, there are three buttons: 'Help', 'OK', and 'Cancel'.

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: ProcessFlowTest1

Namespace: http://xmlns.oracle.com/Application20_jws/ProcessFlowTest1/ProcessFlowTest1

Template: Asynchronous BPEL Process

Service Name: processflowtest1_client

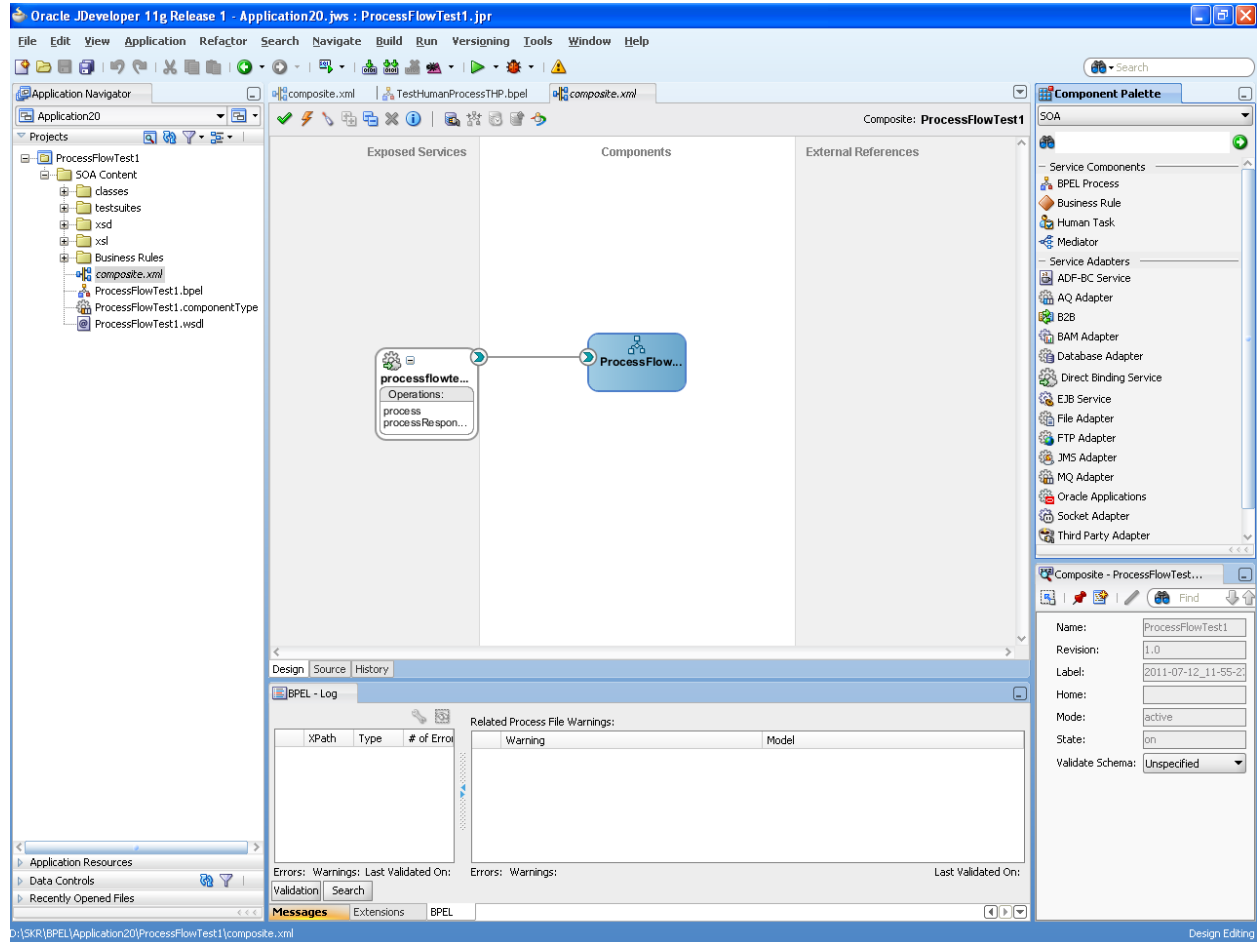
☒ Expose as a SOAP service

Input: {http://fcubs.iflex.com}transaction

Output: {http://fcubs.iflex.com}transaction

Help OK Cancel

6. Process is created.



7. Open the .bpel

The screenshot displays the Oracle JDeveloper 11g IDE with the BPEL editor open for 'ProcessFlowTest1.bpel'. The interface is divided into several panes:

- Project Navigator (Left):** Shows the project structure for 'ProcessFlowTest1', including folders like 'classes', 'testsuites', 'xsd', 'xsl', 'Business Rules', and files like 'composite.xml', 'ProcessFlowTest1.bpel', 'ProcessFlowTest1.componentType', and 'ProcessFlowTest1.wsdl'.
- Component Palette (Right):** Lists various BPEL activities and components, such as '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', 'FlowN', 'IM', 'Invoke', 'Java Embedding', 'Phase', 'Pick', 'Receive', 'Receive Signal', 'Remove Entity', 'Reply', 'Scope', 'Sequence', 'Signal', 'SMS', 'Switch', 'Terminate', and 'Throw'.
- Design Area (Center):** Displays the BPEL process diagram. It features a 'Partner Links' section on the left with a link named 'processflowtest1_...'. The main process flow starts with a 'receiveInput' activity, followed by a 'callbackClient' activity, and ends with a final state. A 'Test Hierarchy' pane is visible on the left side of the design area.
- Bottom Status Bar:** Shows validation results: 'Errors: 0 Warnings: 1' and 'Last Validated On: 12 Jul 2011 06:27:18 GMT'. It also includes a 'Messages' tab and a 'BPEL Editor' label.

8. Drag and drop 'Assign' activity

The screenshot shows the Oracle JDeveloper 11g Release 1 interface. The main window is titled "Oracle JDeveloper 11g Release 1 - Application20.jws : ProcessFlowTest1.jpr : D:\SKR\BPEL\Application20\ProcessFlowTest1\ProcessFlowTest1.bpel". The interface includes a menu bar (File, Edit, View, Application, Refactor, Search, Navigate, Build, Run, Versigning, Tools, Window, Help), a toolbar, and several panes.

Application Navigator (Left Pane): Shows the project structure for "ProcessFlowTest1", including folders for "classes", "testsuites", "xsd", "xsl", "Business Rules", and files for "composite.xml", "ProcessFlowTest1.bpel", "ProcessFlowTest1.componentType", and "ProcessFlowTest1.wsdl".

Component Palette (Right Pane): Lists various BPEL activities and components, including "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", "FlowN", "IM", "Invoke", "Java Embedding", "Phase", "Pick", "Receive", "Receive Signal", "Remove Entity", "Reply", "Scope", "Sequence", "Signal", "SMS", "Switch", "Terminate", "Throw", and "BPEL Services".

Design View (Main Canvas): Displays a BPEL process diagram. It includes a "Partner Links" section on the left, a central flow area with a "receiveInput" activity, an "Assign_I" activity (highlighted with a red 'X' icon), and a "callbackClient" activity. The flow starts with a "start" node, goes to "receiveInput", then to "Assign_I", and finally to "callbackClient".

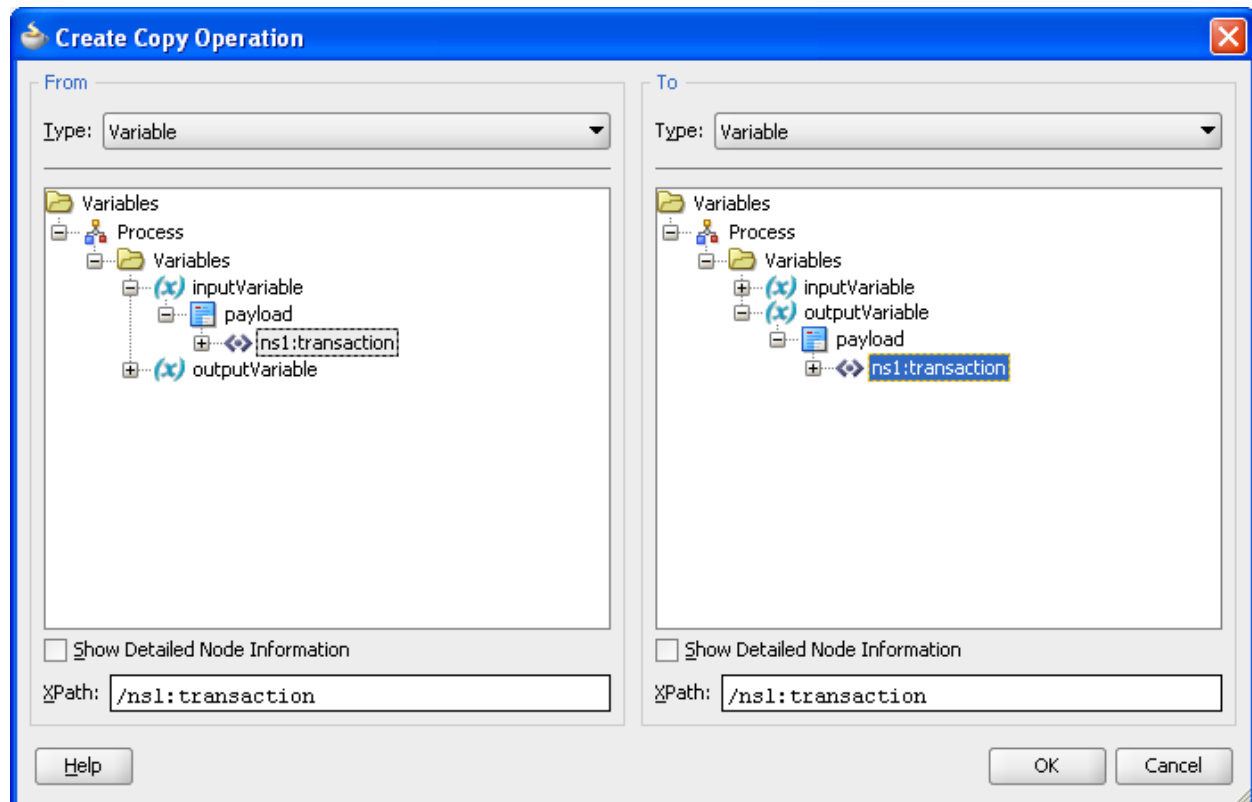
BPEL - Log (Bottom Pane): Shows warnings for the process. The "ProcessFlowTest1.bpel Warnings:" table has the following data:

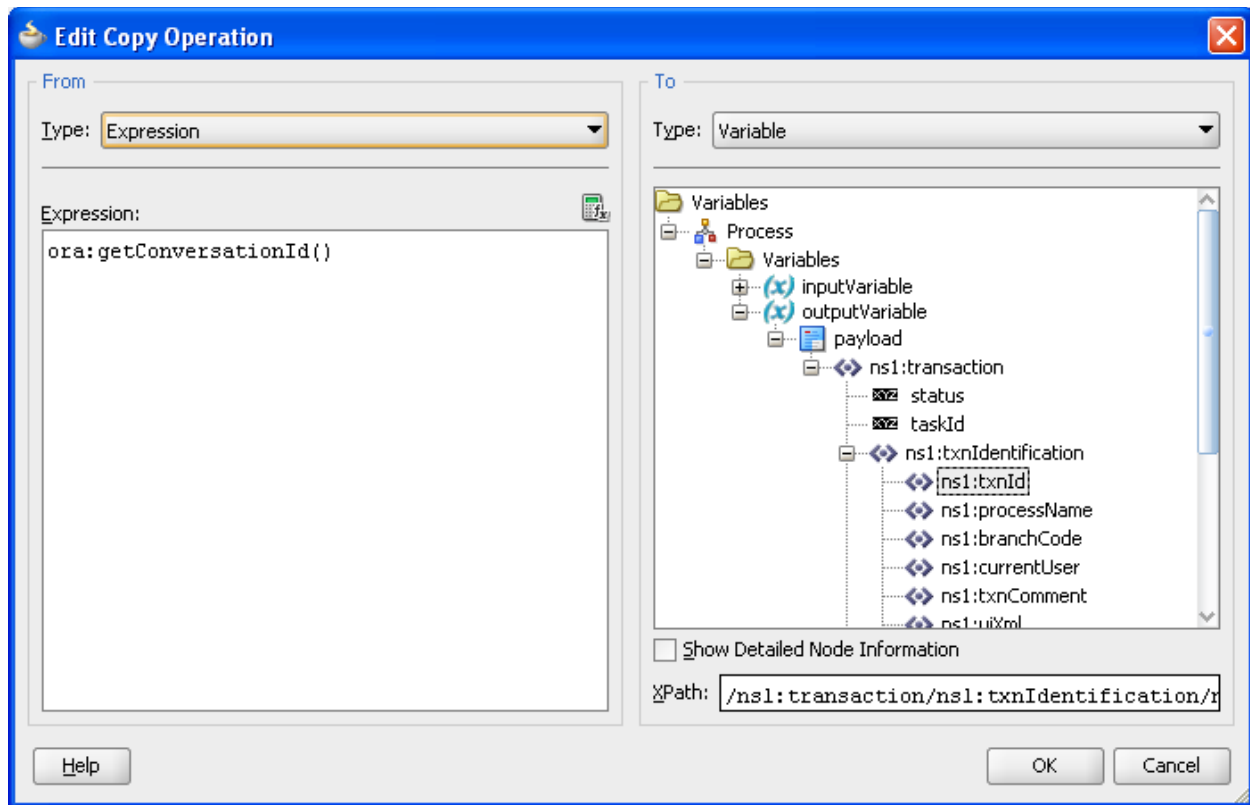
XPath	Type	# of Errors
/process/sequence/invoke	Invoke	0

The "Related Process File Warnings:" table is empty.

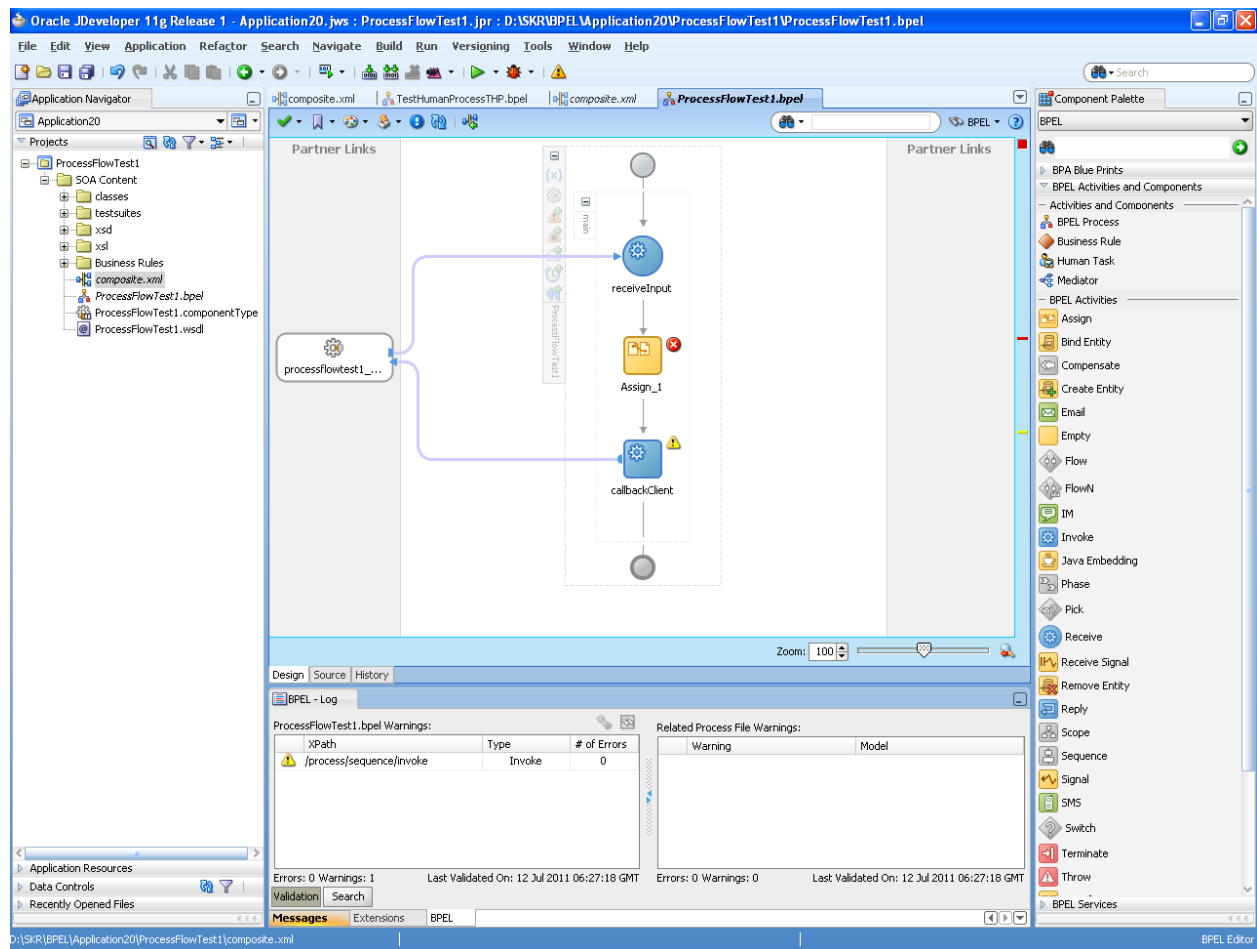
Status Bar (Bottom): Shows "Errors: 0 Warnings: 1" and "Last Validated On: 12 Jul 2011 06:27:18 GMT". The "Messages" tab is selected, showing "BPEL" as the current context.

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.


Add Task Parameter

First, choose a parameter source:

Source:

Define this parameter's type:

☐ Type: 
Includes standard simple XML types and types found in project schemas

☒ Element: 
Define type by reference to elements found in project schemas



Parameter Name:

☐ Editable via worklist

[Help](#) [OK](#) [Cancel](#)

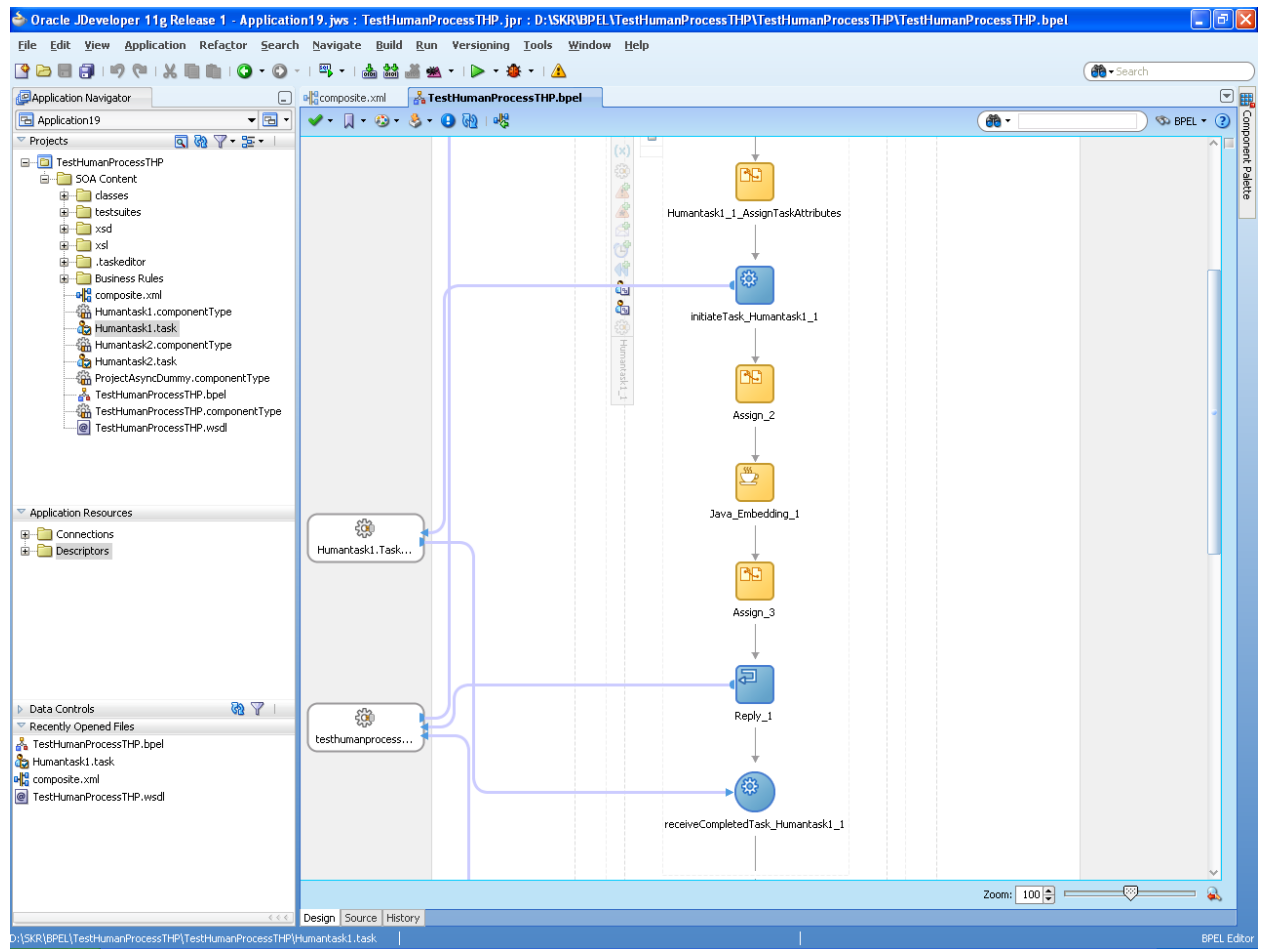
Use External Routing

Class Name:

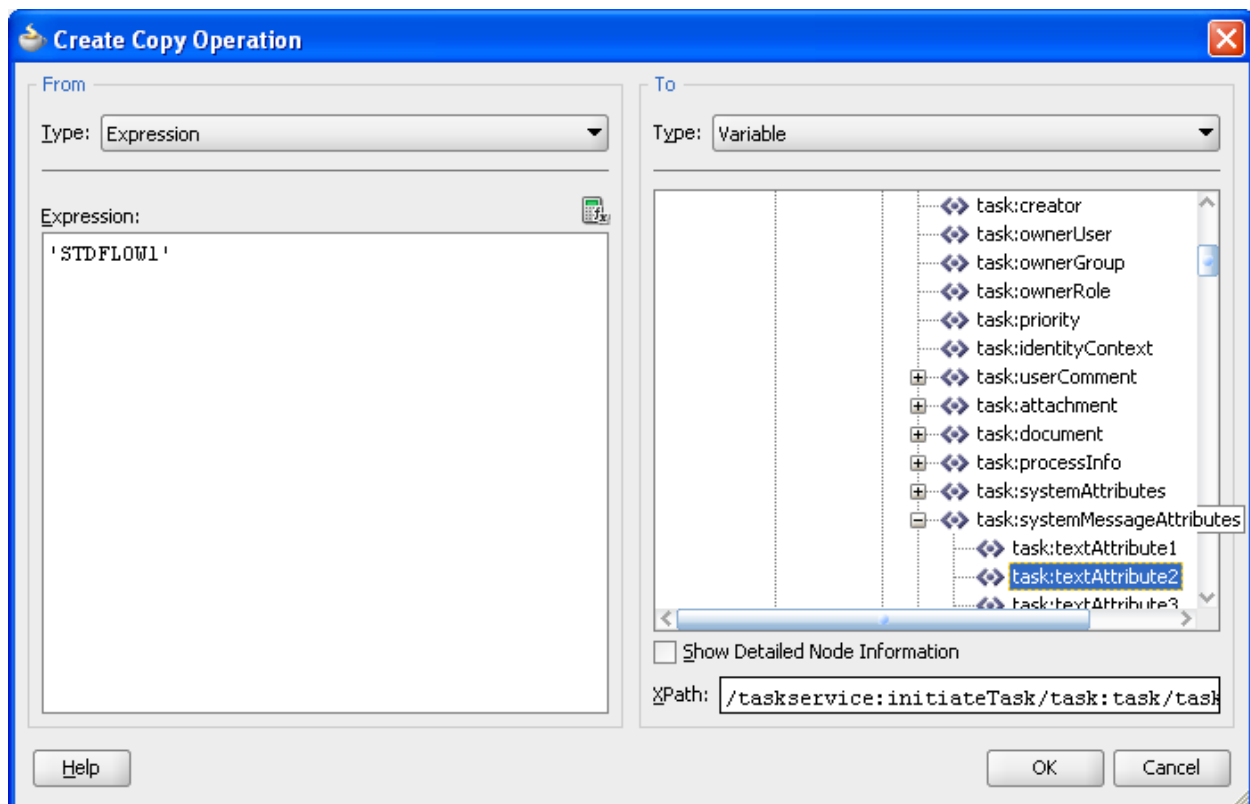
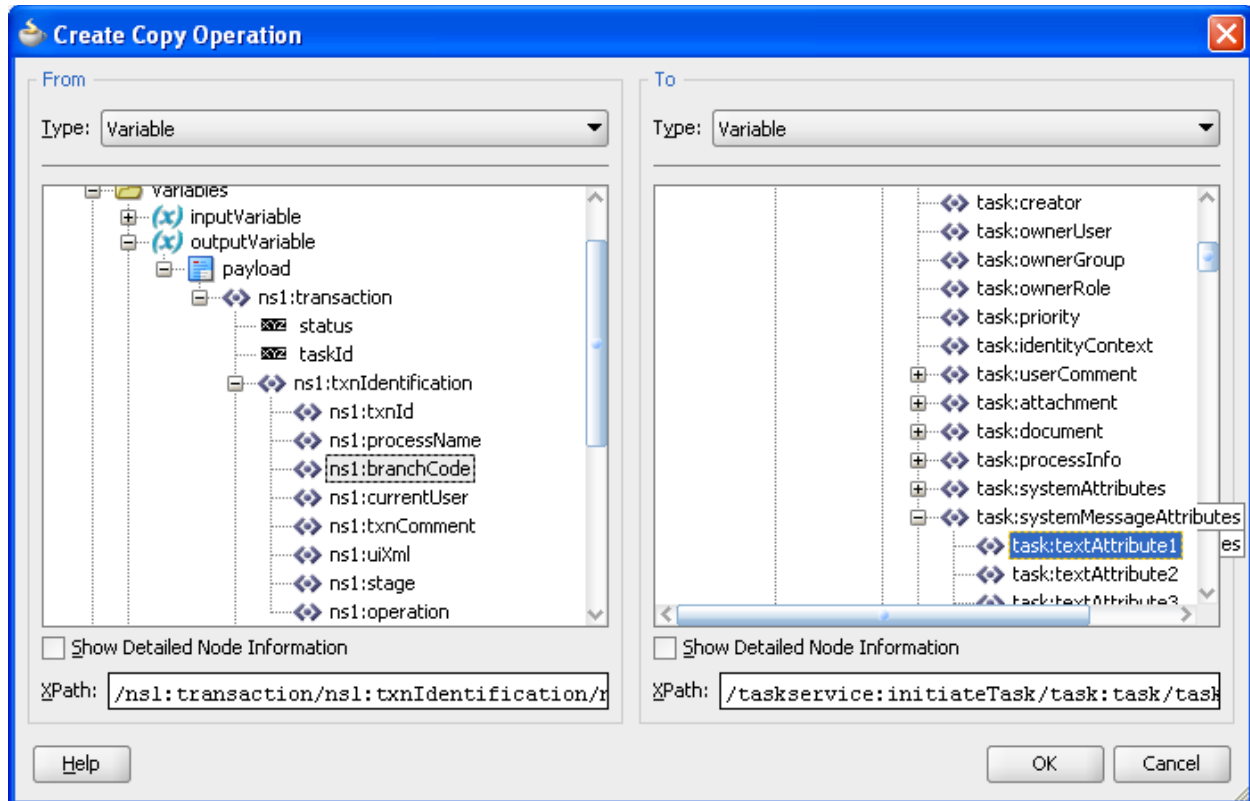
Name:	Value		
-------	-------	---	---

[Help](#) [OK](#) [Cancel](#)

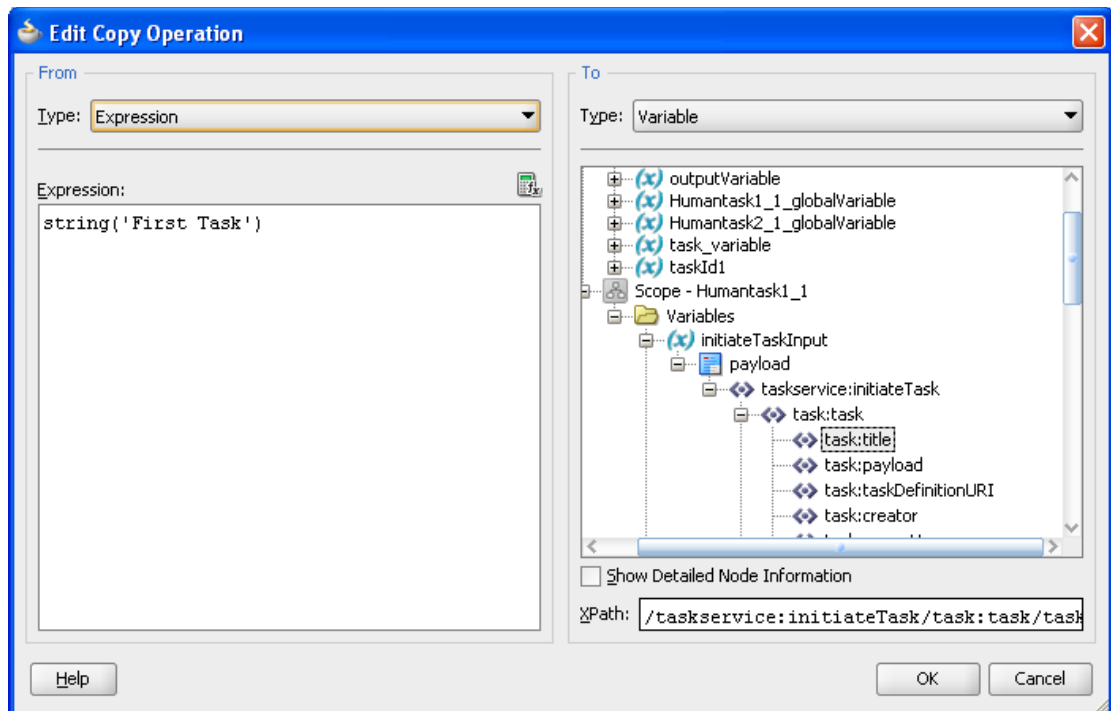
12. Expanded Humantask shown below.



