Oracle FLEXCUBE BPMN Process Flow Definition Guide Oracle FLEXCUBE Universal Banking Release 14.0.0.00 [February] [2018]

FINANCIAL SERVICES

# **Table of Contents**

1.	INTRO	ODUCTION	1-2
1.	.1 B.	ACKGROUND	1-2
1.		IFFERENCE BETWEEN BPEL AND BPMN	
1.		DVANTAGES OF BPMN OVER BPEL	
1.		PMN Components	
1.		REATING NEW BPMN PROCESS	
	1.5.1	Creating a BPMN Process Model	1-8
	1.5.2	Creating the business object	1-12
	1.5.3	Creating a new process	1-15
	1.5.4	Creating the data objects	1-17
	1.5.5	Adding a ADF BC Service Adapter	1-19
	1.5.6	Adding the Created ADF-BC Adapter to the Process	1-22
	1.5.7	Assigning Inputs to the Start Node in the Process	1-24
	1.5.8	Adding a Throw Event to the Process	1-28
	1.5.9	Creating and Implementing Human Tasks	1-31
	1.5.10	How to get the Conversation Id	1-35
	1.5.11	Adding Gateways to the Process	1-40
	1.5.12	Mapping Flexcube Roles to Human Task	1-45
	1.5.13	Creating and Implementing system tasks	
1.	.6 D	EPLOYING THE PROCESS	1-58
1.	.7 R	ETAIL LENDING BPMN PROCESS	
	1.7.1	Retail Lending BPMN Process Flow Diagram	1-61
	1.7.2	Guidelines followed in Retail Lending Process flow	
	1.7.3	Naming Conventions Followed in Retail Landing Process Flow	1-62
1.	.8 A	CRONYMS AND ABBREVIATIONS	1-64
1.	.9 Ri	EFERENCES	1-64



# 1. Introduction

## 1.1 Background

This document provides a brief idea about the BPMN components and to create new BPMN process flows using the BPMN components.

## 1.2 Difference Between BPEL and BPMN

BPEL is an XML-based language for describing a business process in which most of the tasks represent interactions between the process and external Web services. The BPEL process itself is represented as a Web service, and is realized by a BPEL engine which executes the process description. BPMN is a standard set of diagramming conventions for describing business processes. It is designed to visualize a rich set of process flow semantics within a process and the communication between independent processes. It is intended to support capture of sufficient detail to allow it to be the source of an executable process description.

## 1.3 Advantages of BPMN over BPEL

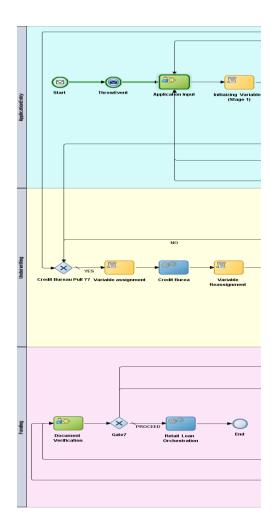
- Simple and Understandable methodology.
- BPMN provides a standard notation for modelling Business Processes that can be used and understood by all types of Business Professionals, including the business analysts who create the processes, the technical developers who create the technology that will carry out those processes and the business managers who oversee those processes.
- Business Users will be able to easily read and understand a BPMN business process diagram
- Tasks can be linked in any form (similar to "goto" in programming languages) Where as BPEL imposes more restrictions (closer to a "real" programming language).
- Structures work that needs to be done, whether this be automated or manually.
- Resembles more like a Flow Diagram (i.e) What You See is What You Get.

## 1.4 **BPMN Components**

#### Swimlanes

Swimlanes are used for grouping flow objects based on the roles defined within your process.





#### The None Start Event:

 $\bigcirc$ 

None events are always used to define the beginning of sub-processes.

The none start event cannot have incoming sequence flows. It can only have default out-going sequence flows.

#### The None End Event

The none end event is always used to mark the end of a sub-process and event sub-process.

## The Message Start Event





The message start event cannot have incoming sequence flows. Message start events require a default outgoing sequence flow.

To expose a process as a service, your process must begin with a message start event.

The message start event responds to a message sent to a specific process.

#### The Message End Event



The message end event is used to send a message to another process or service when the process is completed.

The message end event is always used with either a message start event or message catch event.

#### The Message Throw Event



The message throw event enables you to send a message to another process or service.

#### The Message Catch Event



The message catch event is frequently used with the message throw event to communicate with another BPMN process.

#### The Signal Start Event



The signal start event is a response to a signal broadcast to multiple processes.

Signals can be broadcast from a BPMN process using the signal throw event. Using a combination of signal throw and signal start events, you can invoke multiple processes simultaneously.

#### The Timer Start Event





The timer start event triggers the creation of a process instance based on a specific time condition.

### The Timer Catch Event



You can use timer event as boundary events on an activity. Timer events can be defined as either interrupting or non-interrupting boundary events.

#### The Error Catch Event



When a service or process fails with an error, the error catch event triggered. Similar to a catch in BPEL.

#### The Error End Event



The error end event "throws" an error, which can be captured in another part of the process.

#### The Terminate Event



The terminate end event is used to immediately terminate a process.

#### **Interactive Activities**

User	Process participants interact with your business application using User Tasks.
Usei	
Complex	Uses a complex routing flow that is defined within the Human Task.



FYI	Bases assignment on the participant, role, or group defined in the swimlane. Similar to the user interactive activity, but the FYI activity does not wait until completion before continuing.
Group	Uses the group vote pattern. The assignee for this automatically set to the role/group associated with the Lane. This interactive activity can only be added to swimlanes that are assigned to roles or groups.
Initiator	The initiator pattern is used to create a process instance.
Management	Uses the management chain pattern where the assignee is set to the management chain pattern for the process participant belonging to the group or role assigned to the swim lane.

## <u>The Manual Task</u>



The manual task does not allow you to manipulate data objects. Data objects associated with the previous flow element are passed through as-is to the next flow element.

### The Service Task



The service task enables you to communicate with other processes and services. Process analysts can add the service task when they know that a process must invoke an external service or process.

## The Business Rule Task





Business rules are statements that describe business policies or describe key business decisions.

## The Script Task



The script task is used to change values of data objects within your process.

It is often used to set initial values of data objects at the beginning of a process.

### The Subprocesses

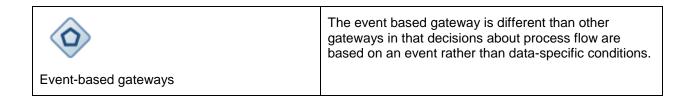


Subprocesses are contained as part of the parent subprocess. Subprocesses must begin with a start none event and must end with a none end event.

#### <u>Gateways</u>

Exclusive gateways	The exclusive gateway enables you to split your process into two or more paths. However, the process only continues down one of these paths even if multiple outgoing sequence flows are present. Exclusive gateways can have conditional outgoing sequence flows and must have at least one default outgoing sequence flow.
Inclusive gateways	The inclusive gateway enables you to split your process into two or more paths. Unlike the exclusive gateway, however, a token may flow down one or more of these paths depending on how the outgoing conditional sequence flows are evaluated.
Parallel gateways	The parallel gateway enables you to split your process into two or more paths when you want your process flow to follow all paths simultaneously. The parallel gateway is useful where your process must perform multiple tasks in parallel.
Complex gateways	The complex gateway splits a process similar to an inclusive gateway. However, it enables you to define a condition that determines if the instance can continue even if not all of the tokens have arrived at the complex gateway merge.





For more Information regarding BPMN Components Refer the Link given in the References.

## 1.5 Creating new BPMN Process

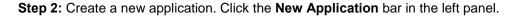
Follow the below steps to create a new process using the BPMN

## 1.5.1 Creating a BPMN Process Model

**Step 1:** Open JDeveloper Studio from the Windows Start menu. When prompted to select a role, choose the **Studio Developer**. Click OK.

👌 Select Role	x
Select the role that matches your requirements. You can also change roles using the Roles page in preferences.	
Role:	
<ul> <li>Studio Developer (All Features) Includes all features.</li> </ul>	
Customization Developer Configures the product for customizing metadata.	
Database Developer     Includes only features for core database development.	1000
Java Developer     Includes only features for core Java development.	
Java EE Developer     Includes only features for core Java EE development.	•
Always prompt for role selection on startup	
OK Can	cel

Close the Daily Tips window.





👌 Oracle JDeveloper 12c	A 14 1000	
<u>File Edit View Application</u>	Refa <u>c</u> tor <u>S</u> earch <u>N</u>	<u>l</u> avigate
9 🕫 🗄 🗒 🔊 🖗	🛈 • 🗘 • 🔒	
Applications × Application Serve	ers	-
New Application		
Open Application		

The BPM Application wizard opens. Select BPM Application in the **Application Template panel**.

