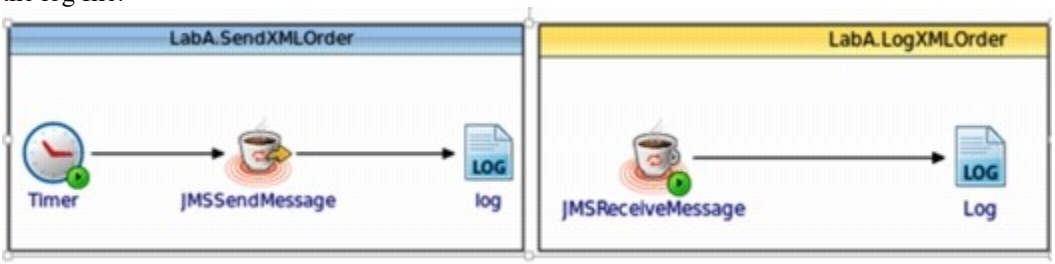
Lab A - Exercise 1: Send and Receive XML Message using Queue

Overview

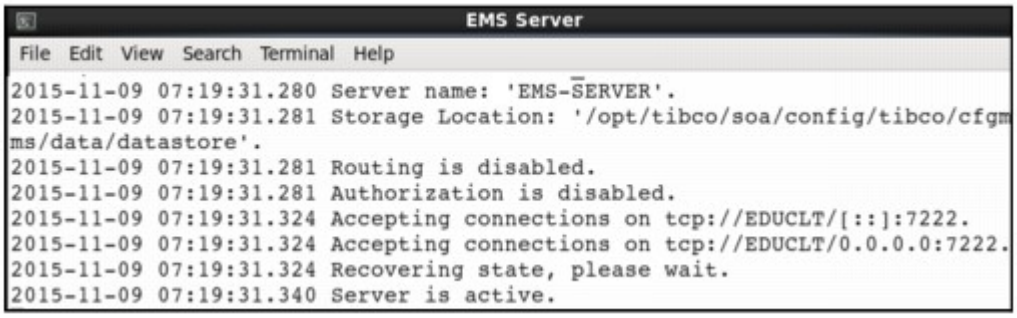
In this exercise, you develop the BusinessWorks process to send an order as a message of type XML text to a queue. Then you develop a process which receives the order as a text message and stores the content in the log file.



Steps

1. Start the TIBCO EMS Server.

- Select *Applications* > *TIBCO* > *EMS* > ***TIBCO EMS Server - Start***
  - Verify the message confirming that the server is active



Note: You can minimize the EMS Server console window but you must keep it open to have EMS server running.

2. Launch TIBCO Business Studio and define a new workspace.

- Open a Terminal window and change directory to **`/opt/tibco/soa/studio/3.6/eclipse`**
- Enter the following command to launch TIBCO Business Studio
  `./TIBCOBusinessStudio`
- Workspace: **`/home/tibco/EMS206/workspace`** (overwrite default value)

Tip: The following Error may be displayed:

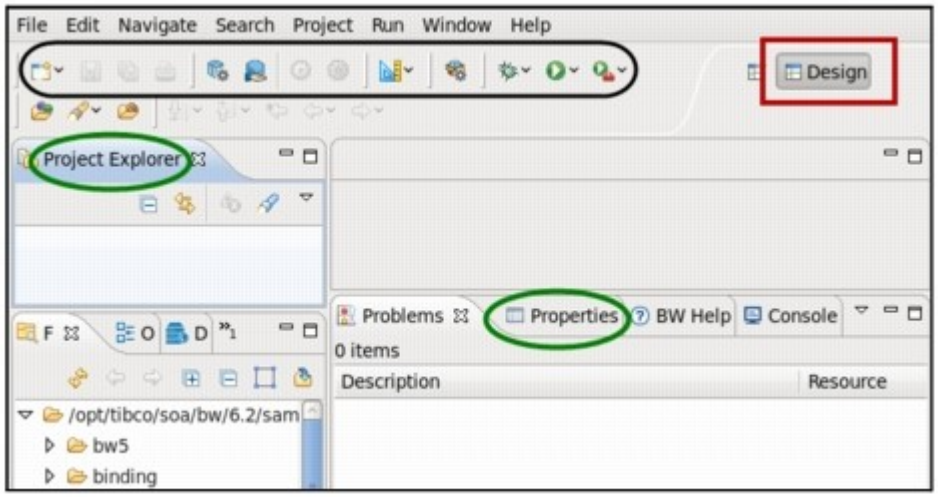
Failed to load JavaHL Library.

