BPEL 101 Tutorial

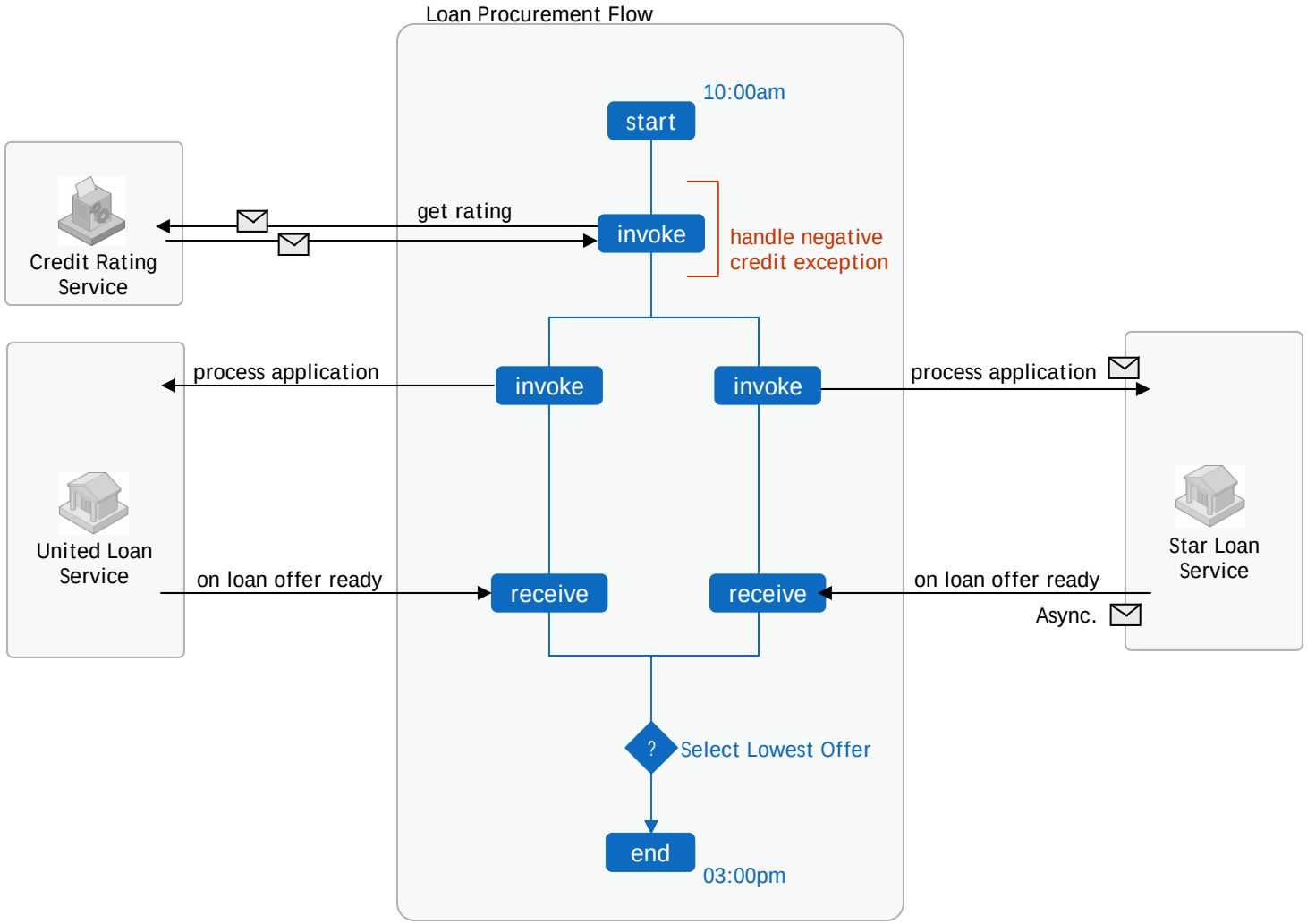
Learn BPEL through the development of a Loan Procurement Process Flow

1. **Requirements of the Loan Procurement Process Flow**
2. **Quick Tour/Demo**
3. **BPEL Code Review**
   Anatomy of a BPEL process. Invoke. Fault Handlers. Assign. Receive. Switch.
4. **Architecture of the Oracle BPEL Process Manager**
5. **Next Steps: Get up an running with BPEL in less than 15 minutes**

