

## Lab A - Exercise 3: Send Notification using Topic

### Overview

In this exercise, you will create a process using a JMS Send Message activity to send the shipment notification out to a JMS topic. Any interested party would subscribe to the topic to receive the notification. You will create a process which uses the JMSReceiveMessage activity to subscribe to that topic and then verify that if two instances of that process are started, they both receive the shipment notification.

### Steps

1. Create SendShipmentNotification process definition and add activities to it.

- Add a new process to **LabA** folder
  - ◆ Name: **SendShipmentNotification**
- Add the following activities to the process
  - ◆ *General Activities* > **Timer**
  - ◆ *JMS* > **JMS Send Message**
  - ◆ *General Activities* > **Log**
- **Save**

2. Configure transitions.

- Add the following transitions:
  - ◆ **Timer** to **JMSSendMessage**
  - ◆ **JMSSendMessage** to **Log**
- **Save**

3. Configure JMSSendMessage as a topic publisher.

- Double-click **JMSSendMessage** activity
  - ◆ Notice that this opens the configuration for that activity in the *Properties* view below
- In *Properties* > **General** configure as follows:
  - ◆ Name: **JMSTopicPublisher**
  - ◆ Messaging Style: **Topic**
  - ◆ JMS Connection: Browse for **EMS206project.JMSConn** shared resource
  - ◆ Destination Topic: **topic.sample**
- Select *Input* tab
  - ◆ *Activity Input* > **Body: "Shipment on its way"**

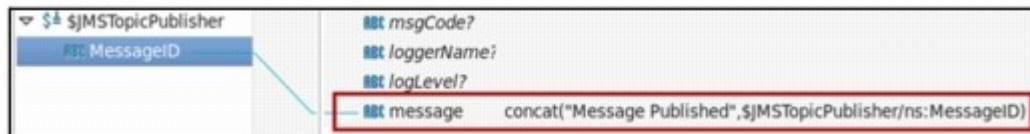




- Save the project

4. Configure the Log activity.

- Select the **Log** activity
  - ◆ Under *Properties*, select the *Input* tab and map as shown:



- Save the changes

5. Create a process to subscribe to the topic and add activities and a transition.

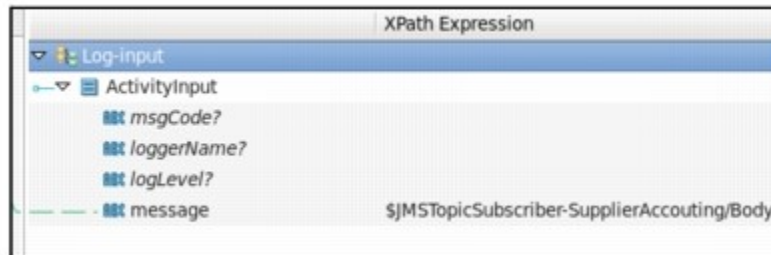
- Add a new process definition to the LabA folder
  - ◆ Name: **ReceiveShipmentNotification-Customer**
- Add activities to the process
  - ◆ **JMSReceiveMessage**
  - ◆ **Log**
- Add a transition from the **JMSReceiveMessage** to the **Log**
- Save the project

6. Configure the topic subscriber process activities.

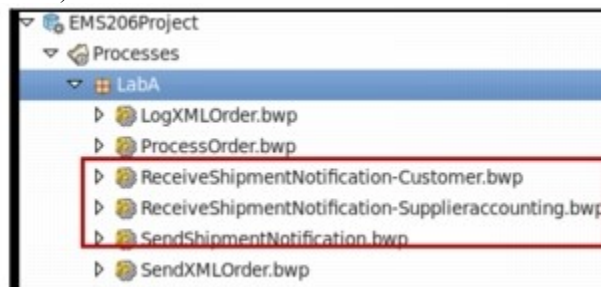
- Select **JMSReceiveMessage** and in *Properties* > **General** configure:
  - ◆ Name: **JMSTopicSubscriber-Customer**
  - ◆ Messaging Style: **Topic**
  - ◆ JMS Connection: Browse for JMS Connection shared resource
  - ◆ Destination Topic: **topic.sample**
  - ◆ **Save**
- Select **Log** and in *Properties* > **Input** configure the mapping:
  - ◆ From the available *Data Source* on the left, select the element **\$JMSTopicSubscriber-Customer/Body**
  - ◆ Drag it to the *Activity Input* element **Log/ActivityInput/message** (on the right)
  - ◆ **Save**