These errors were encountered: ....no libsvnjavahl-1 in java.library.path....

Solution: Click **OK**, and execute the following steps:

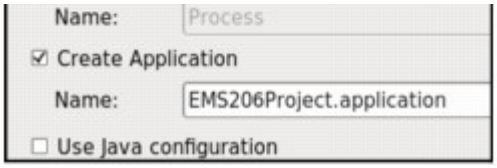
- Select *Window* > *Preferences* > *Team* > ***SVN***
  - SVN interface Client: Change default "javaHL(JNI) Not Available" by selecting **SVNKit(Pure Java) SVNKit v1.7.9.9659** from the list
- Click **OK** and restart Business Studio

- Close the *Welcome* tab
- Notice that Business Studio opens in the **Design** perspective and shows:
  - Views: *Project Explorer, Properties*, *Problems*, *Console* etc.
  - Workbench toolbar (at the top) with buttons for frequently used commands



3. Create a new BusinessWorks Application module.

- Click the "New BusinessWorks Application Module" icon  on the toolbar
  OR
  In the *Project Explorer* view, right-click and choose *New* > ***BusinessWorks Application Module***
  - Project name: **`EMS206Project`**
  - Use default location: checked
  - **Uncheck** Create empty process
  - Create Application: checked
  - Click **Finish**



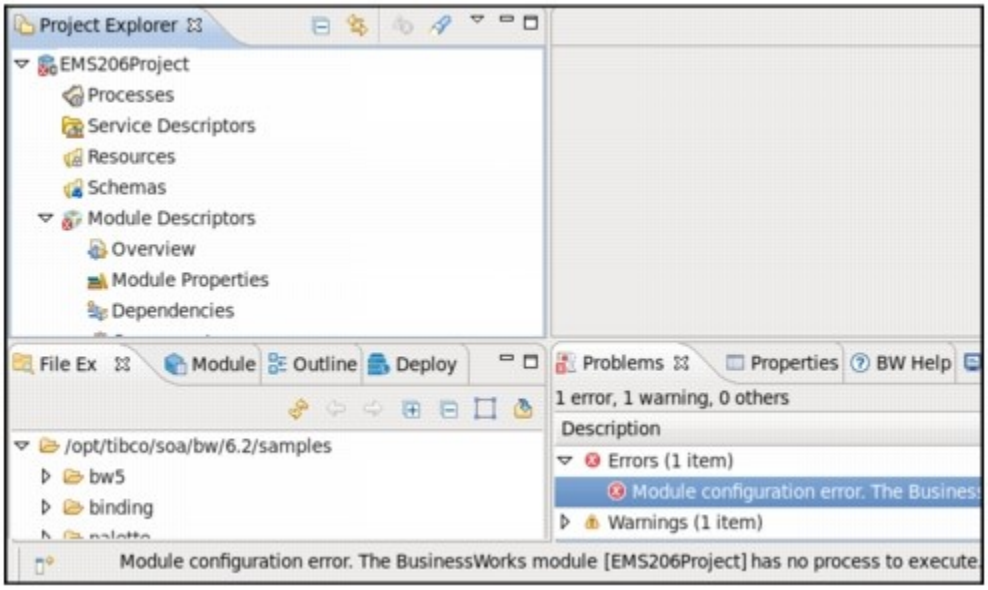
- Observe that the project is created, and that it has several folders

- The *Overview* window (view) is opened by default
- **Close** *Overview*

Tip: When beginning development, define components individually to understand how they are related to each other.