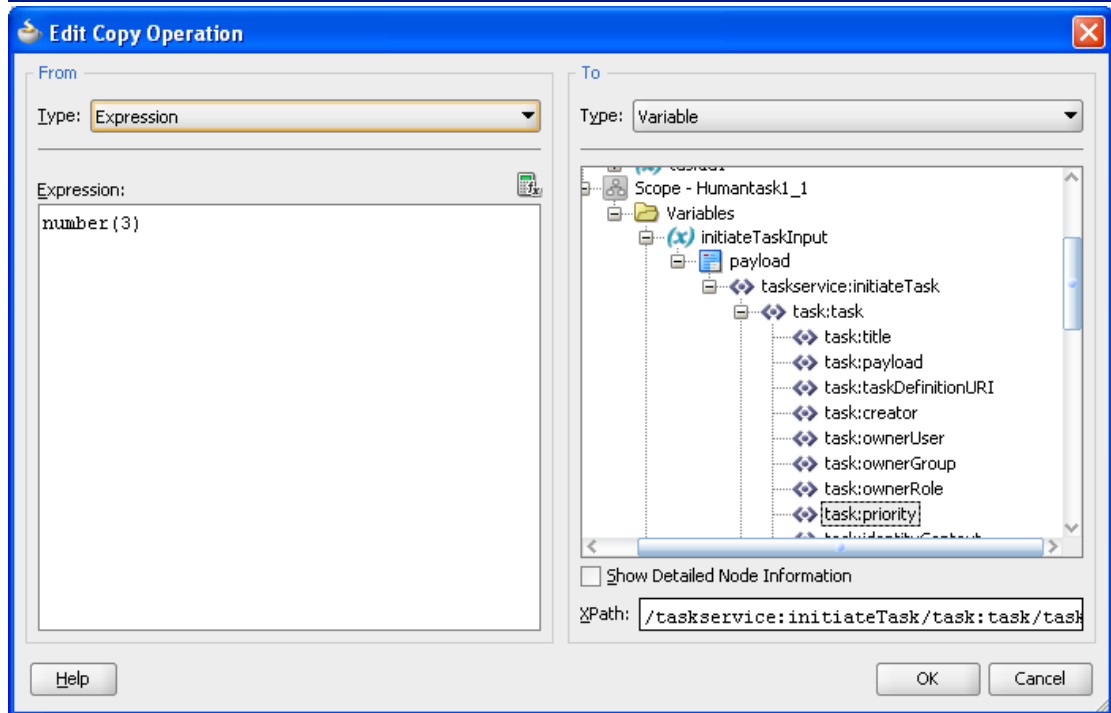
13. Human task is created.



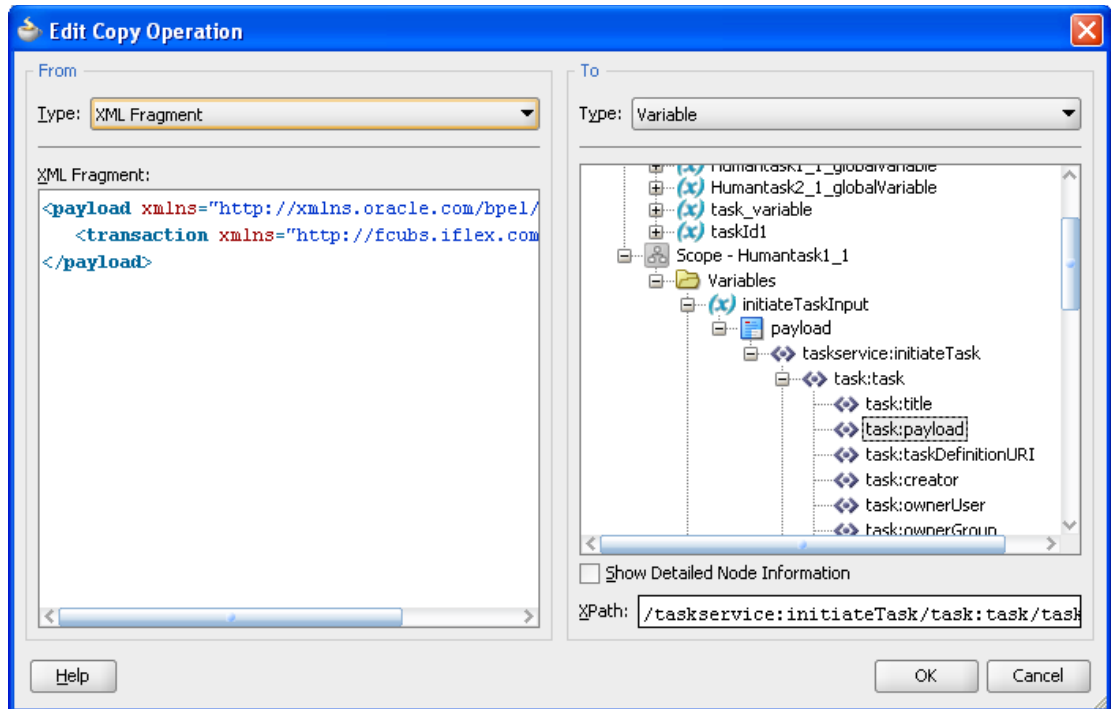
Note : Other automatic assignments which are defaulted on creation of Human Task are as follows-



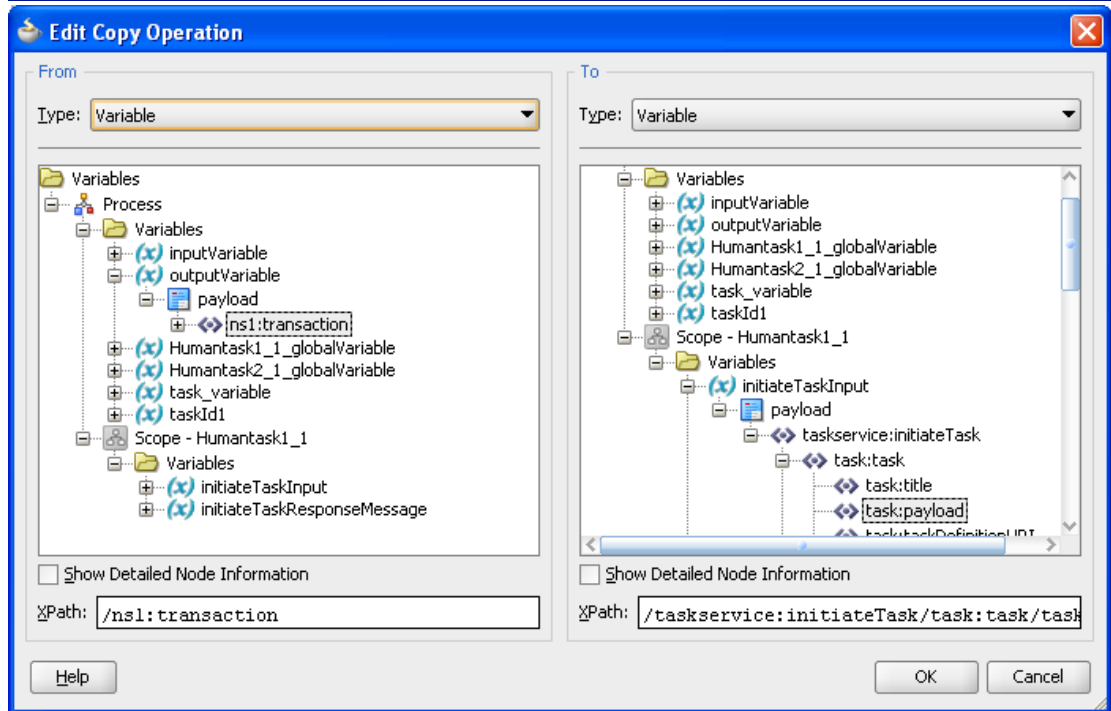
i.



ii.

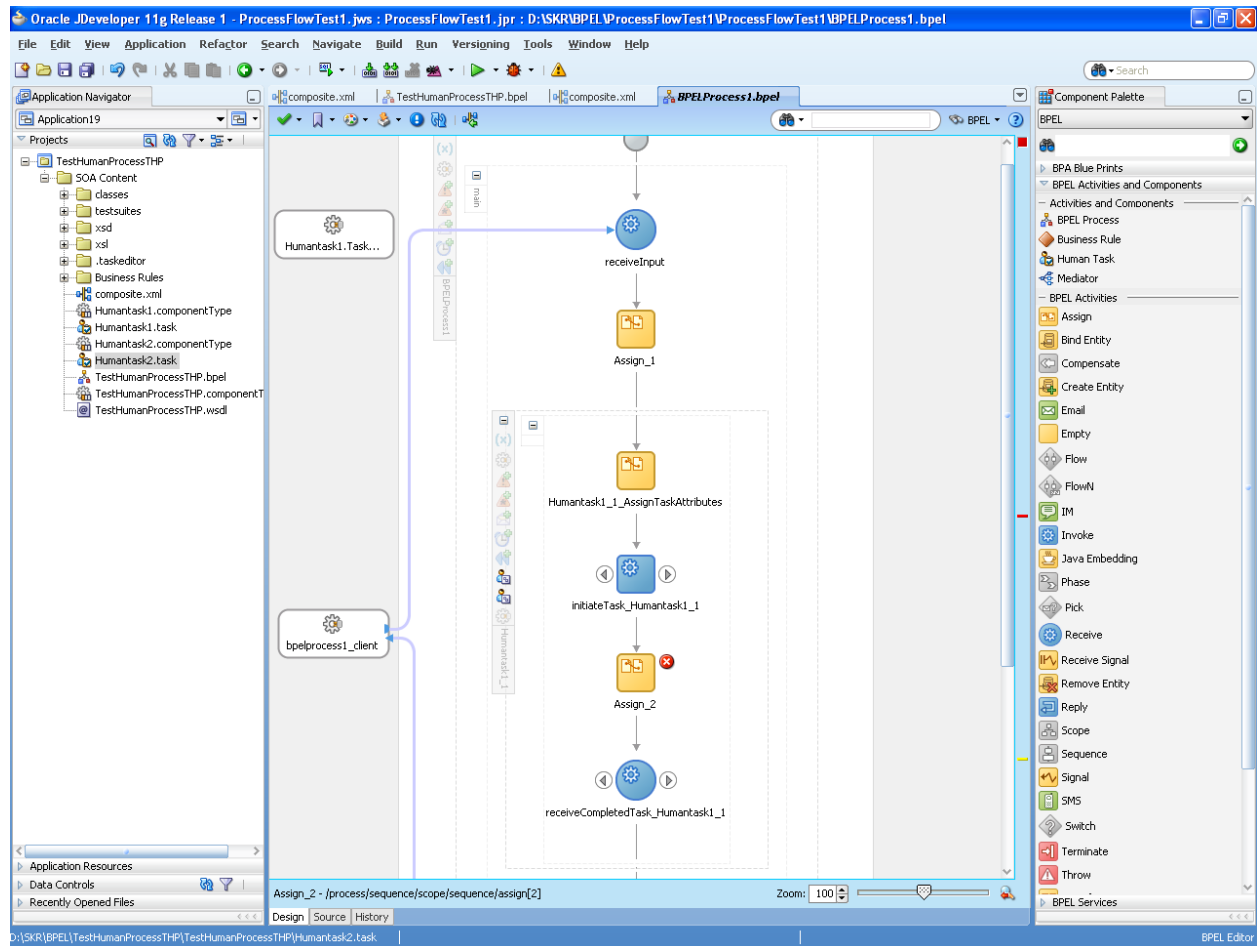


iii.