New Gallery				
Q				
<u>C</u> ategories:	Items: Show All Descriptions			
Categories: Applications Connections Deployment Descriptors Deployment Profiles Diagrams Java Maven Projects UML Unit Tests 	Items:			
Contexts and Dependency Inje	Service Bus Application			
Help OK Cancel				

Name the application ex: "**Demo**" and accept the default directory for storing application files (C:\JDeveloper\mywork).



O Create BPM Application	- Step 1 of 3		×
Name your application	n		<b>F</b>
Application Name     Project Name     Project SOA Settings	Application Name:  Demo Directory:  C: \Developer \mywork\Demo Application Package Prefix:		Browse
Help	< <u>B</u> ack	<u>N</u> ext > <u>F</u> inish	Cancel

#### Click Next.

**Step 3:** Enter the Project Name ex: "**DemoProject**". Notice that **BPM** and **SOA** are selected as project technologies by default. Click Finish.

Create BPM Application	- Step 2 of 3				×
Name your project			0101010101010101	<sup>019</sup> 0101010101	5
Application Name     Project Name	<u>P</u> roject Name: Dir <u>e</u> ctory:	DemoProject C:\JDeveloper\mywork\D	Demo\DemoProject		Bro <u>w</u> se
Project SOA Settings	Project Featur BPM BPM Technol SOA Suite SOA Suite is		OA(Service Oriente	ed Architecture) i	applications.
Help		< <u>B</u> ack	<u>N</u> ext >	Einish	Cancel

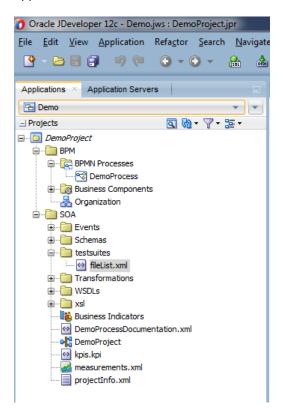
Click Next.

Step 4: Choose any composite which is required for the design.



O Create BPM Application	- Step 3 of 3
Configure SOA settin	
Project SOA Settings	Composite Name: DemoProject Start from:  Stant dromposite SOA Template Empty Composite Composite With BPEL Process Composite With BPM Process Composite With Mediator Composite With Mediator Composite With BPMN Process Composite With Case Management Composite With Spring
Help	Customizable

**Step 5:** In the upper left corner of the JDeveloper Studio window, you see the **Navigator panel**. This contains two tabs that will be important to you as you perform this tutorial: The **Application Navigator** tab and the **BPM Project Navigator** tab. Currently the Application Navigator tab is selected by default. You can see the **Demo** application appearing in the drop-down list just above the panel and the **DemoProject** appearing as the parent node within the panel. The fact that it appears in italics indicates that there are unsaved changes.



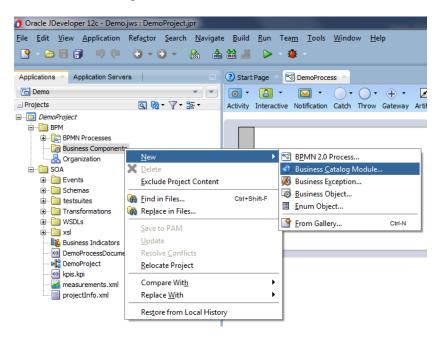


Click the Save All icon on the main toolbar.

## 1.5.2 Creating the business object

Step 6: Now you will create a business object capable of storing multiple pieces of data.

Business objects are stored in modules within the Business Catalog. In the BPM Project Navigator, expand the **DemoProject** node. Right click on Business Components and select **New** > **Business Catalog Module** 



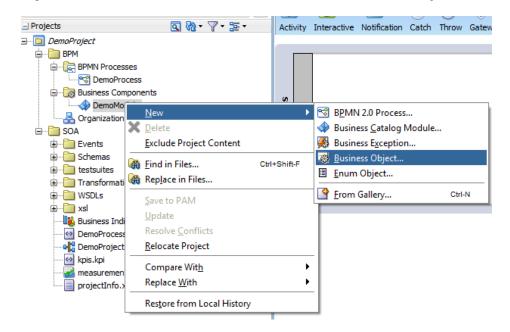
When prompted to name the new module, enter the desired name ex: "**DemoModule**" and click OK.

Create Module	
Module: DemoModule	
Help	OK Cancel

The **DemoModule** module now appears beneath the Business Catalog node.



👌 Oracle JDeveloper 12c - Demo.jws : DemoProject.jpr : C:\JD
<u>File Edit View Application Refactor Search Navigat</u>
Applications × Application Servers
🔁 Demo 👻 💌
_ Projects 💽 🖓 ▾ 🎦 ▾
DemoProject
BPMN Processes
😪 DemoProcess
🖨 📷 Business Components
DemoModule
🖮 🛅 SOA
🖶 🖳 Events
🗄 🖳 Schemas
🕀 💼 testsuites
🗄 🖓 🧰 Transformations
🕀 💼 WSDLs
i sl
Business Indicators
DemoProcessDocumentation.xml
🗠 kpis.kpi
projectInfo.xml



Right click the **DemoModule** module and select **New > Business Object**.



In the Create Business Object window, enter desired Name ex: "**BusinessObject**" and accept **DemoModule** as the Destination Module and check the check box and Click on the Search Icon.

Create Business Object	The Amplement Party of State of State	<b>— X —</b> )
Business Object		
Name:	BusinessObject	
Destination Module:	DemoModule	<b>Q</b>
Based on External Schema		9
Based on External Schen	na	OK Cancel

On click of Search Icon the following window appears, now click on the Icon at right most Corner to browse the location of the **xsd** that is to be loaded. For example **txn.xsd** in this project is loaded. After loading the **xsd**, it prompts for the copy of the **xsd** to the project. Click Ok. So that the **xsd** is copied to the project.

Note : In order to load txn.xsd file, need to plae it under schemas folder.

👌 Type Chooser	x
	2
Type Explorer     □ Project Schema Files	
i txn.xsd	
Ivpe:	
Show Detailed Node Information	
Help OK Ca	incel

Ex: C:\JDeveloper\mywork\Demo\DemoProject\SOA\Schemas



After loading the xsd, window should appear like the below and select the xsd and Click Ok.



Transaction XSD →

👌 Type Chooser	<b>X</b>
	*
Type Explorer	
Erroject Schema Files	
transaction	
	r i i i i i i i i i i i i i i i i i i i
Type: {http://fcubs.iflex.com}transaction	
Show Detailed Node Information	
Help	Cancel

Click OK again and Save all.

## 1.5.3 Creating a new process

**Step 6:** To create a **new** process within this project, first click the **BPM Project Navigator** tab. Then right click on Processes and select **New** > **Process**.



Applications × Application	tion Servers	🥐 s	tart P	age ×
🔁 Demo	• •			
Projects	■ № • ♥ • № •			JDEVELOPER
🖃 🛅 DemoProject				
🖨 🛅 BPM				
BPMN Pre=			-	
- Business	New	•	2	BPMN 2.0 Process
- 🚠 Organiza 🕽	🕻 <u>D</u> elete		-	Business Catalog Module
🖻 🛅 SOA	Exclude Project Content		<u></u>	Business Exception
🕀 📄 Events 🚍			-	Business Object
😥 📄 Schemas 🍕	Eind in Files Ctrl+Shift	t-F		Enum Object
🖨 🚞 testsuite 📢	🗑 Replace in Files			Enum Object
🗠 fileLi	Save to PAM		9	From Gallery Ctrl-N
🕀 📄 Transfor	-		-	
🕀 🚞 WSDLs	<u>U</u> pdate			Samples & Demos
🕀 📄 xsl	Resolve Conflicts			
👪 Business	Relocate Project			
DemoPre -	0 1171			
🗠 kpis.kpi	Compare Wit <u>h</u>	•		
🥁 measure	Replace <u>W</u> ith	•		
	Restore from Local History			
-			-	

In the BPM Process wizard, select the Asynchronous Service pattern.

### Click Next.

O BPMN 2.0 Process Wizar	rd	×
BPMN 2.0 Process W	izard	
Definition	Name: DemoProcess	۲
Arguments	Description:	۲
<ul> <li><u>Initial Implementation</u></li> <li><u>Advanced</u></li> </ul>		
	Directory: C:\JDeveloper\mywork\Demo\DemoProject\SOA\processes	Q
	Type:	
	S Asynchronous Service Creates a process with an asynchronous interface definition	n
	Start End	
	Synchronous Service	
Help	< Back Next > Finish Car	ncel

Since we will add the inputs later Click finish.