4. Observe the application error indicators and analyze the errors.

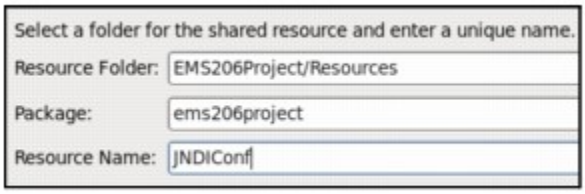
- Notice that the *EMS206Project* is marked by the red "Error" marker



- The project has one error (indicated by the red marker), and a warning
- Select *Problems* view in the bottom right section to examine error details
  - Click on the error to display the full description at the bottom of the workbench
  - The error message indicates that no process has been defined (yet)

5. Define a JNDI configuration resource to connect to the local EMS Server.

- In Project Explorer, select *EMS206Project* > ***Resources***
  - Right-click and select *New* > ***JNDI Configuration***
    - Resource Name: **`JNDIConf`**

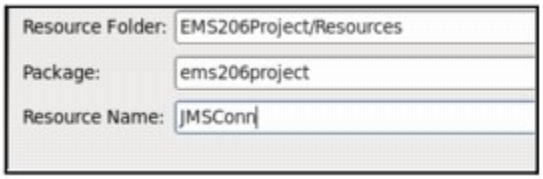


- Click **Next**
  - Verify default settings in the next window

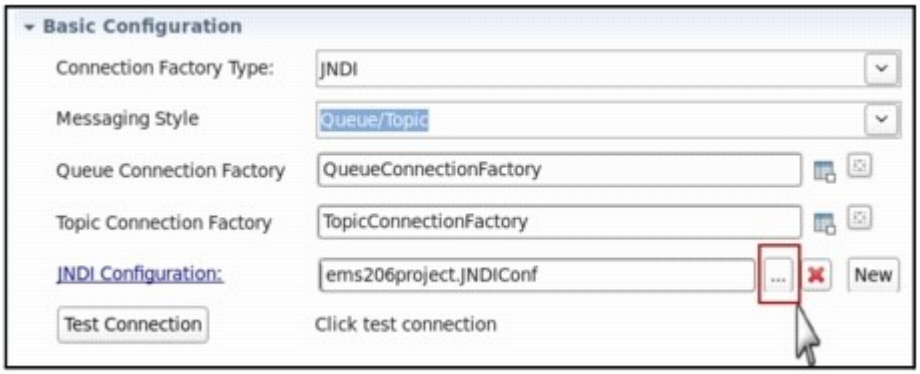
- Click **Finish**
- **Close** the *JNDI configuration* tab (editor) in the top panel

6. Add a JMS connection resource that uses the JNDI resource you just created.

- In Project Explorer, select *EMS206Project* > ***Resources***
  - Right-click and select *New* > ***JMS Connection***
    - Resource Name **`JMSConn`**
    - Click **Finish**



- The *JMS configuration* editor is opened
- Update it as follows:
  - Messaging Style: **Queue/Topic** (select from dropdown)
    - Queue/Topic Connection Factory fields appear with default values
  - JNDI Configuration: **JNDIConf** (Use the **Browse** button to select)



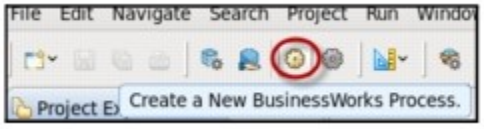
- **Test Connection** and verify success
  - Note that the EMS server has to be running for this lab.
- **Save** your project
- **Close** the *JMS Connection Resource* tab

7. Create a new process to send an XML message to a queue.

- In *Project Explorer*, select *EMS206Project* > ***Processes***
  - Right-click and choose *New* > ***Folder***

- Folder name: **`LabA`**
- Select the new folder and create a new process in it
  - Right-click and choose *New* > ***BusinessWorks Process***

    OR use the "Create a New BusinessWorks Process" icon in the toolbar



- Set these values
  - Package: **LabA** (default)
  - Process Name: **`SendXMLOrder`**
  - Modifiers: **public** (default)
  - Patterns: **Empty Process** (default)
  - Click **Finish**
- Notice that the empty process is created and the Process Editor is opened

Tip: As soon as the Process Editor opens, the Palette View is displayed next to it to the right.