David
dave@collz

ORACLE

Requirements of a Loan Procurement Flow

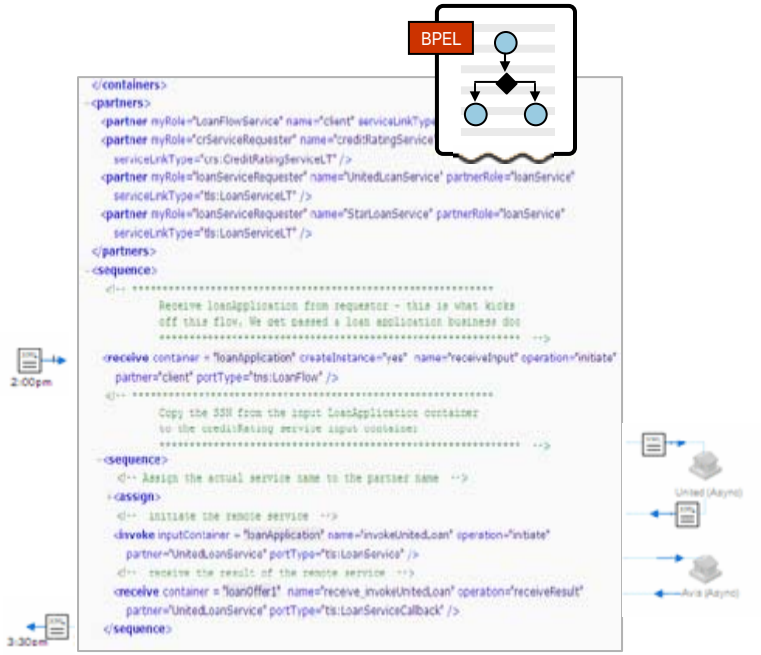


XML Web Services and BPEL applied

BPEL is a programming abstraction that allows developers to compose multiple discrete Web Services into an end-to-end process flow. It has built-in support for asynchronous interactions, flow control and compensating business transactions. It integrates with XPath, XSLT and XQuery for XML data manipulation

BPEL Applied:

- **<invoke>** a credit rating service synchronously
- **<assign>** and manipulate XML documents
- **<scope>, <faultHandlers>** catch and manage exceptions when customer has a bad credit history
- Initiate asynchronous loan processors in parallel <**flow>** of execution
- **<receive>** asynchronous callbacks from long-running loan processors
- **<switch>** to the lowest loan offer



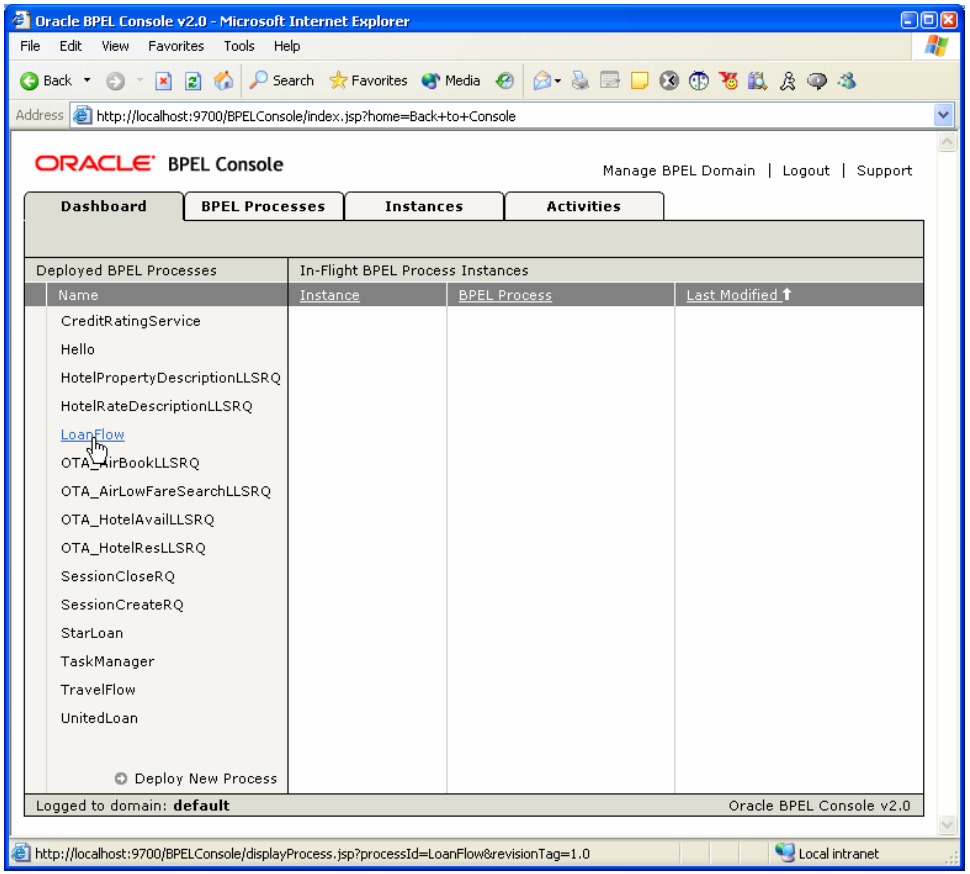
LoanFlow BPEL Scenario (LoanFlow.bpel)

Quick Tour of the Loan Flow BPEL Process

Let's imagine that you have implemented and deployed Loan Flow. Here are the instructions for initiating a test instance of the flow through the BPEL Console. The BPEL Console lists all the BPEL Processes deployed in the BPEL Process Manager.

Q: How do I initiate a new instance of the Loan Flow BPEL Process?