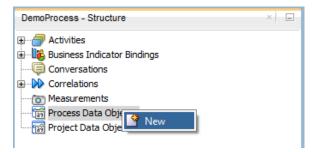


BPMN 2.0 Process Wizar	d			-	X
Arguments					<b>~</b>
<u>Definition</u> Arguments	Input Output Arguments Definition	I			<b>∔</b> ⁄×
Advanced	Name		Туре		
Help		< <u>B</u> ack	<u>N</u> ext >	Einish	Cancel

## 1.5.4 Creating the data objects

**Step 7:** When a process has been given focus, a detailed outline of its structure appears in the Structure pane in the **lower left corner** of the JDeveloper window.

Right click on Process **Data Objects** in the Structure pane and select New.



In the Create Data Object popup, enter the Name and click the ellipses button to open another window to search for complex data types.



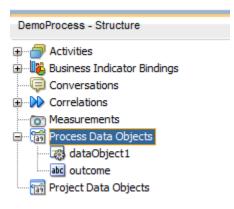
🕜 Cre	ate Data Object		×
Name:	dataObject1		
Type:	abc string		-
	✓ Auto initialize	string	
Help	)	OK	Cancel

select BusinessObject from the list of components appearing below. Click OK.

ſ	👌 Browse Types	×
	Find:	3
	999 int	-
	S boolean	
I	39E double	
	1999 decimal	
	dateTime	
	999 long	
	<>duration	
	base64Binary	
	99E float	
	999 byte	
	999 short float	
	20 date	
	20 time	
	🐯 BusinessObject	Ţ
	BusinessModule.BusinessObject	
	Help OK C	ancel

Back in the Create **Data Object** window, click OK again. The data object now appears in the **Structure pane**.

Create another process **data object** of type String to hold the **Outcome**.





## 1.5.5 Adding a ADF BC Service Adapter



Step 8: Copy the Flexcube\_interface.wsdl

wsd

to Project Location

Vame	Date modified	Туре	Size
🐌 businessCatalog	9/7/2012 4:25 PM	File folder	
📙 businessParameter	9/7/2012 4:13 PM	File folder	
📙 classes	9/7/2012 4:13 PM	File folder	
📙 config	9/7/2012 4:13 PM	File folder	
🎍 lib	9/7/2012 4:13 PM	File folder	
📔 processes	9/7/2012 5:24 PM	File folder	
li resources	9/7/2012 4:13 PM	File folder	
📔 SCA-INF	9/7/2012 4:13 PM	File folder	
isimulations	9/7/2012 4:13 PM	File folder	
📔 testsuites	9/7/2012 4:13 PM	File folder	
📙 xsd	9/7/2012 4:15 PM	File folder	
📔 xsl	9/7/2012 4:13 PM	File folder	
activityGuide.agdl	9/7/2012 4:13 PM	AGDL File	1 K
💼 composite.xml	9/7/2012 5:14 PM	XML Document	1 K
default.bpmn	9/7/2012 4:13 PM	BPMN File	5 K
DemoProcess.componentType	9/7/2012 5:24 PM	COMPONENTTYP	1 K
🗟 DemoProcess.wsdl	9/7/2012 5:24 PM	Web Service Descr	3 K
DemoProcessDocumentation.xml	9/7/2012 5:24 PM	XML Document	2 K
DemoProject.jpr	9/7/2012 4:24 PM	JPR File	17 K
Flexcube_interface.wsdl	8/21/2012 2:06 PM	Web Service Descr	2 K
measurementActions.xml	9/7/2012 5:24 PM	XML Document	1 K
🔮 measurements.xml	9/7/2012 5:24 PM	XML Document	1 K
🔮 organization.xml	9/7/2012 5:24 PM	XML Document	1 K

Now go to composite.xml drag and drop the ADF-BC Service Adapter from the Component Palette.



3) Start Page A grant and a start Page A grant and a start page A grant and a start and			Components		×
🗸 💱 🚾 🗶 🖏   🕼 🧔 🖉   🕀 🖶 🦚 🕅		DemoProject	Q.*		¢
Exposed Services	Components	External References	SOA		•
Exposes services	DemoProcess		Components BPEL Process BPEL Process Human Task Technology ADF-BC CemoProcess- Cemo	 Business Rule Spring B2B CC School School Sc	Case Case Case Case Case Subprocess BAM BAM BAM Case Case Case Case Case Case Case Case

Name the Adapter as <Process\_Name>\_client\_ep.

O Create ADF-BC Serv	ice		x
ADF-BC Service			55
Create an ADF-BC se	rvice.		
<u>N</u> ame:	RetailLending_client_ep		
<u>Type</u> :	Service 💌	_	
WSDL URL:			<u>م</u> [
Port Type:		•	
Callback Port Type:		-	
copy wsdl and its	s dependent artifacts into the project.		
Help		ОК	Cancel

Now Load the Flexcube\_interface.wsdl from the project location.



👌 WSDL Chooser	r						×		
Application Server	File System	Dracle Enterprise Repository	Project Libraries	SOA-MDS	UDDI	WSIL			
Location:	C:\JDevelop	per (mywork \Demo \D	emoProject\SO	A			) 📰 🗉		
Work	File <u>Typ</u> e: Web	terface.wsdl aube_interface.wsdl Service Definition F	iles (*.wsdl)						
Selection: file:/C:/	JDeveloper/mywo	rk/Demo/DemoProje	ect/SOA/Hexcub	e_interface.wsdl		ОК	Cancel		
🔿 Create ADF	-BC Service						×		
ADE PC Com	ADE-RC Service								

	Cicate Abr be serv		_	
	ADF-BC Service		5	
	Create an ADF-BC se	rvice.		
			_	
	<u>N</u> ame:	RetailLending_client_ep		
l	<u>Т</u> уре:	Service		
	WSDL URL:	C:\JDeveloper\mywork\Demo\DemoProject\SOA\Flexcube_interface.wsdl	۱	
	_			
	Port Type:	initate_ptt	•	
			_	
	Callback Port Type:	No Callback		
	conv wsdl and its	s dependent artifacts into the project.		
	Copy visor and its	superiorit and the project.		
	Hala		Caraa	
	Help	OK	Cance	
n i Pa				

Click OK

## 1.5.6 Adding the Created ADF-BC Adapter to the Process

Step 9: Double Click the Start Event a Property Window Appears.

e composite.xm/ × SDemoProcess ×	
	🦓   🛅 Layout   🧥 Show Warnings   🔍
Start End	
Properties - Start	
Basic Implementation	
Implementation Type: 🙆 Message	
Message Exchange	
Type: 😡 Define Interface	
Conversation: <ul> <li>Default</li> <li>Advanced</li> </ul>	
Define Interface	
Arguments Definition	<b>- +</b> <i>∕</i> ×
Name Type	
Operation Name: start	
🗱 Data Associations 🔊 Dorrelations	Log Handlers
Message Headers Service Properties	
Help	OK Cancel

Step 10: Now Change the Type as Use Interface and Click the



👌 Properties - Start	X
Basic Implementation	
Implementation Type: S Message	<b></b>
Message Exchange	
Type: 🙀 Use Interface	•
Conversation: <ul> <li>Default</li> <li>Advanced</li> </ul>	
Use Interface	
Reference:	٩, 🖉
Operation:	-
2010 Data Associations Di Correlations	Log Handlers
Message Headers Service Properties	
Help	OK Cancel

Select the Adapter from the Window

👌 Service		x
Search:		
Search Results:		
····· @ RetailLending_client_ep		
Help	ОК	Cancel

Click OK.

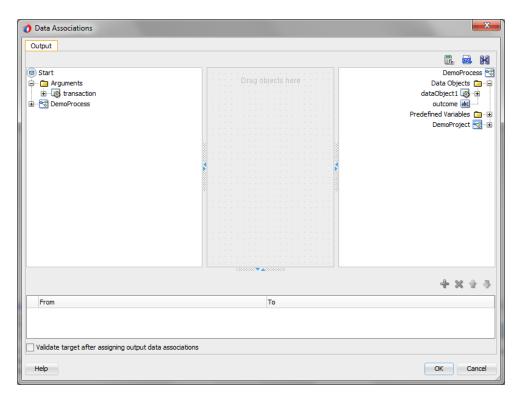


Basic Implementation		
Implementation Type:		
	essaye	
Message Exchange	-	
Type: 🐝 Use Int		
Conversation: <ul> <li>Default</li> </ul>	Advanced	
Use Interface		
Reference: RetailLending	_dient_ep	🤍 🏈
Operation: initiate		
🗱 Data Associations	D <u>Correlations</u>	Log Handlers
Dessage Headers	Service Properties	
Help		OK Can

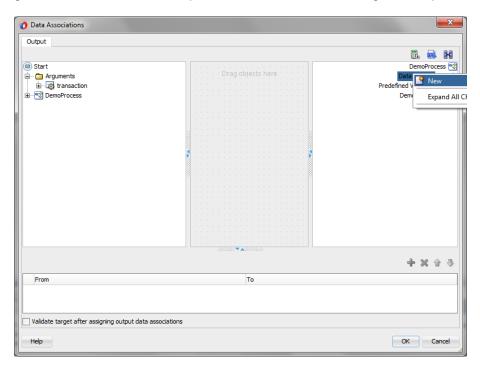
## 1.5.7 Assigning Inputs to the Start Node in the Process

Step 11: Click Data Associations to map the output for the Start node.