8. Add a JMS Send Message activity to the process.

- Right-click in the Process Editor window and choose *Add Activity* > *JMS* > ***JMS Send Message***

Note: The activity will be configured in the steps that follow.

9. Add two more activities to the process.

- Locate the *Palette* view (right corner)
- Click *Palette Library* > ***General Activities***
  - A submenu *General Activities* opens
  - Select the **Log** activity and drop it to the Process Editor window

Tip: Do not drag the activity from the Palette into the Process Editor window - simply select the activity and then click into the Process Editor window to drop it.

- Similarly add a **Timer** activity (*General Activities* > ***Timer)***
- Position **Timer**, **JMSSendMessage** and **Log** from left to right in the Process Editor window

10. Create transitions between activities.

- Click on the **Timer** activity to select it

- Click the transition icon  that appears on the right of **Timer**

- Drag the transition icon to the **JMSSendMessage** activity
  - This defines a transition between the two activities

- Similarly, connect **JMSSendMessage** to the **Log** activity
- **Save** changes

11. Configure the JMSSendMessage activity.

- Click on the **JMSSendMessage** activity to select it
- Select the *Properties* view below the Process Editor
  - Observe the list of tabs on the left hand side

Tip: Each tab in the *Properties* view for an activity contains elements of the configuration for that activity.

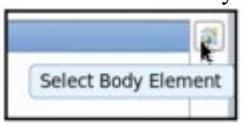
- In *Properties* > ***General*** tab, update as shown (for other fields, keep their default values):
  - Name: **`JMSSendMessage`**
  - Messaging Style: **Queue** (select from dropdown)
  - JMS Connection: **EMS206project.JMSConn** (browse and select the resource created in a previous step)

  - Destination: **`queue.sample`**
  - 

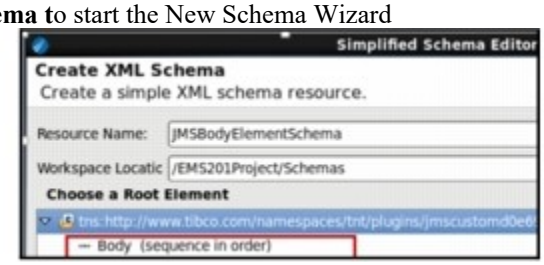
Message type: **XML Text** (select from dropdown)

12. Define the structure of the inbound XML message.

- With **JMSSendMessage** selected, click the *Properties* > ***Input Editor*** tab
  - Click the "Select Body Element" icon in the right corner



- Click **Create New Schema t**o start the New Schema Wizard



  - Accept defaults for Resource Name and Workspace Location
  - Double-click **Body,** rename it to **`Root`** and press **<Enter>**
  - Click the "Add Primitive Element" icon **(**circled)
    - This adds an element below Root
  - Rename "primitive" to **`OrderID`**

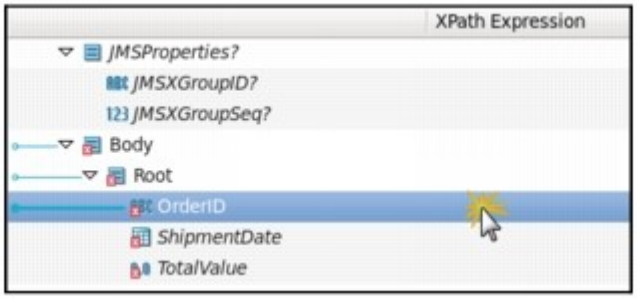
  - Press <Enter> to save the changes
  - Add two more primitive elements and configure as shown:
    - Name: **`ShipmentDate`**, type: **Date&Time**, subtype: **Date**