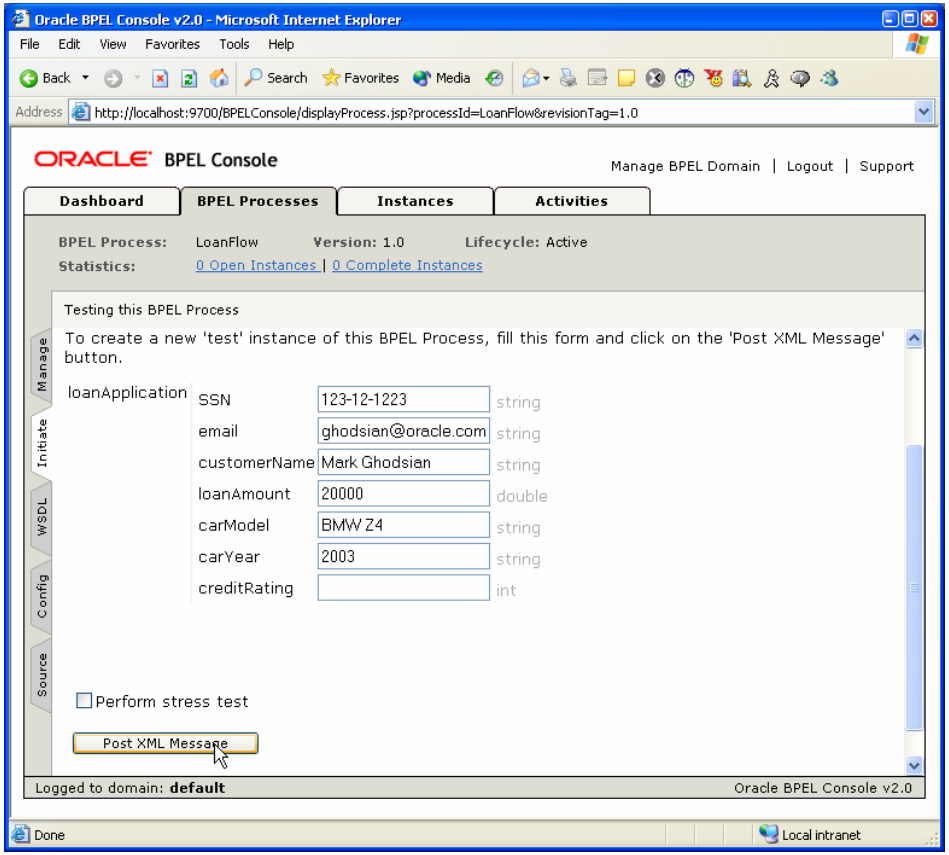
- Go to the BPEL Console **http://localhost:9700/BPELConsole**
- Click on the **LoanFlow** link



Automated Testing Interface

The Console introspects the interface of the LoanFlow BPEL Process and creates an HTML form reflecting the structure of the input document of that flow (in this case a XML Loan Application)

- Fill up the loan application form
- Click on the **'Post XML Message'** button

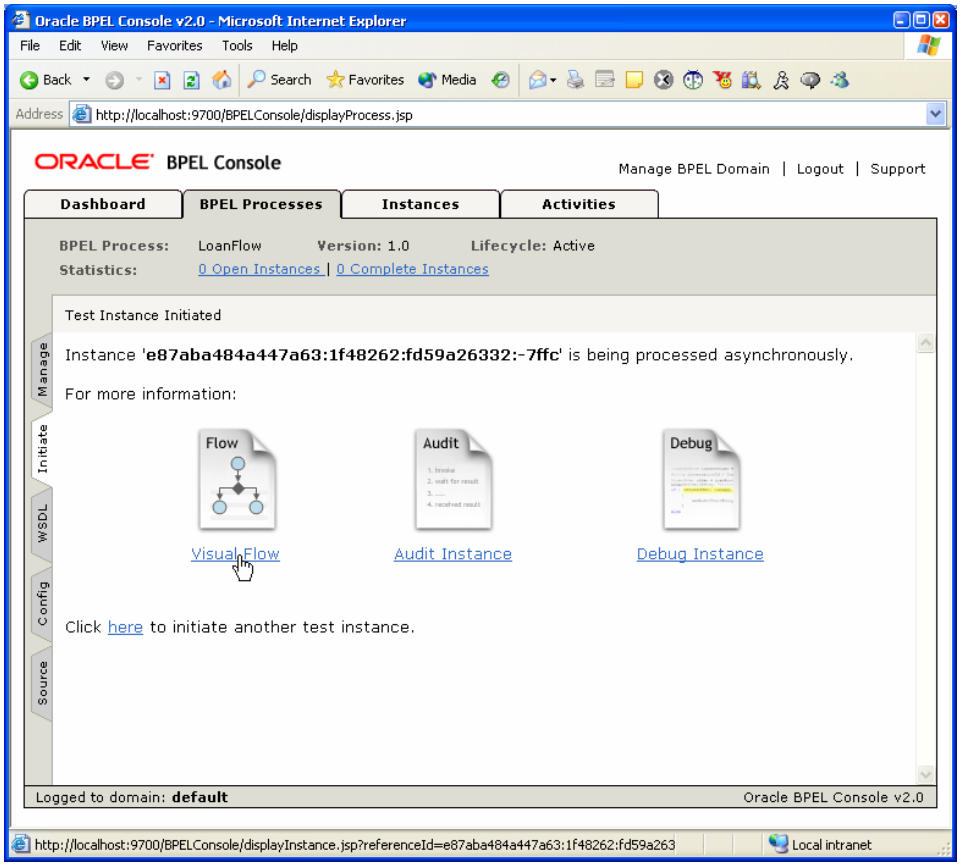


A New Instance of the LoanFlow Has Been Initiated

Behind the scenes, an XML Loan Application document is generated and posted to the BPEL Orchestration Server. The server consumes the message, initiates a flow instance and starts processing it asynchronously.