Right-click **Data Objects**, on RHS and add a variable **g\_Output**. **g\_Output** will be used as a global variable which will be updated in all the activities throughout the process.





👩 Cre	ate Data Object	x						
Name:	g_output							
Type:	abc string	-						
	<r>    <b< th=""><th>-</th></b<></r>	-						
	iii base64Binary							
Help	199E float							
	999 byte							

👌 Browse Types	×
Find:	
999 int         Solean         999 docimal         100 dateTime         999 long         <-> duration         11 base64Binary         12 base64Binary         12 base64Binary         13 base64Binary         14 base         15 base64Binary         15 base64Binary         16 base64Binary         17 base         18 base64Binary         19 byte         10 byte         11 byte         12 byte         13 byte         14 byte         15 byte         15 byte         16 byte	long
string Help	OK Cancel

Click OK

Now Map the Out Argument of the Start node to the g\_output global variable.



Data Associations     Output		
Start     Arguments     보급 장 transaction     DemoProcess	Drag objects here	DemoProcess 😭 Data Objects 🎦 - 🖨 dataObject I 🕃 - Ə outcome 🔐 – outcome 🔐 – Ə
		Predefined Vaniables 🔂 – G DemoProject 📆 – G
From: transaction	To: g_output	🖪 🕂 🗙 🕆 🕹
From	To 酸 g_output	
Validate target after assigning output da	a associations	OK Cancel

Step 12: Set the End Node as None by selecting implementation type as none.

Properties - End	×
Basic Implementation	
Implementation Type: None	<b></b>
Force commit after execution	
Help	K Cancel



## 1.5.8 Adding a Throw Event to the Process

**Step 13:** Drag and Drop the Message Throw Event from Events Pane in Component Palette to the Process. Or

Go to Window - > Components - > Events

Oracle JDeveloper 12c - Demojins : DemoProject.jpr : C:\/Developer\mywork\Demo\DemoProject\SOA\processes\DemoProcess.bpmn										
Elle Edit View Application Refactor Search Navigate Build Bun Team Iools Window Help										
Applications × Application Servers	(?) Start Page ×	📲 DemoProject 💉 🖻	S DemoProcess ×				Resources			
🔁 Demo 👻 💌	👩 • 🙆 •	🖂 • 🔘•	○· ⊕ · Z ·		0	Search I 📲	💁 - Q (Name			
	Activity Interacti	ve Notification Catch	Throw Gateway Artifacts		Q.	Search V				
EmoProject     BPM										
🖨 🕞 BPMN Processes			Throw None Throw Mess Throw S							
DemoProcess     DemoProcess     DemoProcess     DemoProcess										
	ess		End None End Message End En	ror						
SOA	DemoProcess		End Signal End Terminate							
🕀 🛅 Schemas	0 a	Start	End							
testsuites     Transformations	ă	Start	End							
🗑 🛅 WSDLs										
xsl										
DemoProcessDocumentation.xml										
DemoProject     Flexcube_interface.wsdl										
kpis.kpi										
measurements.xml										
pojecano.xm										
							Business Catalog			
Application Resources							E A Business Catalog E DemoModule			
🗈 Application Resources							Errors			
± Recent Files							Events			
Working Sets	A Highlight Level: V					100% 🔻 🕵 🛅 🛄	🗄 🛶 References			
DemoProcess - Structure ×		Collaboration History					Rules			
Activities			mulations Documentation			9	Types			
Business Indicator Bindings     Conversations	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	@ <sup>2</sup> •   <b>#</b>			File Lo	cation Project 🔻	🔤 👔 XsdTxnXsd			
Orrelations	- Description					industri indject				
Measurements     Process Data Objects										
dataObject1										
g_output			Success! Buil	l completed with 0 errors, 0 warnings, (	infos					
Project Data Objects										
	-	-								
	Live Issues: De	moProcess Build ×								
te <u>B</u> uild <u>R</u> un Tea <u>m</u> <u>T</u> ools <u>W</u> indow <u>H</u> elp										
a 🚵 🚜 🕞 - 🌞 -						0	<b>₹</b> (Search			
						Comments in Drawn				
Start Page ×      Of DemoProject ×      DemoProces						Components × Resources				
· · · · · · · · · · · · · · · · · · ·				Q Search		Q*(				
Activity Interactive Notification Catch Throw Gatew	ay Artifacts					BPM Analytic				

690				$\smile$ $\lor$					Q Search	) 6				
Activity	Interactive	Notification	Catch 1	Throw Gateway	<ul> <li>Artifact</li> </ul>	cts					BPM Analytic			
											Activities			
											Events	message	None	Signai
											٥			
SS											Timer	_		
ē				$\cap$							Throw Events			
DemoProcess	L '	<b>—</b>	,									$\bigcirc$		
Der		Start		End							Message	None	Signal	
											Start Events			
												$\bigcirc$		(0)
											Message	None	Signal	Timer
											End Events			
											R		$\bigcirc$	
											Error	Message	None	Signal
											0			



A Properties window appears. Now select the implementation as **Use Interface** and type as Synchronous. The Throw Event will be automatically implemented.

🕐 Properties - ThrowEvent	
Basic Implementation	
Implementation Type: 🔘 Message 👻	
Force commit after execution	
Message Exchange	
Type: 🙀 Use Interface 🔻	
Conversation: <ul> <li>Default </li> <li>Advanced</li> </ul>	
Use Interface	
Type: Asynchronous O Synchronous	
Reference:	
Operation:	
Error:	
A Data Associations Data Associations	
Message Headers 🔠 Service Properties	
Help OK Cancel	

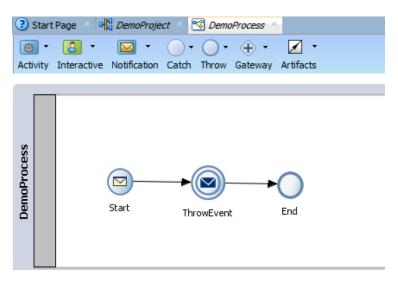
Select the reference as RetailLending\_Client\_ep.

👌 Properties -	ThrowEvent	×
Basic Implem	nentation	
Implementation	Type: 💿 Message	•
Force commi	it after execution	
Message Exch	ange	
Type:	🙀 Use Interface	•
Conversation:	: <ul> <li>Default  </li></ul> <li>Advanced</li>	
Use Interfac	æ	
Type:	<ul> <li>Asynchronous </li> <li>Synchronous</li> </ul>	
Reference:	RetailLending_client_ep	Q 🔗
Operation:	initiate	•
Error:		S. Ø
and Data Associ	iations DD Correlations	Log Handlers
I Message He	eaders Service Properties	
Help		OK Cancel



### Click OK.

Now our Process Look Likes this





## 1.5.9 Creating and Implementing Human Tasks

**Step 14:** Expand the Activities pane in the Component Palette and from the Interactive section, click and drag a User activity, dropping it onto the sequence flow between Throw and End events.

Components ×	Resources		
Q.*			(
BPM Analytic			
Activities			
Activity	Business Rule	Call	Event Subprocess
		I	<b>1</b>
Manual	Receive	Script	Send
<b>(#</b> )	<b>E</b>	<b>2</b>	
Service	Subprocess	Update	
Interactive			
<b>2</b> *	<b>.</b>	<b>_</b> ₽⇔	
Complex	FYI	Group Vote	Initiator
•2	<u>a</u>		
Management	User		
Notification			
			2
TM	Mail	SWS	Hear

The **user task** properties window will be opened. In the basic tab enter an appropriate name and then go to **implementation tab** and click add Icon of the **Human Task** column.