    - Name: **`TotalValue`**, type: **Decimal**, subtype: **Double**

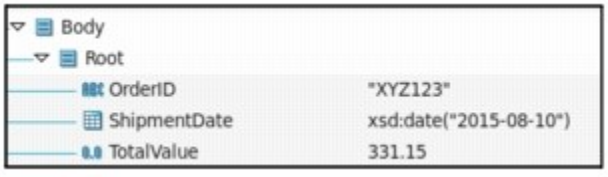
- Select **Root** and click **OK**

13. Configure the input mappings for JMSSendMessage.

- Select the *Properties* > ***Input*** tab
- Under *Activity Input* on the right, locate the message body (scroll down if necessary)
  - Click on **OrderID** and then click below the **XPath Expression** heading to open the XPath Expression Editor



    - Type in the value **`"XYZ123"`**
    - Press <Enter>
- Similarly, configure all the input data as follows:



  - **Save** your changes
    - Notice that the error marker is now removed from the **JMSSendMessage** activity

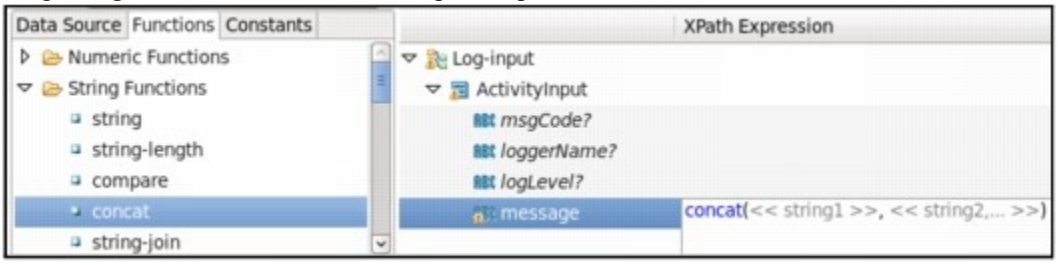
14. Configure the log activity.

- Select **Log** and navigate to the *Properties* > ***Input*** tab
- Create the following mapping for *Activity Input*

  > ***message***
  - Type in the mapping directly (as described in the previous step)

    OR

    Build the expression using Xpath Expression Builder:
    - Click *Functions* tab
    - Drag *String Functions* > ***concat*** to the Xpath Expression Editor



    - Type **`"Message Sent "`** over the first argument <<string1>>

- Drag **`$JMSSendMessage/MessageID`** on top of <<string2>>
- **Save**

15. Verify the configuration for the Timer.

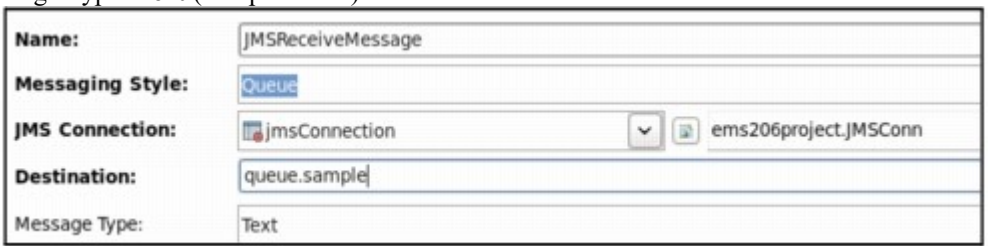
- Select **Timer**
- In *Properties* > ***General***, confirm that the **Run Once** option is checked

16. Create a process to receive the message and write it to the log.

- Add a new process to the *EMS206Project/Processes/**LabA*** folder and configure it:
  - Name: **`LogXMLOrder`**
- Add the following activities to the process
  - **JMS Receive Message** activity from the *JMS* palette
  - **Log** activity from the *General Activities* palette
- Create the following transitions:
  - From **JMSReceiveMessage** to **Log**
  - **Save** your project