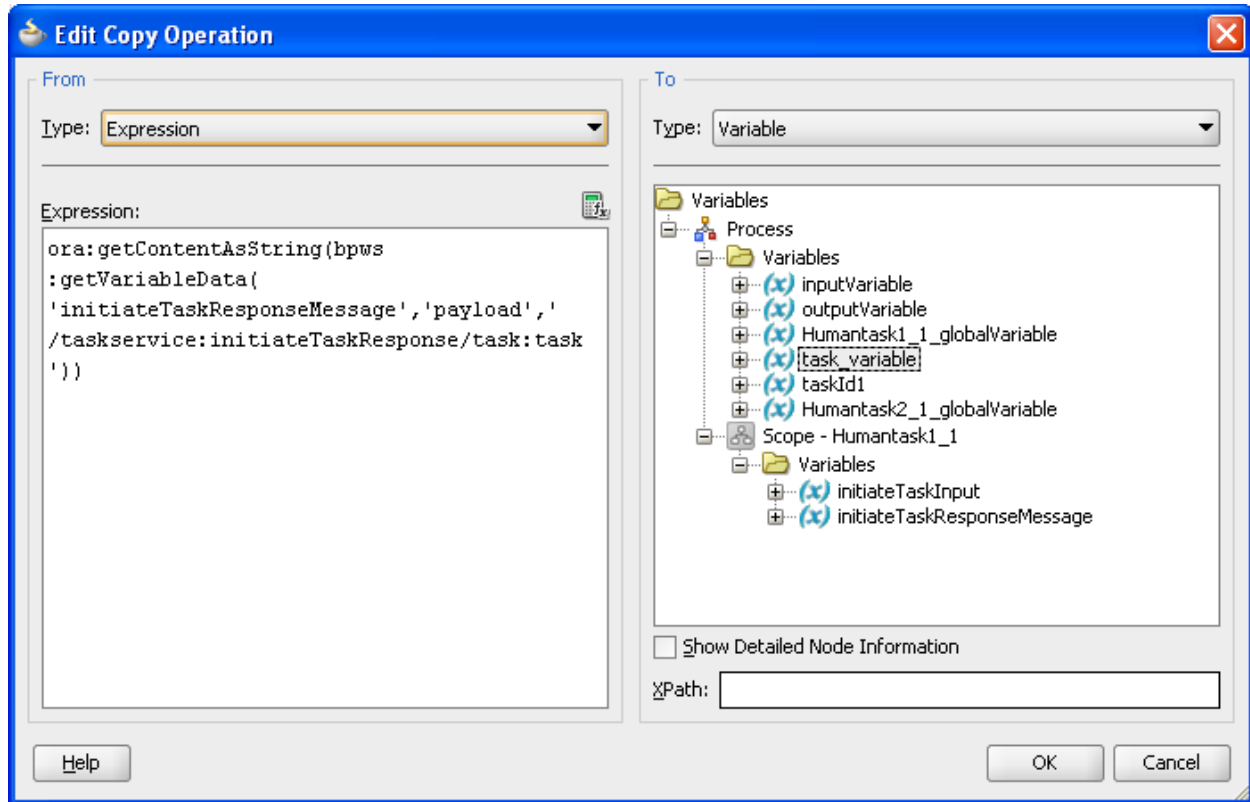


iv.

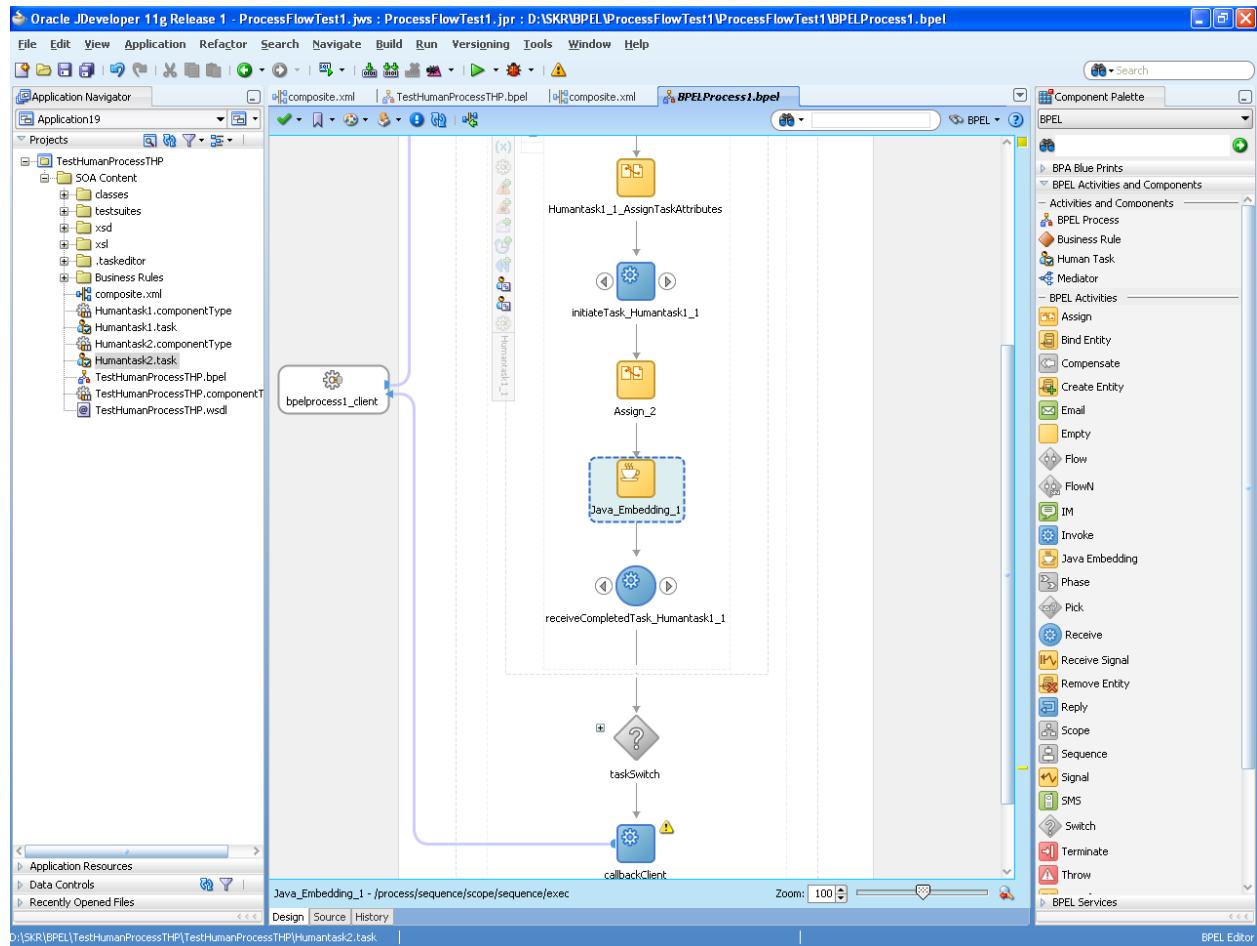
15. Put another 'Assign' activity in Human task