👌 Properties - UserTa	isk	x
Basic Implementation	n	
Implementation Type:	🙆 User Task	-
Human Task:	• • •	. 🥔
😑 Human Task Attribu	tes	
Title: 📓 Plain Te	ext 💌	f <sub>x</sub>
Priority: 📃 Literal	•	) 🥢 🛛
Re initiat	te	
Advanced		
Stata Associations	D <u>Correlations</u> <u>Log Handlers</u>	
Help	OK C	ancel

Create **human task** window is opened, Change the Name and title accordingly, Click on the add Icon and map **parameter** and **outcome** target respectively and Check the editable field in the parameter slab.



	🕐 Create H	Human Tas	k		X	👌 Browse Data Objects
	General					Drag Data Objects to Parameters table and Outcome Target
	Name:	Application	Entry	Pr	iority: 3 (normal)	field (or use CTRL+I/CTRL+O).
H	Title:	Application	Entry			
l	Outcomes:	APPROVE,R	EJECT		Q,	
	Pattern:	Simple	•			😪 DemoProcess 白… 🛅 Data Objects
		Performer:	<ul> <li>Ourrent lan</li> </ul>	e participant 🔵 Pre	evious lane participant	🕂 🗤 🙀 dataObject1
			Exclud	le previous participan	ts	E gitteme 
			Exclud	e previous participan	it in current task	E Predefined Variables
000	Parameters	:		+ ×		DemoProject
	Parameter		Name	Туре	Editable	
	transaction		dataObject1	Der module.Busin	ie 🗸	
	Outcome ta	rget:	outcome	/	_ + <i>4</i>	Help
	Help				OK Cancel	

Click OK and click on Data Associations in the User Task properties window.

🕜 Properties - UserTask	x
Basic Implementation	
Implementation Type: 🔲 User Task	•
Human Task: 🔄 ApplicationEntry	<b>/</b>
Human Task Attributes	
Title: 🔊 Plain Text 🔻	
Priority: 🔲 Literal 🔻	<i>«</i>
Re initiate	
2010 Log Handlers	
Help OK Can	cel



Data Associations     Input Output		×
R DemoProcess		🗓 📠 🕅 UserTask 🖪
Image: Second state of the second	Drag objects here	execData
		🖬 👍 🗙 🔶 🎩
From From dataObject1 From dataObject1	To: transaction To	[], + × +      [],     [], + × +      [],
Validate target after assigning input data as	sociations	OK Cancel

Click Ok in **User Task** properties window.

Role properties window appears.

Role properties	
Name:	Process Owner 🔻 🗣
Translucent background:	<b>4</b>
Image:	۹. 🏈
Help	OK Cancel

Add New Name

Role			×
Name:	Application Input		
Help		ОК	Cancel

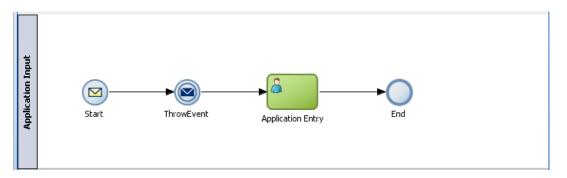


► Role			<b>x</b>	
Name:	pplication Input			22
Help		ОК	Cancel	New
_	Translucent background:	<i>\</i>	_	
	Help		ОК	Cancel

Add a Name to the **Role** Click Ok and Save all.

Role properties	<b>x</b>
Name:	Application Input 🔹 🕂
Translucent background:	A 100 A 1
Image:	۹. 🏈
Help	OK Cancel

Now our Process Look Likes this:



## 1.5.10 How to get the Conversation Id

To get the infra generated conversation id in the process, the below steps needs to be performed in the **first human task** of the process.

This conversation id is considered as **Application Number in each process**. Conversation id will be **unique** for each task.



**Step 1:** On double click on the first human task, you will get the below property window.

In property window, go to the implementation tab and click on the highlighted data association.

1 Properties - UserTask
Basic Implementation
Implementation Type: [ 🙆 User Task 🔹
Human Task: 🛛 🔁 ApplicationEntry
Human Task Attributes
Title: 🔊 Plain Text 🔻
Priority: 🔲 Literal 🔍 🗸 🗸
Re initiate
हैन्द्र <u>Data Associations</u> 🖉 🕪 <u>Correlations</u> 🔳 <u>Log Handlers</u>
Help OK Cancel



Data Associations     Input Output		
apper output		B. 📾 K
SemoProcess         →       Data Objects         →       #         →       #         →       #         →       #         →       #         →       #         →       #         →       #         #       -         #       #         #       -         #       -         #       -         #       -         #       -         #       -         #       -         *       -         *       -	Drag objects here	ting was the UserTask (2) Arguments (2) transaction (2) execData (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)
From: dataObject1	To: transaction	🔜 🕂 🗙 🕆 🦑
From	То	
ataObject1	🧱 transaction	
Validate target after assigning input data associa	tions	
Help		OK Cancel

On clicking the data association, the below input/output window will be opened.

**Step 2:** Drag the highlighted expression icon to the target (**txnld**) then Expression builder window will get opened.

Input Output		
월 DemoProcess - Data Objects - 문화 Data Objects - 문화 Data Objects - 문화 Data Objects - 문화 Data Object - 문화 DemoProject	Drag objects here	UserTask @ Arguments
rom: dataObject1	To: transaction	🔜 🕂 🗙 🕆 🕄
From	То	
🖥 🥵 dataObject 1	🐯 transaction	



Expression builder window.

Txpression Builder	
Mode: 📑 Simple Exp. 🕶	
Build an expression by typing directly into the Expression field and/or inser	t fragments from the fragment editors below the Expression field.
Expression:	<u> </u>
∧ Insert In	-
Variables	Functions
S DemoProcess	string
🗄 🐨 🐻 dataObject1	f() contains
	<u>f()</u> endsWith
E Predefined Variables	<u>f()</u> length
E	<u>f()</u> lowerCase
	f() startsWith
	f() substring
	f() substring
Content Preview:	
contains(value, s)	
Description:	
contains(string, string) : boolean	â
This function returns true if the first argument string contain false.	s the second argument string, and otherwise returns
Help	OK Cancel

Change Mode to XPath Exp.

Add the "ora:getConversationId()" function from the BPEL Xpath Extension Function list



Txpression Builder	
Mode: 🏪 XPath Exp. 💌	
Build an expression by typing directly into the Expression field, using Ctrl+ fragment editors below the Expression field.	Space for XPath assistance if available, and/or insert fragments from the
Expression:	Ø 🕲 🗋
ora:getProcessId()	
\land Insert In	to Expression
Variables	Functions
DemoProcess     DemoProcess     DemoProcess     DemoProcess     DemoProcess     DemoProcet	BPEL XPath Extension Functions       f(i) integer       f(i) getProcessVersion       f(i) getProcessVersion       f(i) getProcessURL       f(i) getProcessId       f(i) getProcessId       f(i) getPreference       f(i) netWodes
Content Preview:	
ora:getProcessId()	
Description: This function returns the id of the current BPEL process. The signatu	
Help	OK Cancel

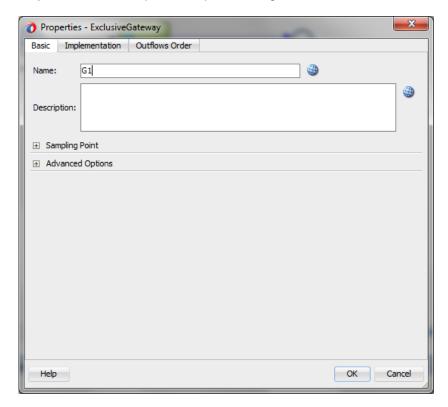
Finally Expression will be added in data association window.

Data Associations		
Input Output		
😴 DemoProcess		UserTask 🖪
🖨 🗁 Data Objects		Arguments 🛅 … 🚊
🗉 🐻 dataObject1		transaction 🗃 🖃
abc outcome		txnIdentification 🔇
🗈 🐻 g_output		txnId abc
🖿 🧰 Predefined Variables		instanceId and
🗠 😽 DemoProject		processName abc
		branchCode 🔤 🔤
		moduleCode and
		currentUser and
		txnComment 🔤
		uiXml 🔤 ·····
		stage and
		taskOutcome 🔤 ·····
		operation and
		realm abc
		transactionData 🊷 🗄
		additionalFields 🊷 🗄
		documents 🍫 🗄
rom: ora:getProcessId()	To: transaction.txnIdentificat	ion.txnId 📃 🖳 🕂 🗙 🎓 🗸
From	То	
dataObject1	transaction	
, ora:getProcessId()	abc transaction.txnIdentificatio	n.txnId
Validate target after assigning input data	associations	
Help		OK Cancel



## 1.5.11 Adding Gateways to the Process

Step 15: Expand the **gateway pane** in the **Component Palette** and click and drag a **Exclusive gateway**, dropping it onto the sequence flow between **Appenty**(humantask) and **End events**.