Q: How do I monitor the processing of the initiated BPEL Process?

- Click on the **Visual Flow** link to visually review the audit trail of that instance



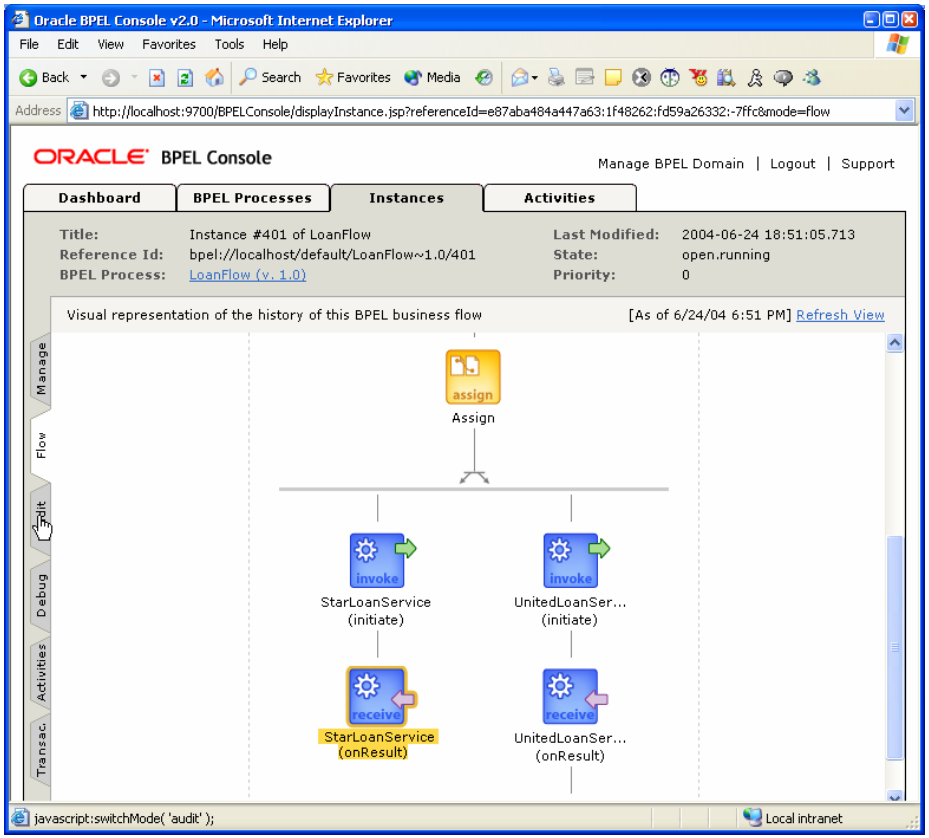
Visual Audit Trail of the BPEL Loan Flow

As you can see, the flow invokes the credit rating service and then initiates in parallel a conversation with 2 asynchronous loan processor services. United Loan has already called back with a loan offer but the flow is still waiting for the asynchronous callback from StarLoan

Note: For reliability and scalability reasons, the BPEL Process Manager automatically **dehydrates** all flow instances that are waiting for an asynchronous callback.

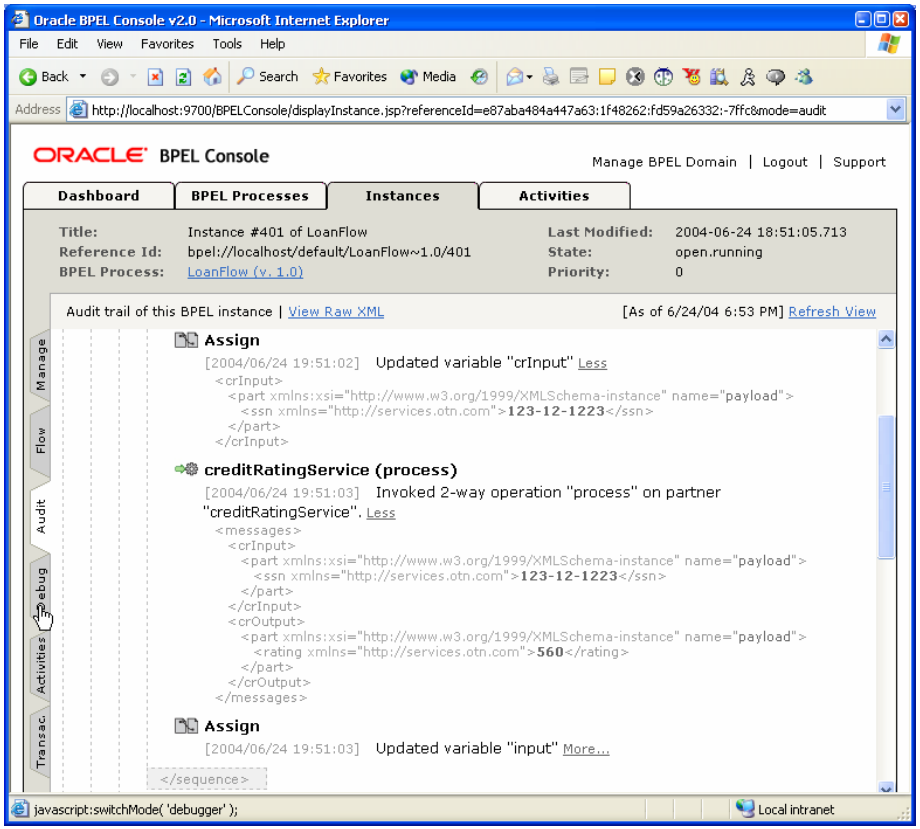
Q: The visual audit trail is nice but can I get more details?