7. Create a copy of the ReceiveShipmentNotification-Customer process.

- Select the LabA folder in the *EMS206Project* tree
- Copy and paste the *ReceiveShipmentNotification-Customer* process into the same folder
  - ◆ Name: **ReceiveShipmentNotification-SupplierAccounting**
- Select the **JMSTopicSubscriber-Customer** activity and change the name
  - ◆ Name: **JMSTopicSubscriber-SupplierAccounting**
- Verify that the input mapping of the **Log** activity is updated automatically



- Verify that you have one process to send notification and two processes to receive it (as shown below)



8. Configure the component list with all three Shipment Notification processes.

- In Project Explorer, open *EMS206Project* > *Module Descriptors* > **Components**
- Delete all the existing components
- Configure components for the following processes:
  - ◆ *Send Shipment Notification*
  - ◆ *Receive Shipment Notification - Customer*
  - ◆ *Receive Shipment Notification - Supplier Accounting*
- **Save** the project

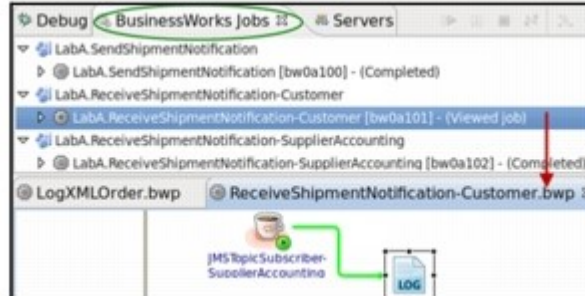
9. Run the debugger and verify the results.

- Right-click on the process diagram and choose **Debug BusinessWorks Applications**
- Select the *Console* view to follow the execution while the debugger is running

```
> 17:48:18.745 INFO [Start Level Event Dispatcher] com.tibco.thor.frwk.Deployer - Started by BusinessSt
[Start Level Event Dispatcher] com.tibco.bw.frwk.engine.BWEngine - TIBCO-BW-FRWK-300002: BW Engine [Main
[Framework Event Dispatcher] com.tibco.thor.frwk.Deployer - TIBCO-THOR-FRWK-300001: Started OSGi Framewo
[Framework Event Dispatcher] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300018: Deploying BW Appl
[Framework Event Dispatcher] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300021: All Application d
[Thread-21] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300006: Started BW Application [EMS206Proj
[Framework Event Dispatcher] com.tibco.thor.frwk.Application - Started by BusinessStudio, ignoring .enab
[bwEngThread:In-Memory Process Worker-1] c.t.b.p.g.Log.EMS206Project.Log - Message PublishedID:EMS-SERVE
```

```
[DWEEngineThread:In-Memory Process Worker-3] c.t.b.p.g.Log.EMS206Project.Log - Shipment on its way
[bwEngThread:In-Memory Process Worker-2] c.t.b.p.g.Log.EMS206Project.Log - Shipment on its way
```

- Select the *BusinessWorks Jobs* view - located right beside the *Debug* view
  - ◆ Observe the order in which components are executed
    - *SendShipmentNotification* is the first - it sends the message to the topic.sample
    - Both Subscriber processes start as they both receive the message
  - ◆ Click the job inside of each component to view the execution path



- Green transitions indicate which path was executed

10. Verify that topic messages are only received by active subscribers.

- Configure the component list to execute only *Send Shipment Notification*
- Start the debugger
  - ◆ Notice that the process executes immediately and publishes the message
  - ◆ Relaunch the debugger to run the process a few times and publish a few more messages
- Configure the component list to include one or both *Receive Shipment Notification* processes
  - ◆ Do not include *Send Shipment Notification*
- Relaunch the debugger
  - ◆ **JMSTopicSubscriber** activities are waiting for the message
    - The messages sent prior to activating the subscribers are not received
- Terminate the debugger
- Switch to **Design** perspective
- Select *File > Close All* to close all Process Editors