Properties window opens if required change the name and click OK.

Step 16: Expand the Events pane in the Component Palette and from the End events section, click and drag a Terminate, dropping it onto the Process editor.

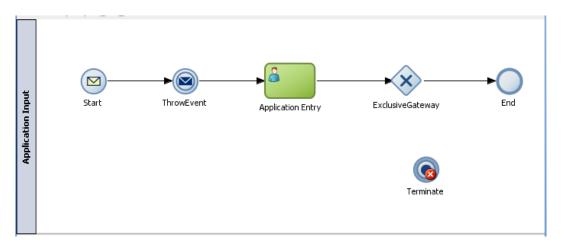
Properties window opens if required change the name and click OK.



👌 Propertie	s - End1	×	
Basic Imp	lementation		
Name:	End1		
Description:		۲	
Is Draft:			
+ Advance	d Options		
Help		OK Cancel	

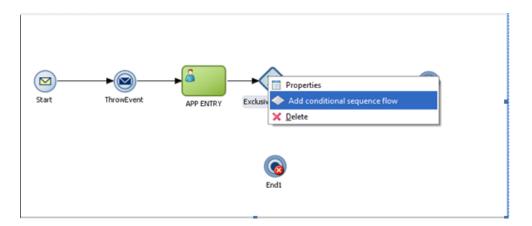


Now our Process looks likes this:

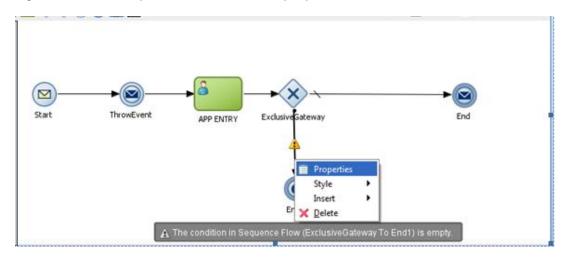


Right click on the **gateway** and select the **Add Conditional Sequence** flow, and connect the gateway to the **Terminate event**.





Right click on the Sequence flow, and select properties.



The **Sequence flow property** window will be opened, enter the name if required and go to properties tab, Click on the Expression Builder.



Transition from Activity: 'Exclusive Gateway' to Activity: 'Terminat Description Properties	e' 🔍
Туре	
Condition	
(	
xpression:	
) Simple Exp. () XPath Exp.	
Help	OK Cancel

And Build the condition for the sequence flow by selecting the object from the list and click **insert into expression** or by dragging and drop the **object** in the expression tab.

Txpression Builder	
Mode: 📝 Simple Exp. 🔻	
Build an expression by typing directly into the Expression field and/or insert	fragments from the fragment editors below the Expression field.
Expression:	S 🕲 🗋
outcome="CANCEL"	
A Insert Int	
Variables	Functions
S DemoProcess	string
🕀 📲 😨 dataObject 1	f() contains
	f() endsWith
Emilia g_output	f() length
🗄 😽 DemoProject	f() lowerCase
	f() startsWith
	f()     substring       f()     substring
Content Preview:	
Description:	
Data Object	
Help	OK Cancel
пер	UK Cancel

Click Ok and again Click OK.

Click Save All.

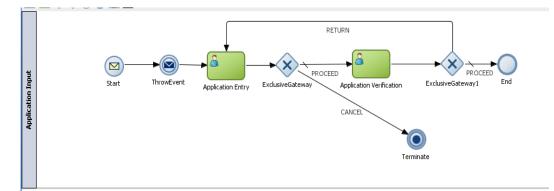


Create another user activity and implement the humantask properties.(follow the same steps as done for the useractivity (**Application Entry**)).

Create another gateway event and connect the conditional end to the first human task(app entry).

Click Save all.

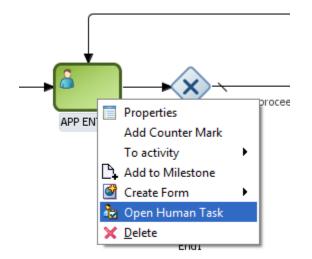
Now the process looks like this:



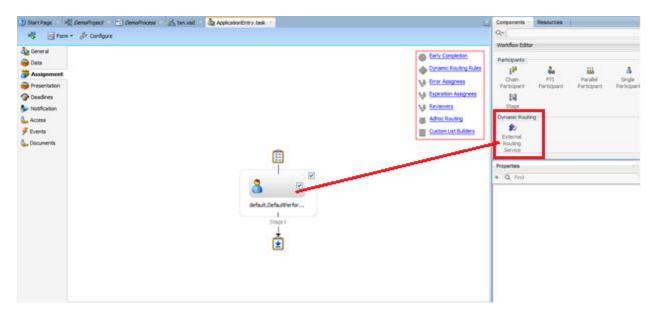


## 1.5.12 Mapping Flexcube Roles to Human Task

Step 17: Right click on the human task and select Open Human Task, it opens in new tab.



Click Assignment tab and click Edit Icon.



Drag and drop the External Routing Service as mentioned in above figure.



🕐 Use External Routing	
Participants and routing defined by external service Fully qualified name of class used for External Rout Cass Name:	e that dynamically determines the participants in the workflow ting
Define Properties that will be used with the routing	service 🕂 💥
Name	Value
Help	OK Cancel

now enter the class name as (com.ofss.fcc.bpel.cac.FCBPELTaskAssignmentComponent).

👌 Use External Routing		×
Fully qualified name of class used fo	external service that dynamically determin r External Routing .FCBPELTaskAssignmentComponent	es the participants in the workflow
Define Properties that will be used v	with the routing service	+ ×
Name	Value	
Help		OK Cancel

Enter the **name** and select by **expression** from the select box and click on +beside it.

<sup>100</sup>The name specified here is Case sensitive.



Use External Routing	
Fully qualified name of class used	y external service that dynamically determines the participants in the workflow for External Routing ac.FCBPELTaskAssignmentComponent
Define Properties that will be use	+ ×
Name functionId	Value By Expression
Help	OK Cancel

Select the stage from the task:payload and Click insert into expression and Click OK.

Expression Builder	×
Build an expression by typing directly into the Expression field, using Ctrl+5 fragment editors below the Expression field.	Space for XPath assistance if available, and/or insert fragments from the
Expression:	S 🕲 🗋
/task:task/task:payload/ns1:transaction/ns1:txnIden	tification/n#1:stage
🔥 Insert Int	o Expression
Schema	Functions
<pre></pre>	String Functions
Content Preview: /task:task/task:payload/ns1:transaction/ns1:txnIdentification/ns1:stage	
Description:	
Help	OK Cancel

Add another element and likewise map the branchcode and click ok .



👌 Use External Routing	×
Fully qualified name of class used	cac.FCBPELTaskAssignmentComponent
Name	Value
functionId	/task:task/task:payload/ns1:transaction/ns1:txnIdentificati
branchCode	By Expression 🔻 1s1:txnIdentification/ns1:branchCode
Help	OK Cancel

Click ok

Click Yes and Save all.

Repeat the same steps for every human task in the process.



## 1.5.13 Creating and Implementing system tasks

Step 18: Drag and Drop BPEL Process to the Composite.xml

P A Service Call can be **Implemented** by **Service Adapters** also.

Components ×	Resources		
Q.+ (			
SOA			
Components			
2			2
BPEL Process	BPMN Process	Business Rule	Case Management
යික	-Frocess		
uw Human Task	Mediator	Spring	Subprocess
Technology	Ficalator	oping	Supprocess
echnology 🌦	÷	<b>1</b>	<b>1</b>
ADF-BC	۳ <b>۵۵</b> AQ	B2B	чш ВАМ
*	(iii)	<u> </u>	8
Coherence	ય≣ Database	<del>دی</del> Direct	EJB
	563	87	5
File	FTP	Healthcare	нттр
÷Čě,	췝	ැ	
JMS	LDAP	MFT	MQ
÷	1	<b>\$</b>	<b>6</b>
MSMQ	REST	SOAP	Socket
UMS			
Applications -			

Drag and drop the BPEL process in the composite.xml as follows.



3 Start Page × Page → Page DemoProject × S DemoProcess × B tor	n.xsd ×		Components ×	Resources
🖌 🕼 🖶 👘 🙆 🔯 I 🕅 🦉 🖉 🖓		DemoProject	Q.*	
Exposed Services	Components	External References	SOA	
RetailLending_client Operations: intilate	Process		Components BPEL Process BPEL Process Break Technology ADF-BC Coherence	BPMN Process Mediator AQ Batabase
			الله MS الله	CDAP
			MSMQ UMS Applications	REST

A Window appears Rename the Process with Desired name ex: RetailLoanOrchestrationBPEL.