- Click on the **Audit** tab to view a more detailed textual representation of the audit trail



Textual Audit Trail of the BPEL Loan Flow

As you can see the textual audit trail is a slightly more detailed view of the history. More specifically, you can view the XML containers/messages exchanged as part of all interactions.

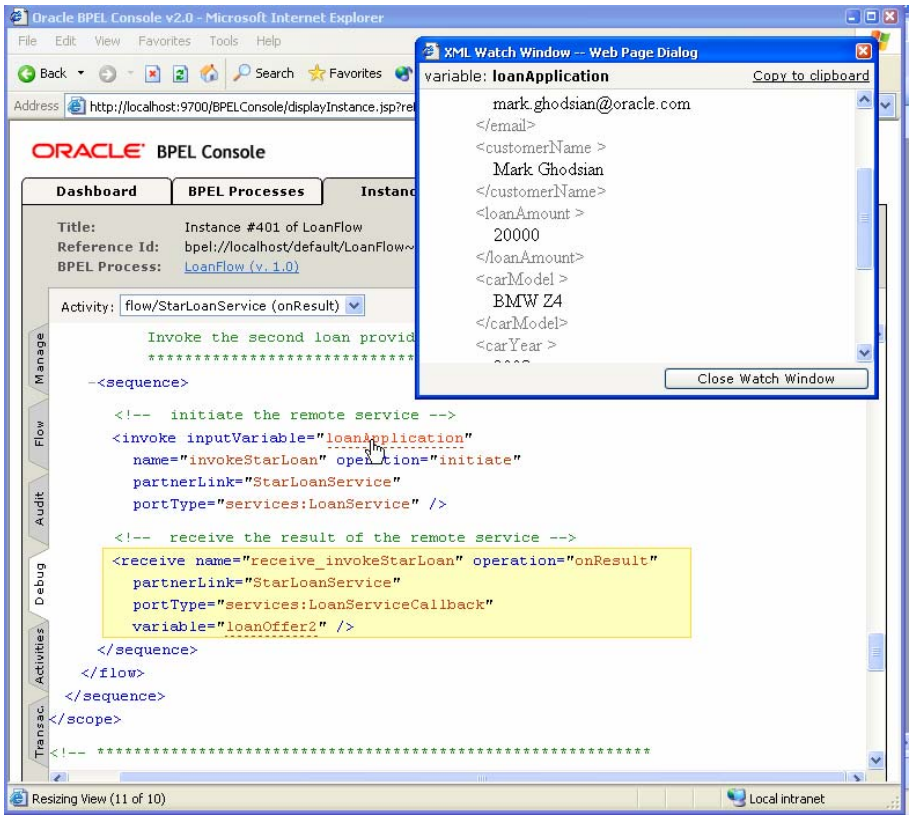


Q: Can I debug the state of the flow instances against my BPEL source code?

- Click on the **Debug** tab to view the state of the dehydrated instance in the context of the BPEL Scenario code.

BPEL Debugger

As you can see, the <receive> activity has been activated and it is waiting for the callback message from StarLoan.



Q: Can I introspect the value of the containers associated with the flow?

1. Click on the **loanApplication** inputVariable to introspect it in the XML Watch Window.

Once the response message arrives, the <receive> activity will consume it and the orchestration server will resume processing the BPEL scenario. But for that to happen you need to switch hats and manually complete the StarLoan loan processing service.