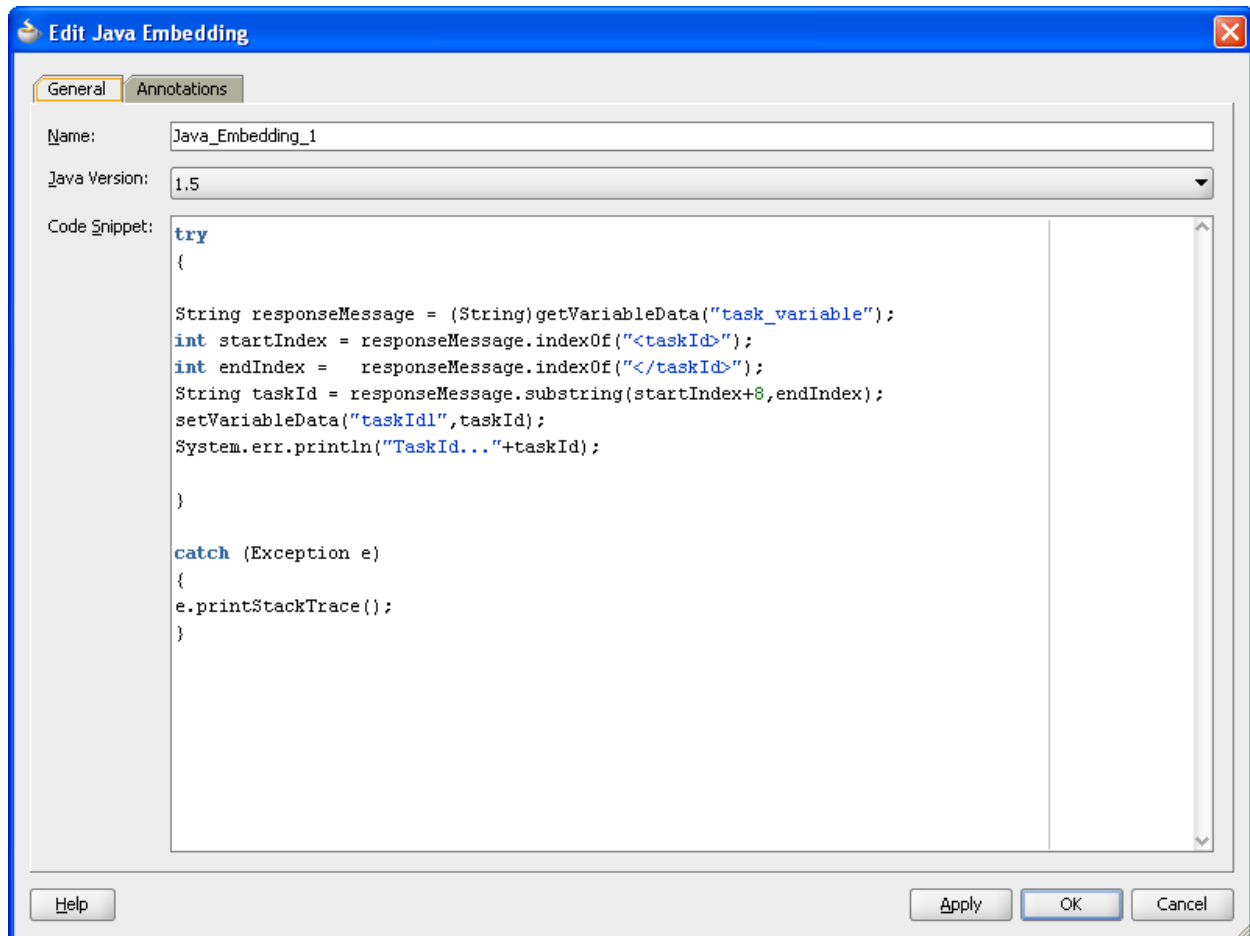
16. Give the 'Copy Operation' for 'assign' activity. Create a new variable by right clicking on 'variables' in Variables -> Process -> Variables



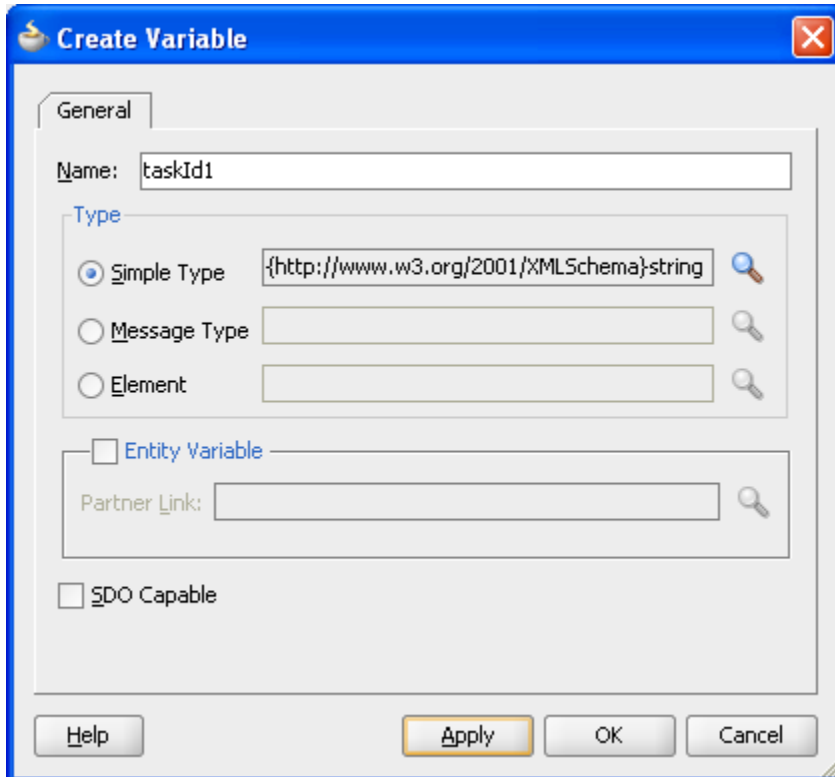
17. Put 'Java embedding' activity after the assign activity.



18. Put the 'Code snippet'



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



The 'Create Variable' dialog box is shown with the 'General' tab selected. The 'Name' field contains 'taskId1'. Under the 'Type' section, 'Simple Type' is selected with the value '{http://www.w3.org/2001/XMLSchema}string'. The 'Message Type' and 'Element' options are unselected. The 'Entity Variable' checkbox is unchecked, and the 'Partner Link' field is empty. The 'SDO Capable' checkbox is also unchecked. At the bottom, there are buttons for 'Help', 'Apply', 'OK', and 'Cancel'.

Create Variable

General

Name:

Type

☒ Simple Type

☐ Message Type

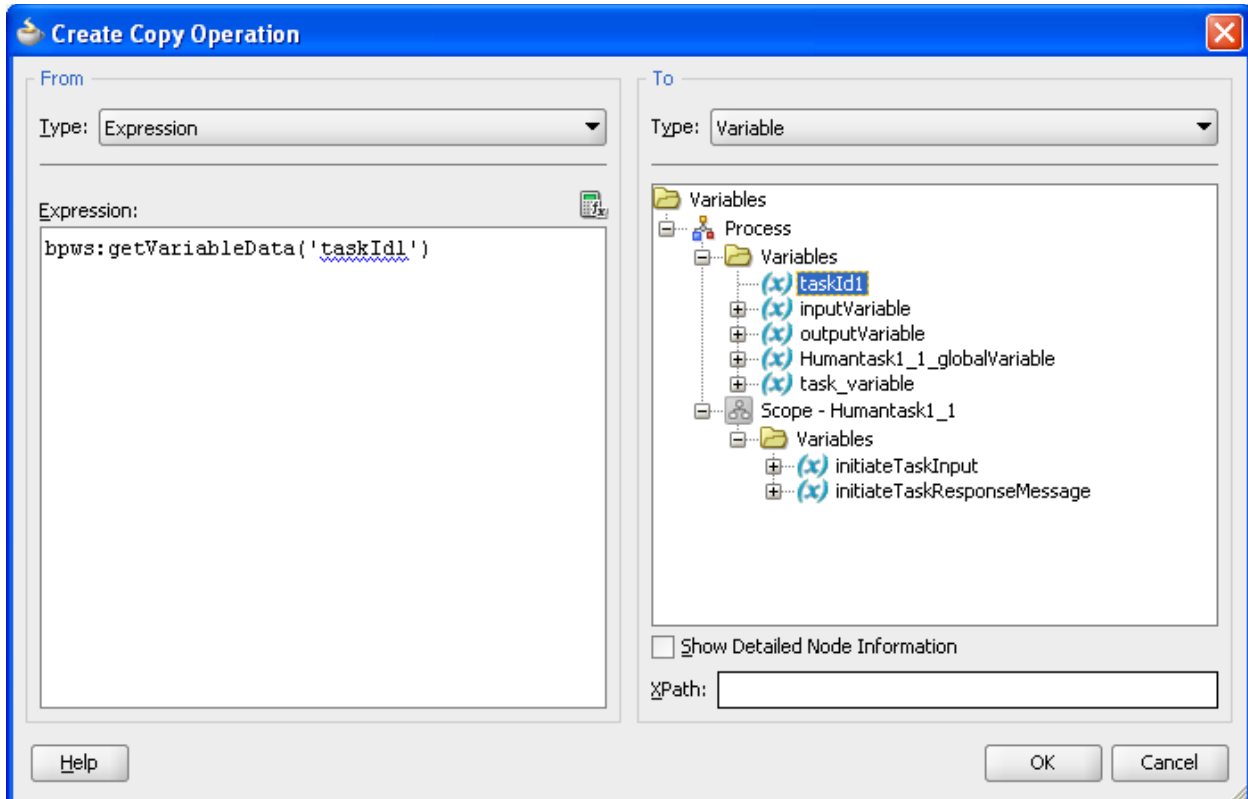
☐ Element

☐ Entity Variable

Partner Link:

☐ SDO Capable

Help Apply OK Cancel



The 'Create Copy Operation' dialog box is shown. The 'From' section has 'Type' set to 'Expression' and the 'Expression' field contains 'bpws:getVariableData('taskId1')'. The 'To' section has 'Type' set to 'Variable'. A tree view on the right shows the variable hierarchy: 'Process' contains 'Variables', which contains 'taskId1', 'inputVariable', 'outputVariable', 'Humantask1_1_globalVariable', and 'task_variable'. 'Scope - Humantask1_1' contains 'Variables', which contains 'initiateTaskInput' and 'initiateTaskResponseMessage'. The 'Show Detailed Node Information' checkbox is unchecked, and the 'XPath' field is empty. At the bottom, there are buttons for 'Help', 'OK', and 'Cancel'.