💩 Create BPE	L Process
	s cess is a service orchestration, based on the BPEL specification, used to describe/execute a ocess (or large grained service), which is implemented as a stateful service.
BPEL 1.1 Sp	pecification O BPEL 2.0 Specification
<u>N</u> ame:	RetailLoanOrchestrationBPEL
Namespace:	http://xmlns.oracle.com/Demo/DemoProject/RetailLoanOrchestrationBPEL
Template:	😹 Asynchronous BPEL Process 🔹 🥥
Ser <u>v</u> ice Name:	retailloanorchestrationbpel_client
	Expose as a SOAP service
	Input: [http://xmlns.orade.com/Demo/DemoProject/RetailLoanOrchestrationBPEL}process
	Qutput: [xmlns.oracle.com/Demo/DemoProject/RetailLoanOrchestrationBPEL}processResponse]
Help	OK Cancel

Click the **Input** and **Output** and chose the element type.

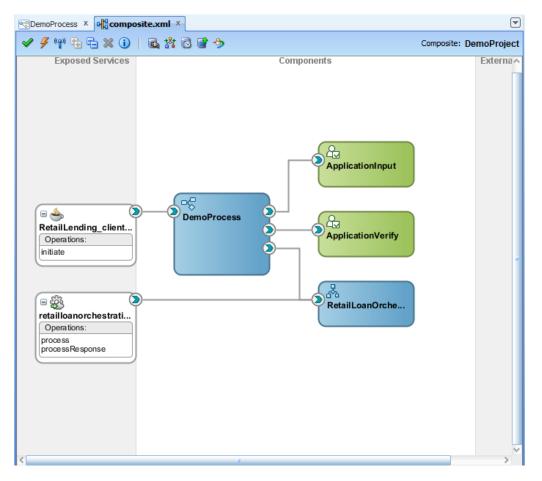


👩 Type Cho	oser	Statements and			×
					2 6
	ect Schema ApplicationE ApplicationE ApplicationV ApplicationV TaskStateM TaskStateM txn.xsd ect WSDL Fi	EntryPayload.xsd EntryWorkflowTask.xsd /erificationPayload.xsd /erificationWorkflowTas lachine.xsd			
Type:					
	ailed Node I	Information			
Help				OK	Cancel
					)
👌 Create BPEL P	rocess				
BPEL Process					
		rchestration, based on the BP vice), which is implemented as		d to describe/ex	ecute a business
O BPEL 2.0 Speci	ification 💿 BP	PEL 1.1 Specification			
General In M	lemory SOA				
News					
<u>N</u> ame:	RetailLoanOr	rchestrationBPEL			
<u>N</u> ame: Name <u>s</u> pace:		.oracle.com/Demo/DemoProje	ct/RetailLoanOrches	trationBPEL	
	http://xmlns.			trationBPEL	9
Namegpace:	http://xmlns. C:\JDevelope	.oracle.com/Demo/DemoProje	t\SOA\BPEL	trationBPEL	Q
Namespace:	http://xmlns. C:\JDevelope	.oracle.com/Demo/DemoProje er \mywork \Demo \DemoProjec	t\SOA\BPEL	trationBPEL	•
Name <u>s</u> pace: Directory: Template Type:	http://xmlns. C:\JDevelope : <u>W</u> eb Ser	.oracle.com/Demo/DemoProje er \mywork\Demo\DemoProjec vice <u>R</u> EST Service !	t\SOA\BPEL	trationBPEL	
Namespace: Directory: Template Type: Template:	http://xmlns. C:\JDevelope @ Web Ser @ Asynchro retailloanord	.orade.com/Demo/DemoProje er\mywork\Demo\DemoProjec vice BEST Service [ onous BPEL Process	t\SOA\BPEL	trationBPEL	
Namespace: Directory: Template Type: Template:	http://xmlns. C:\JDevelope @ Web Ser @ Asynchro retailloanord	.orade.com/Demo/DemoProje er \mywork\Demo \DemoProjec vice REST Service ! onous BPEL Process hestrationbpel_client	t\SOA\BPEL	trationBPEL	
Namespace: Directory: Template Type: Template:	http://xmins. C:\JDevelope : OWeb Ser Resultionard retailloanord Y Expose as	.orade.com/Demo/DemoProje er\mywork\Demo\DemoProjec vice <u>REST</u> Service ! onous BPEL Process hestrationbpel_client s a SOAP service	t\SOA\BPEL	trationBPEL	• @
Namespace: Directory: Template Type: Template:	http://xmlns. C:\JDevelope @ Web Ser @ Asynchro retailloanord @ Expose as Delivery: Transaction:	.orade.com/Demo/DemoProje er\mywork\Demo\DemoProjec vice <u>REST</u> Service ! onous BPEL Process hestrationbpel_client s a SOAP service	t\SOA\BPEL	trationBPEL	• @
Namespace: Directory: Template Type: Template:	http://xmlns. C:\JDevelope ③ Web Ser Fetailloanord Fetailloanord Expose as Delivery: Transaction: Input: (htt	.orade.com/Demo/DemoProje er \mywork\Demo \DemoProjec vice BEST Service ! onous BPEL Process hestrationbpel_client s a SOAP service async.persist	t\SOA\BPEL No Service	trationBPEL	▼ @  _ 0
Namespace: Directory: Template Type: <u>T</u> emplate:	http://xmlns. C:\JDevelope ③ Web Ser Fetailloanord Fetailloanord Expose as Delivery: Transaction: Input: (htt	.orade.com/Demo/DemoProjec er \mywork\Demo\DemoProjec vice <u>REST Service !</u> onous BPEL Process hestrationbpel_client s a SOAP service async.persist tp://fcubs.iflex.com}transacti	t\SOA\BPEL No Service	trationBPEL	• @ • @ • @

Click Ok

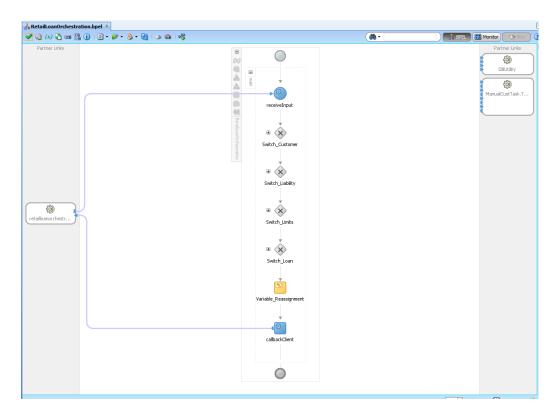
Now the Composite.xml looks like this





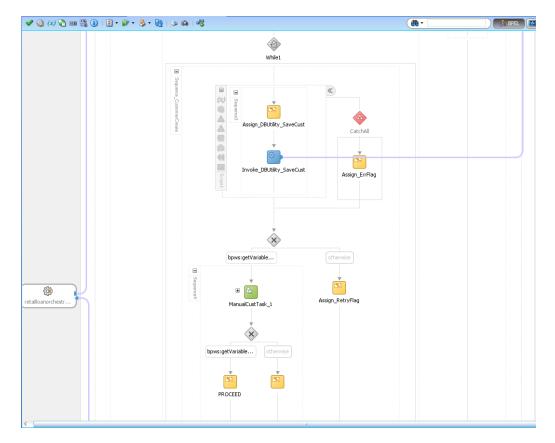
The RetailLoanOrchestration.bpel looks like this after Implementation.





In each Switch node of RetailLoanOrchestration.bpel process DButility Bpel process is called if it fails a manual retry task is initiated to book the RetailLoan





Now go to the Process and add a Service Task from component palette.



🛗 Component Pale 🗴 🙀 🗴 🔔
врм
ee 📀
– Default
Activity
🚓용 Business Rule
Call
Event Subprocess
C Manual
Receive
[ Script
🛃 Send
🛞 Service
Subprocess
- Interactive
Complex Complex
🛃 FYI
Comp Group
🔁 Initiator
Panagement
3 User

Window appears as below:

Properties - ServiceTask		×
Basic Implementation		
Implementation Type: 👩 Ser	vice task	•
Force commit after execution	n	
Message Exchange		
Type: []] Not Implemented		•
8X Data Associations	DD <u>Correlations</u>	Log Handlers
* Message Headers	Service Properties	
Help		OK Cancel
<u>c</u>		

Now Select type as Service Call and click the



Properties - ServiceTask		
Basic Implementation		
Implementation Type: 👸 S	ervice task	•
Force commit after execu	tion	
Message Exchange		
Type: 🥵 Service	Call	▼
Conversation: <ul> <li>Default</li> </ul>	Advanced	
Service Call		
Service:		🏈
Operation:		-
Stata Associations	DD <u>Correlations</u>	Log Handlers
* Message Headers	Service Properties	
Help		OK Cancel

Window appears as below from that select the service **RetalLoanOrchestrationBPEL** which is listed.