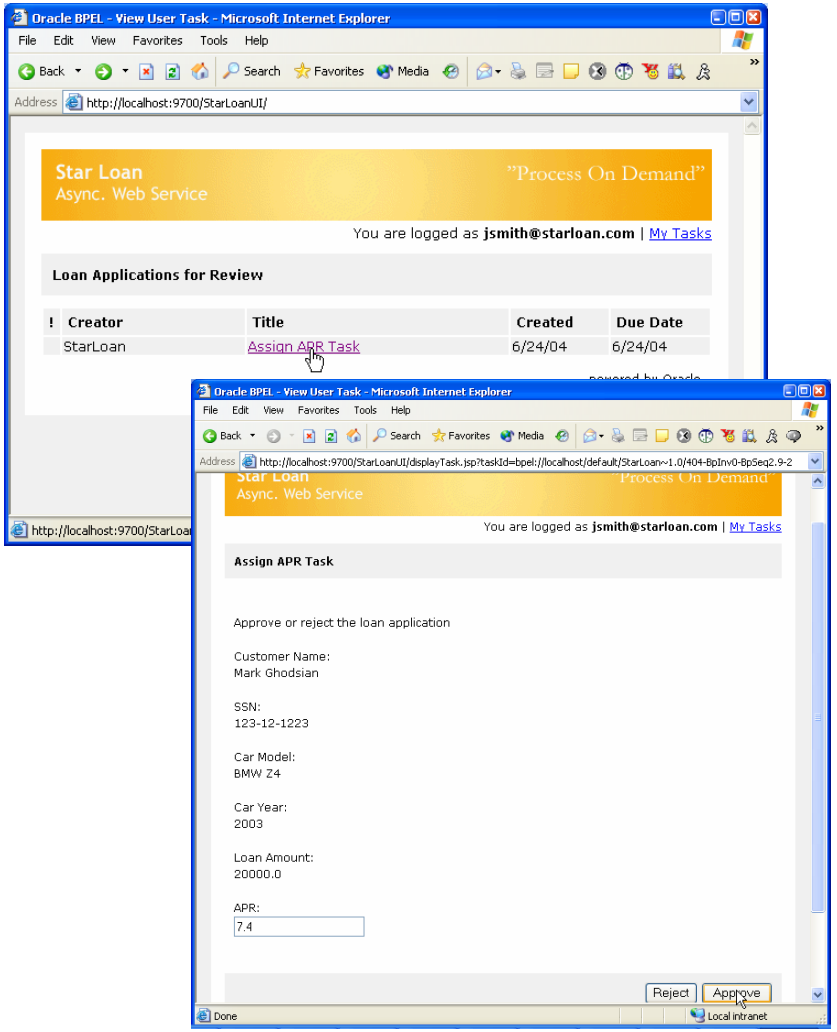
Triggering the Callback from the StarLoan service

Q: How do I complete the StarLoan service and trigger the callback?

- Point your browser at the StarLoan Customer Service portal: **http://localhost:9700/StarLoanUI**

  You will be automatically logged in as StarLoan service rep J. Smith and can select a loan application for review.

2. Select the **Review Loan Application** link for the application with your name in the title.
3. Enter an interest rate and click **"Approve"** to complete the service.
4. Finally, view the audit trail and debugger for your LoanFlow instance (back in the Orchestration Console) to see that the LoanFlow has received and consumed the asynchronous callback and **completed**.



3

Code Review

Code Review

The demo looks OK. Walk me through the BPEL source code!

3

Implementing the LoanFlow BPEL Scenario

Code Review

Q: What is the anatomy of a BPEL Process?

```
-<partnerLinks>
   <partnerLink myRole="LoanBrokerProvider" name="client"
     partnerLinkType="tns:LoanBroker"
     partnerRole="LoanBrokerRequester" />
   <partnerLink myRole="LoanServiceRequester" name="LoanService"
     partnerLinkType="services:LoanService"
     partnerRole="LoanServiceProvider" />
 </partnerLinks>
-<variables>
   <variable name="input"
     messageType="tns:LoanBrokerRequestMessage" />
   <variable name="output"
     messageType="tns:LoanBrokerResultMessage" />
   <variable name="request"
     messageType="services:LoanServiceRequestMessage" />
   <variable name="response"
     messageType="services:LoanServiceResultMessage" />
 </variables>
-<sequence>

   <!-- receive input from requestor -->
   <receive createInstance="yes"  name="receiveInput"
     operation="initiate" partnerLink="client"
     portType="tns:LoanBroker" variable="input" />

   <!-- initialize the input of LoanService -->
  -<assign>
   -<copy>
      <from part="payload" variable="input" />
      <to part="payload" variable="request" />
    </copy>
   </assign>

   <!--  initiate the remote process -->
   <invoke inputVariable="request" name="invoke" operation="initiate"
```

File: LoanFlow.bpel

PartnerLink
References to the services participating in the process flow and their role/port types

Variables
List of messages exchanged between the BPEL process and each of the participating Web Services.

Flow Logic
receive, invoke, assign, pick, wait, throw, terminate,
flow, sequence, switch
Scope, fault handlers, …

[Fault Handler]
Catch and handle faults

[Compensation Handler]
Undo logic

3

<invoke>

Code Review

Q: How do I invoke a remote credit rating Web service?

```
<!-- Invoke the CreditRating Service, the URL of this service's
     WSDL is specified in the deployment descriptor -->
<invoke inputVariable="crInput" name="invokeCR" operation="process"
  outputVariable="crOutput" partnerLink="creditRatingService"
  portType="services:CreditRatingService" />
```

Fragment of LoanFlow.bpel

3

Code Review

<scope> and <faultHandlers>

Q: How do I handle the NegativeCredit fault that might be generated by the credit rating service?

```
<!-- ************************************************************
      Invoke the synchronous creditRatingService. Define a scope
      for handling faults from it and set the credit rating in the
      loan app bus doc if we get a credit rating back. In the case
      of a NegativeCredit exception, set it to -1000.
      ************************************************************ -->
-<scope name="GetCreditRating" variableAccessSerializable="no" >
  <!-- Watch for faults (exceptions) being thrown from
    creditRatingService -->
 -<faultHandlers>
  -<catch faultName="services:NegativeCredit" faultVariable="crError" >
    <!-- For now, just set creditRating to -1000 for
         negative credit exceptions -->
    -<assign>
     -<copy>
       <from expression="number(-1000)" />
       <to part="payload" query="/loanApplication/creditRating"
         variable="input" />
      </copy>
     </assign>
   </catch>
  </faultHandlers>
```