17. Configure activities in the LogXMLOrder process.

- Select the **JMSReceiveMessage** activity from the process and configure as follows:
  - Messaging Style: **Queue** (select from dropdown)
  - JMS Connection: Browse and select the **EMS206project.JMSConn** connection
  - Destination Queue: **`queue.sample`**
  - Message Type: **Text** (accept default)



- Configure the **Log** activity input
  - Select the *Properties* > ***Input*** tab
  - Drag the **`$JMSReceiveMessage/Body`** from *Data Source* to the **`message`** in *Activity Input*
- **Save** your project

18. Create a list of components to test your application module in design-time environment.

- Configure the list of application module components to be executed
  - In Project Explorer, double-click *EMS206Project* > *Module Descriptors* > ***Components***
  - You will see two components - one for each process in your application module:

- *LogXMLOrder*
- *SendXMLOrder*

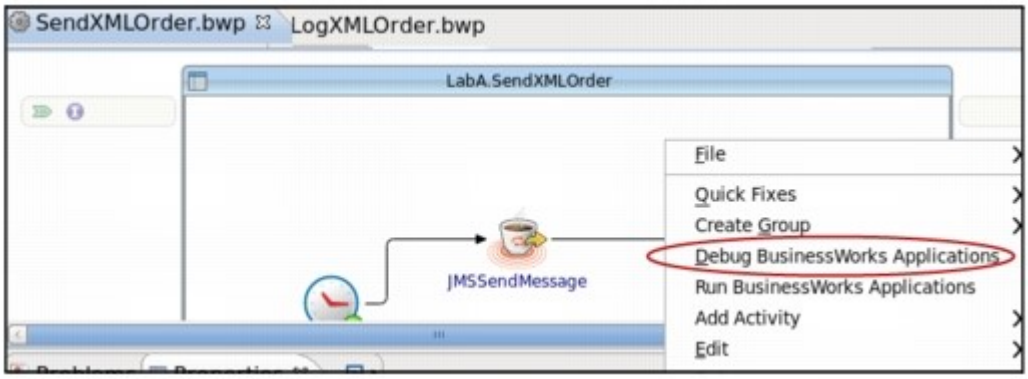
Tip: When you test your application in the design-time environment, Business Studio uses this list to bundle runtime components which will be executed when you start the debugger.

- Remove the component for the *LogXMLOrder*
  - Click the component **LogXMLOrder** to select it
  - Click the "Remove selected component icon" 
- Verify that your component configuration now contains only the component for *SendXMLOrder*

- **Save** your project

19. Launch the debugger to test processes in Components List.

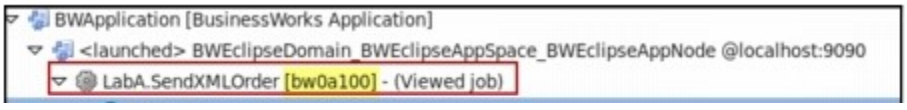
- Right-click inside the **SendXMLOrder** process diagram (within the *SendXMLOrder* tab)
  - Choose ***Debug BusinessWorks Applications*** (this launches the debugger)



  - Notice that Business Studio is now in the **Debug** perspective
    - If prompted to confirm switching to **Debug** perspective, click **Yes**