Create Copy Operation

From

Type:

Expression:

To

Type:

Variables

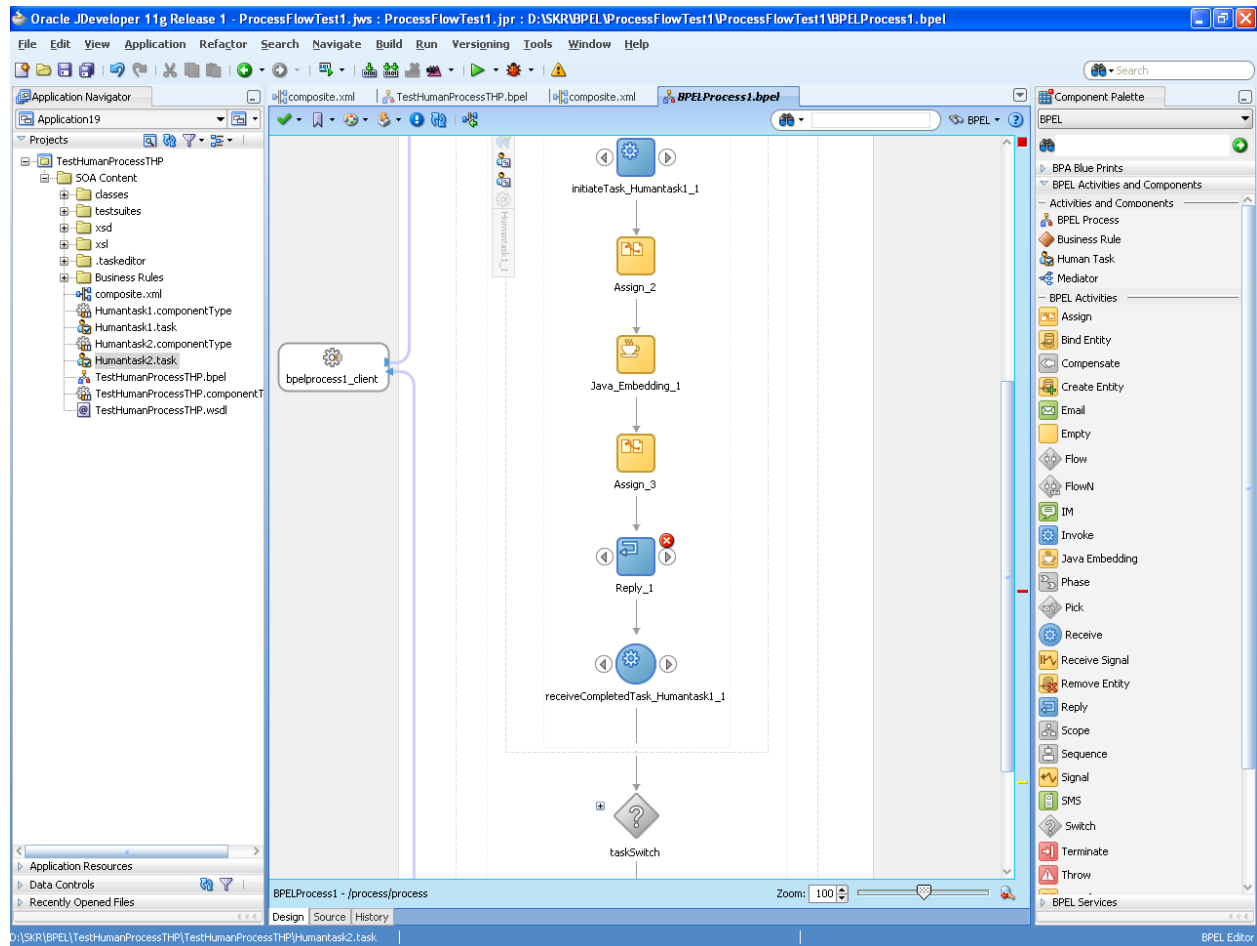
- Process
 - Variables
 - taskId1
 - inputVariable
 - outputVariable
 - Humantask1_1_globalVariable
 - task_variable
 - Scope - Humantask1_1
 - Variables
 - initiateTaskInput
 - initiateTaskResponseMessage

☐ Show Detailed Node Information

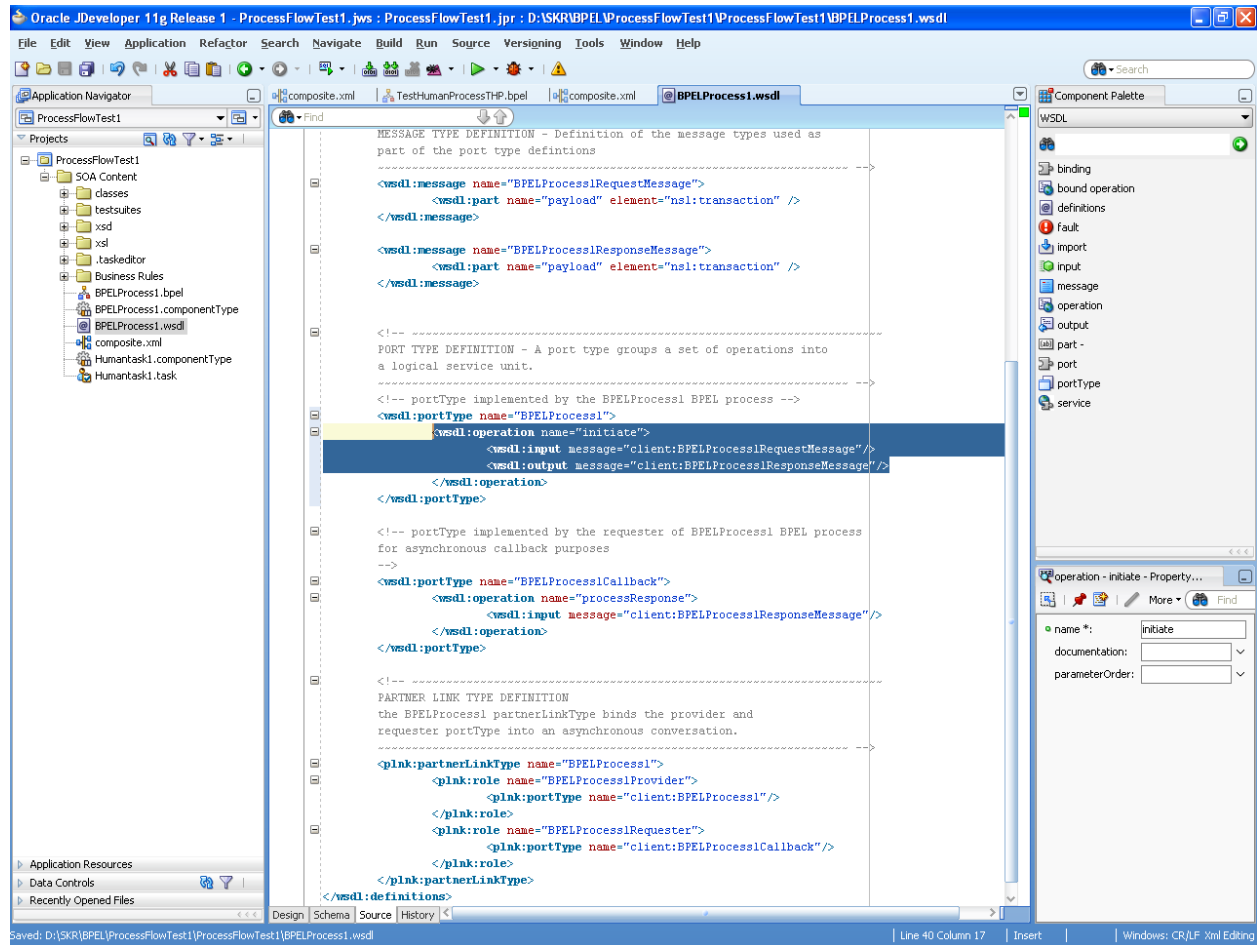
XPath:

Help OK Cancel

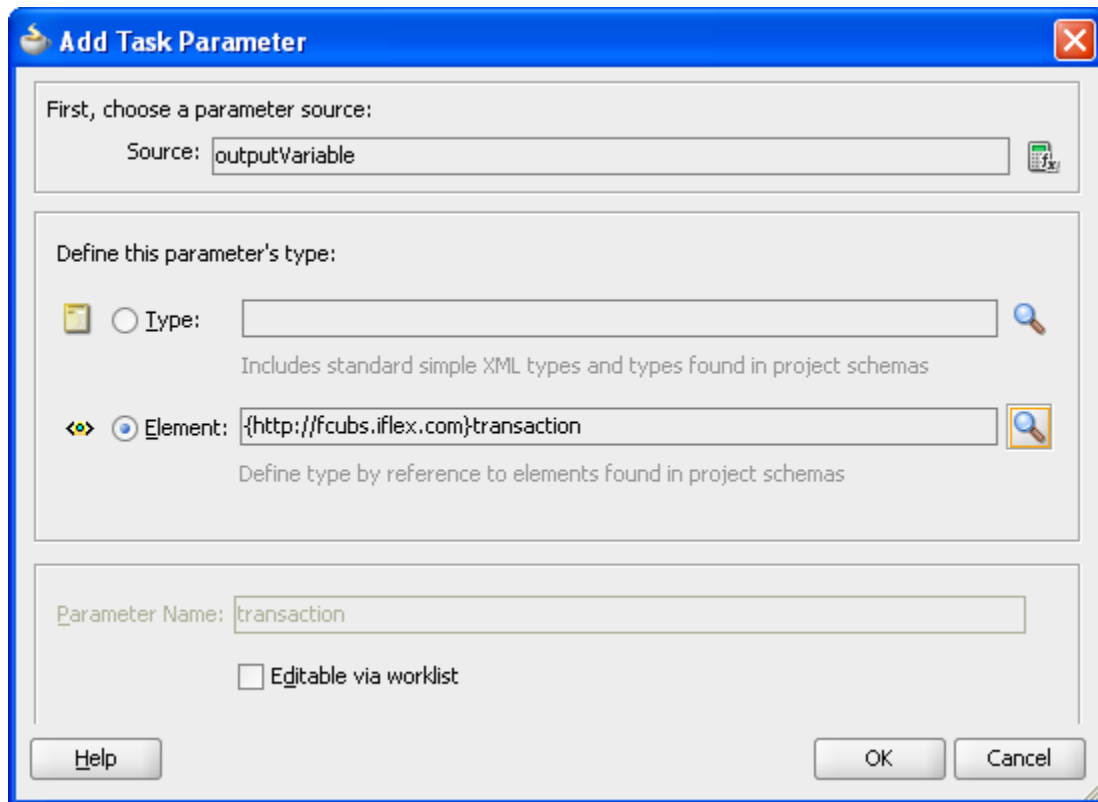
20. Put 'Reply' activity after the 'Assign'



21. Put 'output Response message' <wsdl:output message="client:BPELProcess1ResponseMessage"/> and change operation name to 'initiate' in .wsdl <wsdl:operation name="initiate">



22. Create another human task similarly



The image shows a Windows-style dialog box titled "Add Task Parameter". It has a blue title bar with a standard close button (X) in the top right corner. The dialog is divided into three main sections. The first section, titled "First, choose a parameter source:", contains a text field labeled "Source:" with the value "outputVariable" and a small icon to its right. The second section, titled "Define this parameter's type:", contains two radio button options. The first option is "Type:" with an empty text field and a magnifying glass icon; below it is the text "Includes standard simple XML types and types found in project schemas". The second option is "Element:" with a text field containing "{http://fcubs.iflex.com}transaction" and a magnifying glass icon; below it is the text "Define type by reference to elements found in project schemas". The third section contains a text field labeled "Parameter Name:" with the value "transaction" and a checkbox labeled "Editable via worklist" which is currently unchecked. At the bottom of the dialog are three buttons: "Help" on the left, and "OK" and "Cancel" on the right.