👌 Properties - S	erviceTask	23
Basic Impleme	👌 Service 📃	
Implementation 7	Search:	•
Force commit	Search Results:	
Message Excha	·····	
Type:		-
Conversation:	1 1	
Service Call		
Service:		🔍 🧳
Operation:		
ेद्र <u>Data Associa</u>		<u>s</u>
I Message He		
	Help OK Cancel	
Help		Cancel

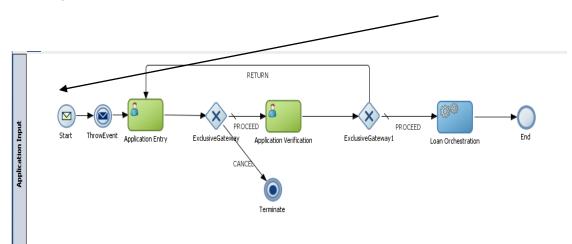
Click Data association



State Objects       Drag objects here         Image: State Objects       Drag objects here         <	D. 📾 H
	ServiceTask ( Arguments - trensaction 🖓 - 🔅
From: dataObject1 To: transaction	🔜 🕂 🗙 🕁 🗏
From To	
ataObject1 🐉 transaction	

Map the **data association** and click ok.

Now the process looks like this:

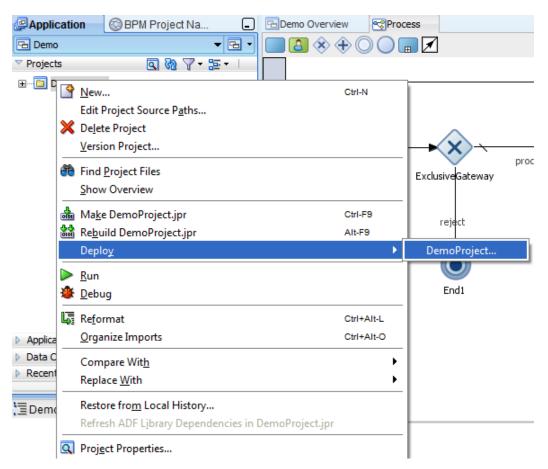




# 1.6 **Deploying the Process**

Step 19: Deploy the **DemoProject**.

In the Application Navigator, right click DemoProject and select Deploy > DemoProject...



The Deploy **DemoProject** wizard opens.

In the **Deployment Action** page of the wizard, select Deploy to Application Server and click **Next**.



😁 Deploy DemoProject	
Deployment Action	01010101010101010101010
Deployment Action	Select a deployment action from the list below. Deploy to Application Server Deploy to SAR
	Deploy this archive to SOA configured Application server(s)
Help	< Back Next > Einish Cancel

In the **Deploy Configuration** page, click the **Overwrite any existing** composites with the same revision IDcheckbox and click Next.

🕹 Deploy DemoProject	
Deploy Configuration	01010101010101010101
Deployment Action	PHO DemoProject
Deploy Configuration	Composite Revision ID
Select Server	Project: DemoProject
Ó Summary	Current Revision ID: 1.0
	New Revision ID: 1.0
	Mark composite revision as default.
	Overwrite any existing composites with the same revision ID.
<	Use the following SOA configuration plan for all composites: Browse
Help	< <u>Back</u> <u>Next</u> <u>Finish</u> Cancel



Click Next.

In the Select Server page, select **server**(**RUNNING WITH SOA**) and click Finish. Deployment will begin.

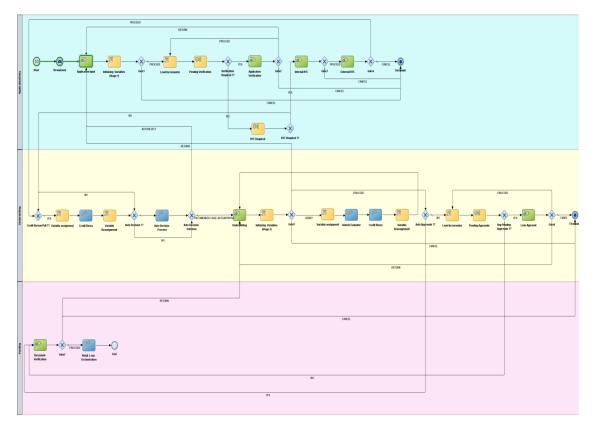
👌 Deploy DemoProject		x
Select Server	01010101010101010101010	5
Deployment Action     Deploy Configuration     Select Server     SOA Servers     Summary	Application Servers: arun IntegratedWebLogicServer (domain unconfigured) ARUN nish SOARND1 localhost V Overwrite modules of the same name	
Help	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Car	icel

Click Finish.You will see a message in the jdeveloper as **Deployment Finished**.



## 1.7 Retail Lending BPMN Process

## 1.7.1 Retail Lending BPMN Process Flow Diagram



## 1.7.2 Guidelines followed in Retail Lending Process flow

txn. xsd

single data object named transactioninput which contains the transaction.xsd

is used throughout the Process Flow to map the Inputs and Outputs.

- Global Fault Handling Standard are maintained across the Process Flow.
- The Fault Handling is done separately in another BPEL Process called **RetalLoanOrchestrationBPEL**.
- Swimlanes are used to differentiate the different Stages in the Process.



- Usage of Flexcube\_interface.wsdl using other wsdl files.
   to initiate the Process rather than
- **com.ofss.fcc.bpel.cac.FCBPELTaskAssignmentComponent** class is used in all Human Tasks to map the Flexcube Roles .
- Naming Conventions are followed as per the Retail Lending Flow Diagram.
- The DBUtility Call and Manual Retry Task are done separately in RetalLoanOrchestrationBPEL BPEL Process.



## 1.7.3 Naming Conventions Followed in Retail Landing Process Flow

#### **Processes**

- RetailProcess(BPM)
- RetailLoan OrchestrationBPEL(BPEL)

#### RetailProcess(BPM)

#### Human Tasks:

- ApplicationInput
- ApplicationVerify
- InternalKYCTask
- ExternalKYCCkeck
- UnderWriting
- UnderWriting\_Approval
- FinalVerification

#### **Business Rules:**

- VerifyAppRules
- KYCRules
- LoanApprovalRules

#### **Exposed Services:**

• RetailLending\_client\_ep

#### **External Reference:**

- VehicleEvaluater
- CreditBureau
- SelectDecisionDBAdapter

Task Name/Event Name	Input	Output
Start Event	-	transactioninput
ThrowEvent	transactioninput	-
User Tasks(all )	transactioninput	transactioninput
VerifyAppRules	VRule_IN	VRule_OUT
KYCRules	KYC_IN	кус_оит



LoanApprovalRules	UnderWrite_IN	UnderWrite_ OUT
Credit Burea	ExCreditBureau_IN	ExCreditBureau_OUT
Auto Decision Process	AutoDecision_IN	AutoDecision_OUT
Vehicle Evaluater	VEvaluator_IN	VEvaluator_OUT

OUTCOME	String
ExternalCreditBureau	String
AutoDecisionReqd	String
AutoDecisionOutput	String

## RetailLoan OrchestrationBPEL(BPEL)

#### Human Tasks:

ManualCustTask

#### Exposed Services:

• retailloanorchestration\_client\_ep

#### External Reference:

DBUtility

Task Name/Event Name	Input	Output
RetailLoan OrchestrationBPE L	inputvariable	outputvariable
DBUtility	Invoke_DBUtility_Sav eCust_initiate_InputV ariable_1	Invoke_DBUtility_SaveC ust_initiate_OutputVaria ble



# 1.8 Acronyms and Abbreviations

RL	Retail Lending
BPMN	Business Process Model and Notation
BPEL	Business Process Execution Language
SOA	Service-Oriented Architecture

# 1.9 <u>References</u>

Retail Loan Origination Oracle FLEXCUBE Universal Banking Release 12.0 [May] [2012]

http://docs.oracle.com/cd/E14571\_01/doc.1111/e15176/model\_bus\_procs\_bpmpd.htm





BPMN Process Flow Definition Guide [February] [2018] Version 14.0.0.0

Oracle Financial Services Software Limited Oracle Park Off Western Express Highway Goregaon (East) Mumbai, Maharashtra 400 063 India

Worldwide Inquiries: Phone: +91 22 6718 3000 Fax: +91 22 6718 3001 https://www.oracle.com/industries/financial-services/index.html

Copyright © 2007, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.