20. Examine the debug view to see process execution details.

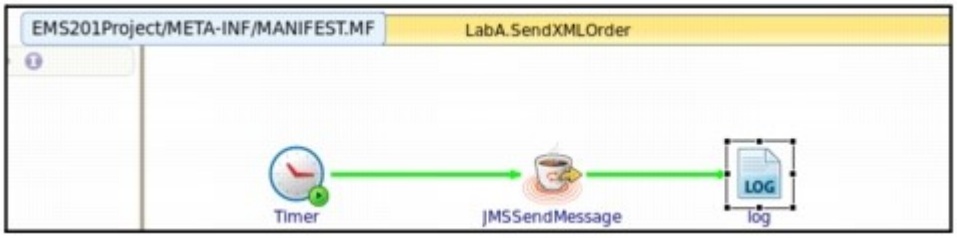
- Locate the *Debug* view (top left corner)
  - Expand the entry: *<launched> BWEclipseDomain_BWEclipseAppSpace...*
  - Confirm that *SendXMLOrder* is completed
- Expand the *SendXMLOrder* job to view its details



- Notice that the **Timer**, **Log** and **JMSSendMessage** activities were executed
- Observe the **JobID** for this job
  - JobID is shown beside the name of the job

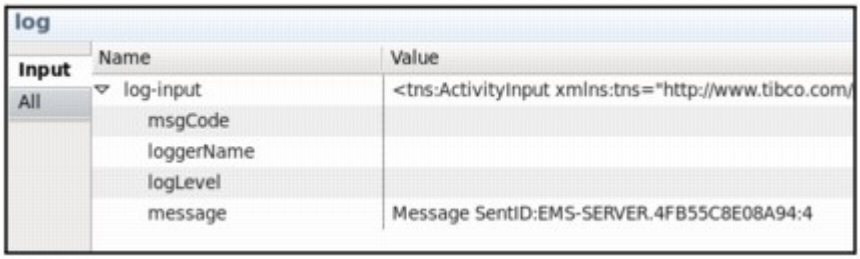
21. Examine interactions between the Debug view and the process diagram.

- In the *Debug* view, click the **log** activity in the list below the *SendXMLOrder* job
  - Notice that this activity is highlighted in the process diagram below
  - Click on other activities and verify that the process highlights the relevant activity
  - When you click on *SendXMLOrder*, all activities are shown connected with a green arrow



22. View input and output data for activities.

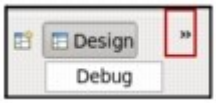
- Locate the *Job Data* view (top right-hand section), and click on it
  - No job data may be displayed at the moment
- Click on any activity in *Debug* view
  - Job data is displayed - depending on the activity the default sub-tab may be *Input* or *Output*
  - Click the *All* sub-tab to view input and output data



  - *Input* tab indicates that an input is provided to the activity
- Click *Input* to see input data for this activity
- **Stop** the debugger when done exploring
  - Select the application entry in the *Debug* view
  - Click the "Terminate" icon 

23. Reconfigure the list of components to test the LogXMLOrder process.

- Switch to the *Design* perspective



  - Use the Perspective bar in the top right corner of the Workbench
  - Select *EMS206Project* > *Module Descriptors* > ***Components***

  OR

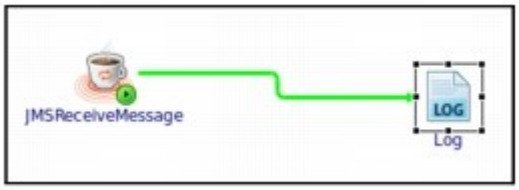
  Still in the *Debug* perspective, select *EMS206Project* tab - it shows the current component list
- Delete *SendXMLOrder* from the components list
- Create a process component for *LogXMLOrder* - use the icon  (top right corner)
- **Save**

24. Test the LogXMLOrder process.

- Start the debugger
  - Right-click on any process diagram and choose ***Debug BusinessWorks Applications***

  OR

  With focus on any process diagram, click the "Debug BWApplication" icon  in the toolbar
- The process *LogXMLOrder* runs immediately, receiving the message sent earlier by *SendXMLOrder*
- Verify that the receiver gets the message even if it was not active when the message was sent
  - Select the **Log** activity in the *Debug* view tree:



  - With the **Log** activity still selected, click the *Input Data* tab in *Job Data* view
    - Notice that the message is received as text in XML format

- **Terminate** the process by clicking the red square button