Add Task Parameter

First, choose a parameter source:

Source:

Define this parameter's type:

☐ Type:

Includes standard simple XML types and types found in project schemas

☒ Element:

Define type by reference to elements found in project schemas


Parameter Name:

☐ Editable via worklist

Help OK Cancel

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

Create Copy Operation

From

Type: Variable

variables

inputVariable

outputVariable

payload

ns1:transaction

status

taskId

ns1:txnIdentification

ns1:txnId

ns1:processName

ns1:branchCode

ns1:currentUser

ns1:txnComment

ns1:uiXml

ns1:stage

ns1:operation

☐ Show Detailed Node Information

XPath: /ns1:transaction/ns1:txnIdentification/

Help

To

Type: Variable

task:taskDefinitionURI

task:creator

task:ownerUser

task:ownerGroup

task:ownerRole

task:priority

task:identityContext

task:userComment

task:attachment

task:document

task:processInfo

task:systemAttributes

task:systemMessageAttributes

task:textAttribute1

task:textAttribute2

☐ Show Detailed Node Information

XPath: /taskservice:initiateTask/task:task/task

OK

Cancel

Create Copy Operation

From

Type: Expression

Expression:

'STDFLOW2'

☐ Show Detailed Node Information

Help

To

Type: Variable

task:systemMessageAttributes

task:textAttribute1

task:textAttribute2

task:textAttribute3

task:textAttribute4

task:textAttribute5

task:textAttribute6

task:textAttribute7

task:textAttribute8

task:textAttribute9

task:textAttribute10

task:textAttribute11

task:textAttribute12

task:textAttribute13

task:textAttribute14

☐ Show Detailed Node Information

XPath: /taskservice:initiateTask/task:task/task

OK

Cancel

Oracle JDeveloper 11g Release 1 - ProcessFlowTest1.jws : ProcessFlowTest1.jpr : D:\SKR\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
 - xsl
 - 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

BPEL - 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

Humantask2.Task...

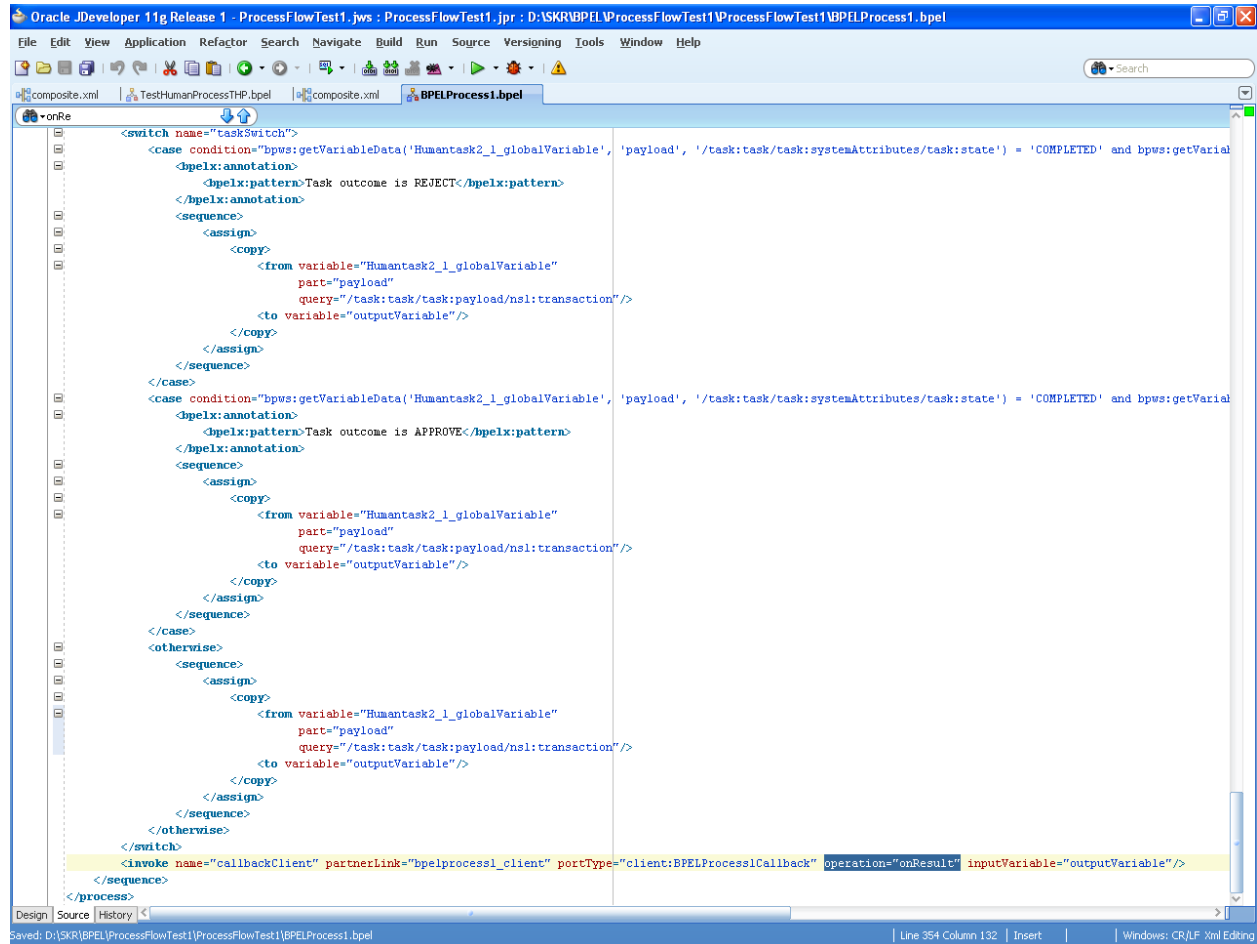
Humantask2_1_AssignTaskAttributes

initiateTask_Humantask2_1

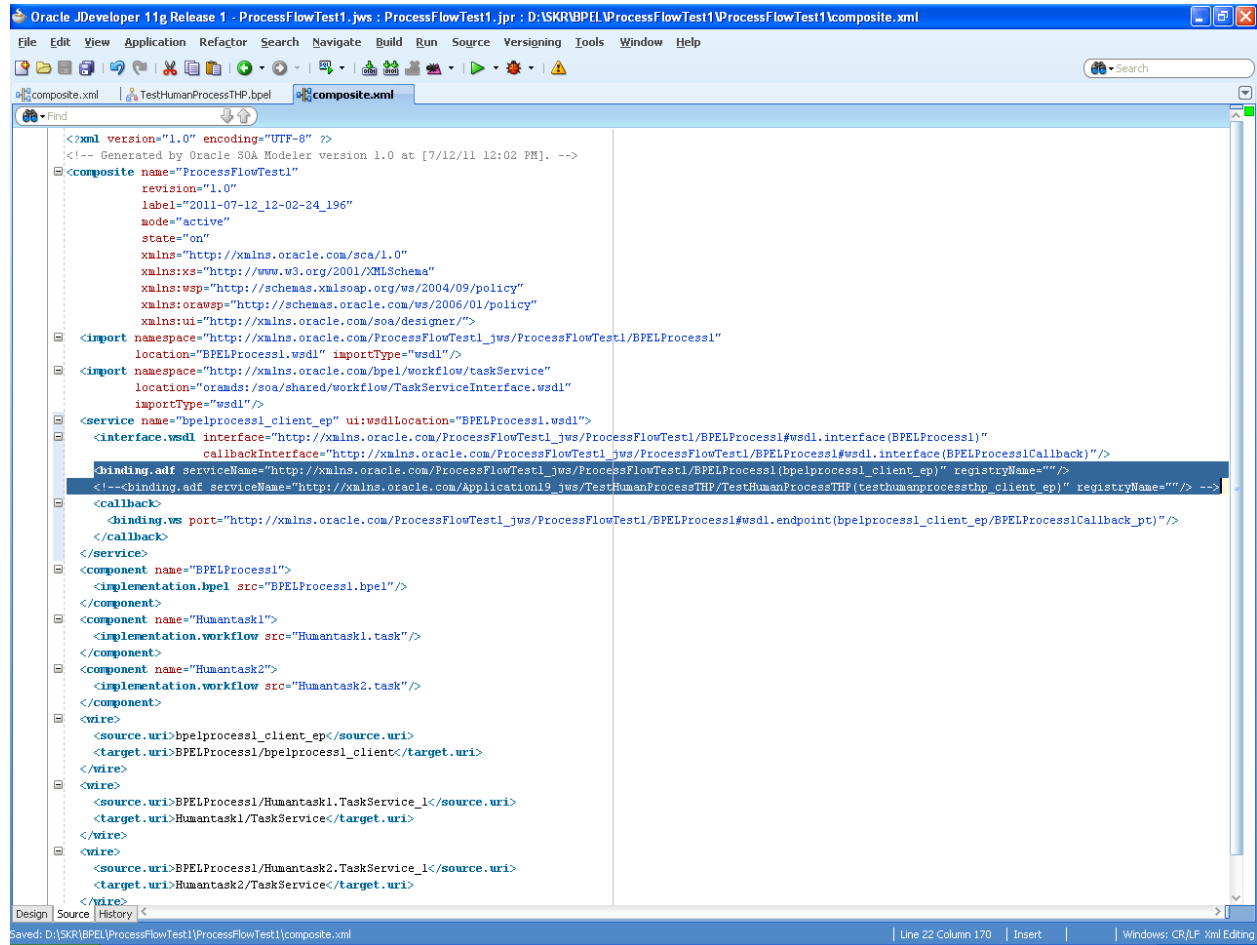
receiveCompletedTask_Humantask2_1

Save: D:\SKR\BPEL\ProcessFlowTest1\ProcessFlowTest1\BPELProcess1.wsd | BPEL Editor

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



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:orawsp="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="orawsp/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/Application19_jws/TestHumanProcessTHP/TestHumanProcessTHP(testhumanprocessth_client_ep)" registryName="" -->
      <binding ws:port="http://xmlns.oracle.com/ProcessFlowTest1_jws/ProcessFlowTest1/BPELProcess1/wsdl:endpoint(bpelprocess1_client_ep/BPELProcess1Callback_pt)"/>
    </binding>
  </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 :