Fragment of LoanFlow.bpel

3

Code Review

<assign>

Q: How do I copy a credit rating field from one XML business document to another one?

```
<!-- Add the credit rating we received to the loan application
     business document -->
-<assign>
 -<copy>
   <from part="payload" query="/rating" variable="crOutput" />
   <to part="payload" query="/loanApplication/creditRating"
     variable="input" />
  </copy>
 </assign>
```

Fragment of LoanFlow.bpel

3

Code Review

<receive>

Q: How do I receive an asynchronous callback from a long-running loan processor service?

```
<!-- invoke first loan provider -->
-<sequence>
   <!--  initiate the remote service -->
   <invoke inputVariable="loanApplication"
     name="invokeUnitedLoan" operation="initiate"
     partnerLink="UnitedLoanService"
     portType="services:LoanService" />
   <!--  receive the result of the remote service -->
   <receive name="receive_invokeUnitedLoan"
     operation="onResult" partnerLink="UnitedLoanService"
     portType="services:LoanServiceCallback"
     variable="loanOffer1" />
</sequence>
```

Fragment of LoanFlow.bpel

Note: For reliability and scalability reasons, flow instances that are waiting for asynchronous callbacks are automatically dehydrated by the BPEL Process Manager into a persistent store. When the asynchronous callback is received, the BPEL Process Manager re-hydrates the instance and resumes the execution of the in-flight instance.

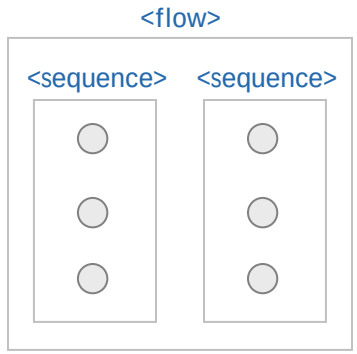
3

<flow>

Code Review

Q: I need to call 2 asynchronous loan processor services. Can I interact with them in parallel and reduce the time required to collect both offers?

```
-<flow>
   <!-- invoke first loan provider -->
  -<sequence>
     <!--  initiate the remote service -->
     <invoke inputVariable="loanApplication" name="invokeUnitedLoan"
       operation="initiate" partnerLink="UnitedLoanService"
       portType="services:LoanService" />
     <!--  receive the result of the remote service -->
     <receive name="receive_invokeUnitedLoan" operation="onResult"
       partnerLink="UnitedLoanService" portType="services:LoanServiceCallback"
       variable="loanOffer1" />
   </sequence>
  -<sequence>
     <!--  initiate the remote service -->
     <invoke inputVariable="loanApplication" name="invokeStarLoan"
       operation="initiate" partnerLink="StarLoanService"
       portType="services:LoanService" />
     <!--  receive the result of the remote service -->
     <receive name="receive_invokeStarLoan" operation="onResult"
       partnerLink="StarLoanService" portType="services:LoanServiceCallback"
       variable="loanOffer2" />
   </sequence>
 </flow>
```



Fragment of LoanFlow.bpel

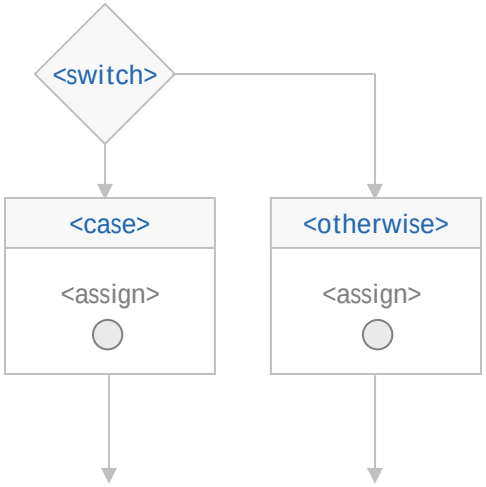
3

<switch>

Code Review

Q: Can I define conditional branches within the execution of the flow? How do I select the loan offer with the lowest rate?

```
-<switch>
   <!-- If loanOffer1 is greater (worse) than loanOffer2 -->
  -<case condition="bpws:getVariableData('loanOffer1','payload','/loanOffer/APR') >
     bpws:getVariableData('loanOffer2','payload','/loanOffer/APR') " >
     <!-- Then take loanOffer2 -->
    -<assign>
     -<copy>
        <from part="payload" variable="loanOffer2" />
        <to part="payload" variable="selectedLoanOffer" />
      </copy>
     </assign>
   </case>
   <!-- Otherwise take loanOffer1 -->
  -<otherwise>
    -<assign>
     -<copy>
        <from part="payload" variable="loanOffer1" />
        <to part="payload" variable="selectedLoanOffer" />
      </copy>
     </assign>
   </otherwise>
 </switch>
```



