

## Lab D - Exercise 2: Test Dynamic Destinations

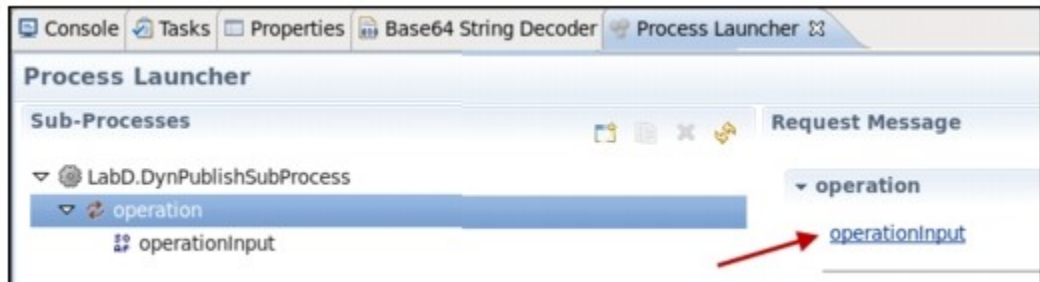
### Overview

In this exercise you send messages on dynamic topics **dest.new**, **dest.ship** and **dest.new.order** using the DynPublisher process. These topics will be dynamically created on the EMS Server. DynSub will receive the messages published on these dynamic topics using wildcards and will filter messages based on message selector.

### Steps

1. Launch the debugger to start the publisher and subscriber.

- Configure the Components list to have:
  - ◆ *InvokePublisher*
  - ◆ *DynSubscriber*
- Launch the debugger
  - ◆ Select *BusinessWorks Jobs* view
    - Notice that the single instance of the Invoke Publisher job is listed as completed
  - ◆ Select the *Process Launcher* view in the bottom of the workspace
    - Select subprocess *DynPublishSubProcess*
    - Click operation input under Request Message

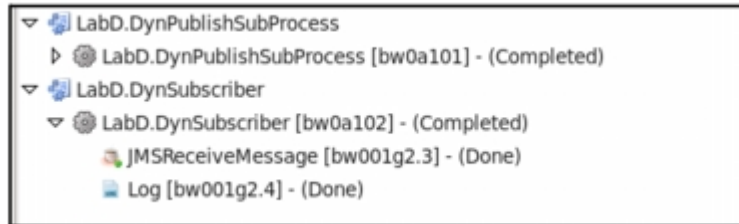


2. Provide input and invoke DynPublisherSubProcess.

- Type in the values for input parameters
  - ◆ destination: **dest.new**
  - ◆ price: **60**

Request Message	
Node	Content
parameters	
operationRequest	
destination	dest.new
price	60

- Invoke the subprocess by clicking "Invoke" button in the top right corner
- Select *BusinessWorks Jobs* view
  - ◆ Verify that the message is published by *DynPublishSubProcess* job
  - ◆ Observe that the *DynSubscriber* job is also executed



- Select **JMSReceiveMessage** activity in *DynSubscriber* job
  - ◆ View *JobData > All*
    - Verify that the message sent to **dest.new** is received

JMSReceiveMessage		
Output	Name	Value
All	JMSReceiveMessage	<ActivityOutput xmlns="http://
	JMSHeaders	<JMSHeaders>\n <JMSDestina
	JMSProperties	<JMSProperties/>\n
	Body	Message sent to dest.new
	OtherProperties	<OtherProperties xmlns="http

### 3. Verify that a topic is created on EMS Server.

- Start EMS Admin tool and connect to EMS Server if necessary
- Verify dynamic topic is created on EMS Server
  - ◆ Execute the command **show topics**
    - Notice that the topic **dest.new** is marked by the '\*' in the first column
  - ◆ Issue the command: **show topic dest.new**

```

tcp://educ1t:7222> show topic dest.new
Topic:                dest.new
Type:                 dynamic
Properties:            *prefetch=64,*st
JNDI Names:           <none>
Bridges:              <none>

```

- ◆ Observe that **dest.new** will be removed from the server in few minutes

### 4. Publish more messages using different dynamic destinations.

- Select *Process Launcher*
- Change parameter:
  - ◆ destination: **dest.ship**
- Invoke one more instance of the *DynPublisherSubProcess*
  - ◆ To do so click "Invoke" button in the top right corner
  - ◆ Verify that both publisher and subscriber jobs are executed again

- Message is published to **dest.ship** and consumed
- Change parameter:
  - ◆ destination : **dest.new.order**
- Invoke one more instance of the *DynPublisherSubProcess*
  - ◆ Observe that this time only the Publisher job runs, but the Subscriber job does not
    - The destination **dest.new.order** does not match its subscription to dest.\*
- Terminate the debugger

5. Update the DynSubscriber to accept all destinations starting with dest.

- Select the *DynSubscriber* process diagram
  - ◆ Select **JMSReceiveMessage** activity
  - ◆ In *Properties* > **General** modify the destination name
    - Destination: **dest.>**

<b>Name:</b>	JMSReceiveMessage
<b>Messaging Style:</b>	Topic
<b>JMS Connection:</b>	jmsConnection
<b>Destination:</b>	dest.>

- ◆ **Save**
- Relaunch the debugger
  - ◆ Select *Process Launcher*
  - ◆ Click on *DynPublisherSubprocess* and then *operations*
  - ◆ Set parameter:
    - destination : **dest.new.order**
- Invoke one more instance of the *DynPublisherSubProcess*
  - ◆ To do so click "Invoke" button in the top right corner
  - ◆ Verify that the message is consumed by *DynSubscriber* on topic **dest.new.order**
- Keep the debugger running

6. Test the message selector.

- In Process Launcher, change the input parameter for price
  - ◆ price: **40**

Process Launcher											
<div> <div> <b>Sub-Processes</b> <ul style="list-style-type: none"> <li>LabD.DynPublishSubProcess               <ul style="list-style-type: none"> <li>operation                   <ul style="list-style-type: none"> <li>operationInput</li> </ul> </li> </ul> </li> </ul> </div> <div> <b>Request Message</b> <table border="1"> <thead> <tr> <th>Node</th> <th>Content</th> </tr> </thead> <tbody> <tr> <td>parameters</td> <td></td> </tr> <tr> <td>operationRequest</td> <td></td> </tr> <tr> <td>destination</td> <td>dest.new.order</td> </tr> <tr> <td>price</td> <td>40</td> </tr> </tbody> </table> </div> </div>		Node	Content	parameters		operationRequest		destination	dest.new.order	price	40
Node	Content										
parameters											
operationRequest											
destination	dest.new.order										
price	40										

- Invoke *DynPublishSubProcess*
  - ◆

Verify that the message is not consumed

***Analysis:*** As the message selector is "appPrice > 50", consumer filters out messages with the appPrice less or equal than 50.

---

- **Terminate** the debugger when done
  - Switch to **Design** perspective
- 



**TIBCO Education Programs**