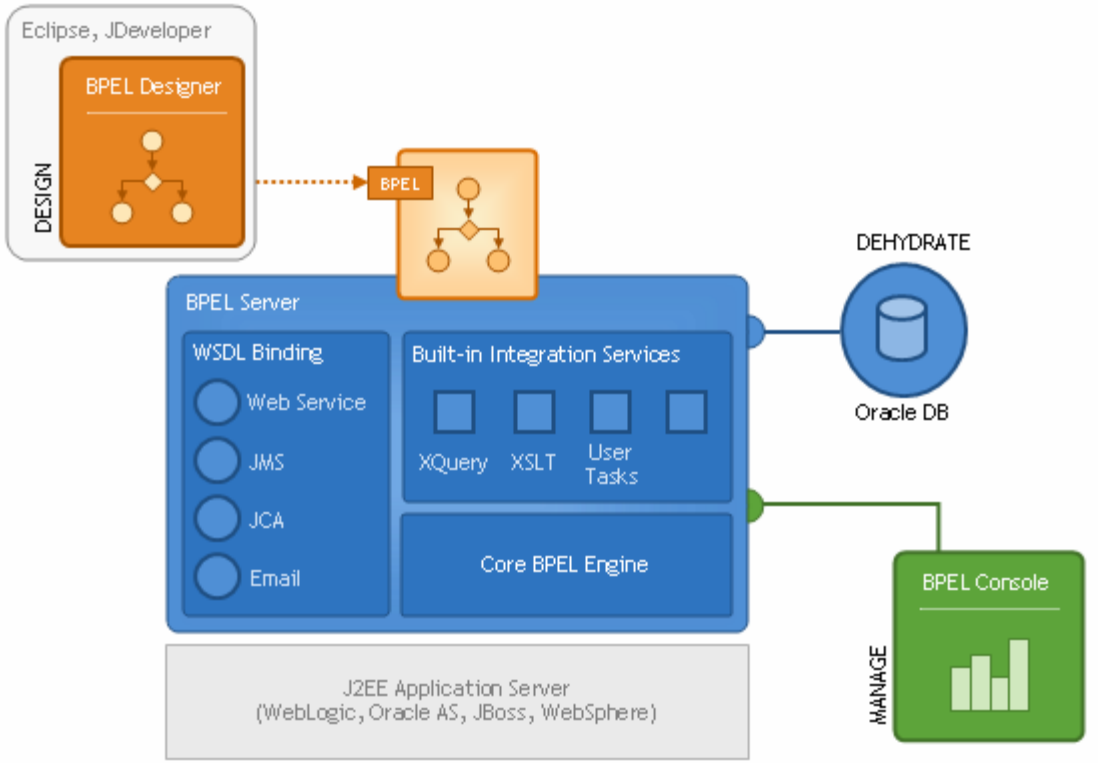
Fragment of LoanFlow.bpel

4

Collaxa

Oracle BPEL Process Manager

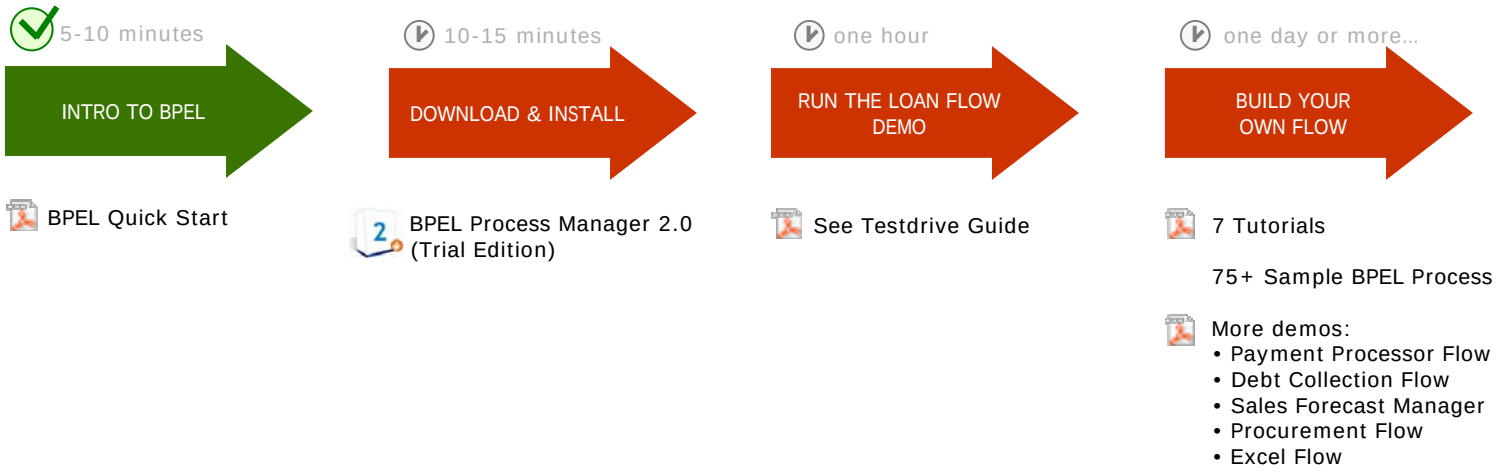
The Oracle BPEL Process Manager offers a comprehensive and easy-to-use solution for designing, deploying and managing BPEL Business Processes



5

Getting Started

Getting Started with BPEL



For more information:
http://otn.oracle.com/bpel