webMethods 8.2
BPM for Developers

Exercise Guide

TRA622-51E

TABLE OF CONTENTS

Exercise 1:
Configure the Runtime and Development Environment

Overview

In this exercise, you will verify that the appropriate servers have been started as Windows services. In case they are inactive, you will manually start them by using the Service Control Panel. In addition, you will check the availability of some required Integration Server packages and configure your development environment preferences.
In this exercise and all subsequent exercises, please replace:
<workshop_dir> by **C:\Training\622-51E**

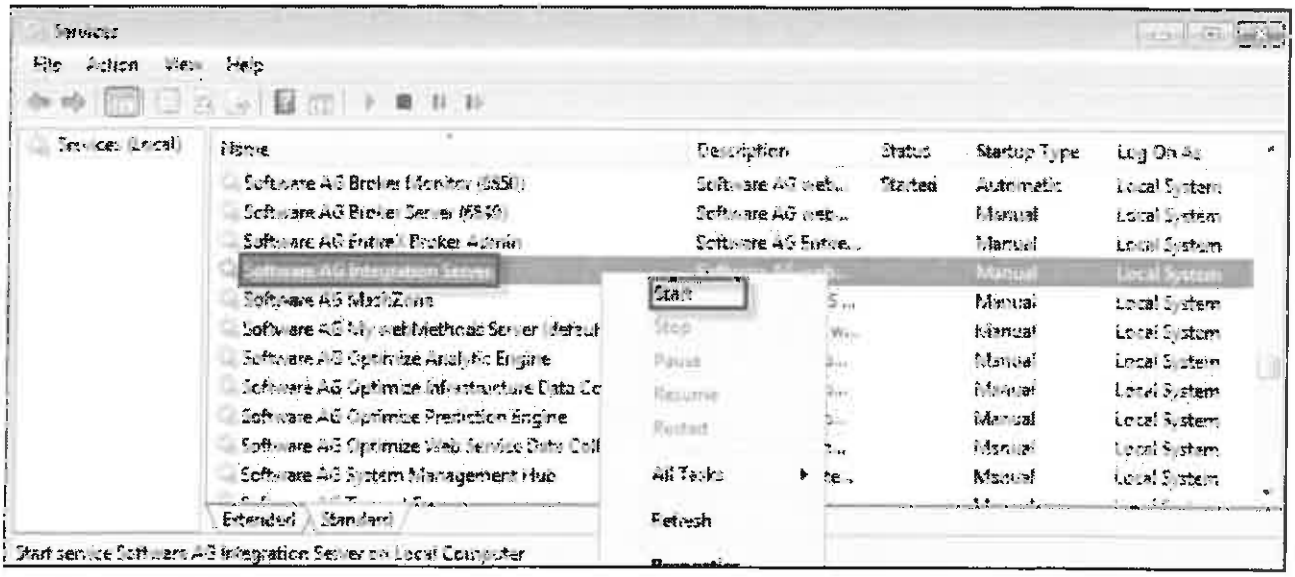
Steps

1. Ensure that the time zone and time setting of your VM mirrors your local time zone and time setting. If necessary, double-click the clock in the lower right corner of the Windows task bar to open the Windows Date and Time Properties of your VM, and adjust the settings.
2. Launch the Windows Services Control Panel to verify the configuration of Windows services.

Verify that the following Windows services are available and in the following state:

SQL Server (SQLEXPRESS)	started
SQL Server VSS Writer	started
Software AG Broker Monitor (6850)	started
Software AG Broker Server (6849)	started
Software AG Integration Server	not started
Software AG My webMethods Server (default)	not started
Software AG Optimize Analytic Engine	not started
Software AG Optimize Infrastructure Data Collector	not started
Software AG Optimize Prediction Engine	not started
Software AG Optimize Web Service Data Collector	not started

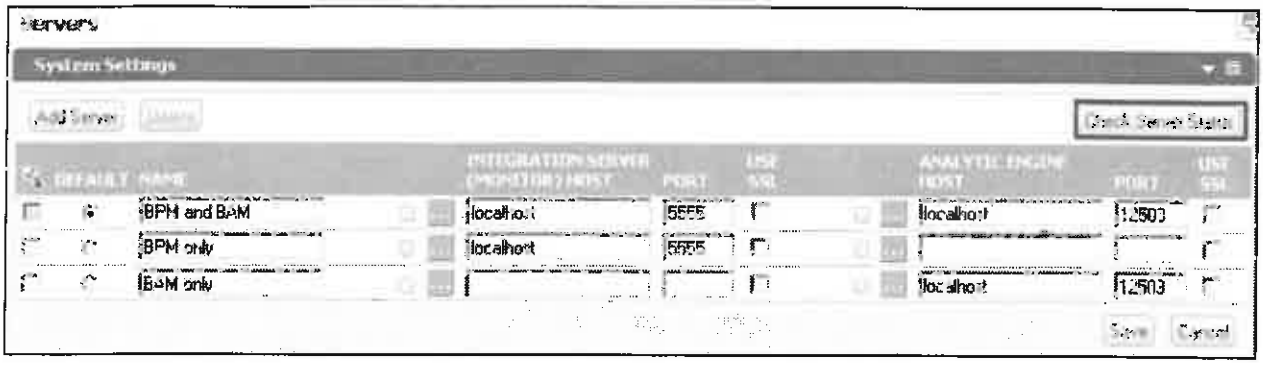
3. Use the Service Control Panel to start the **Software AG Integration Server** as a Windows service:



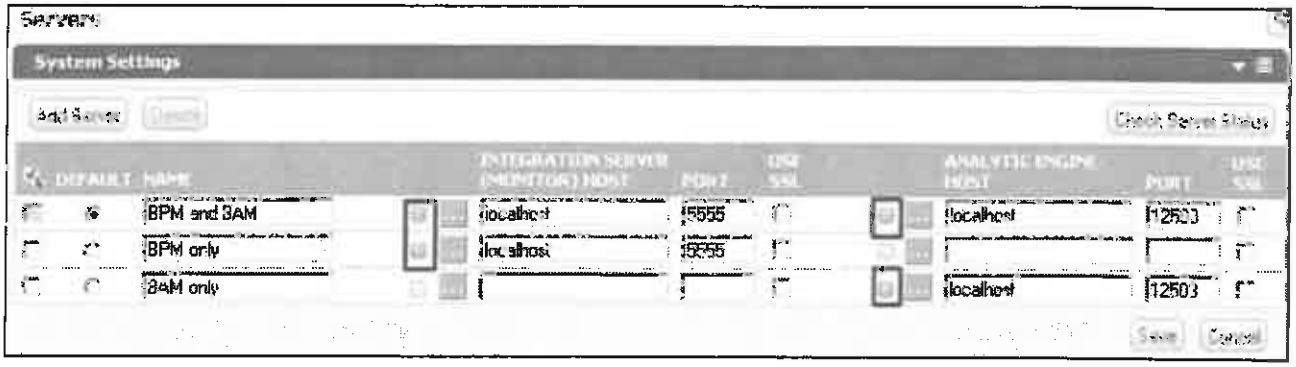
4. Similar, use the Service Control Panel to start the **Software AG My webMethods Server** and the **Software AG webMethods Optimize Analytic Engine** as Windows services.

5. Open a browser and login to My webMethods using the URL **http://localhost:8585**. Provide **Administrator/manage** for authentication.
(*Note*: It may take a while until MWS has completed its startup and is able to serve the URL from above).

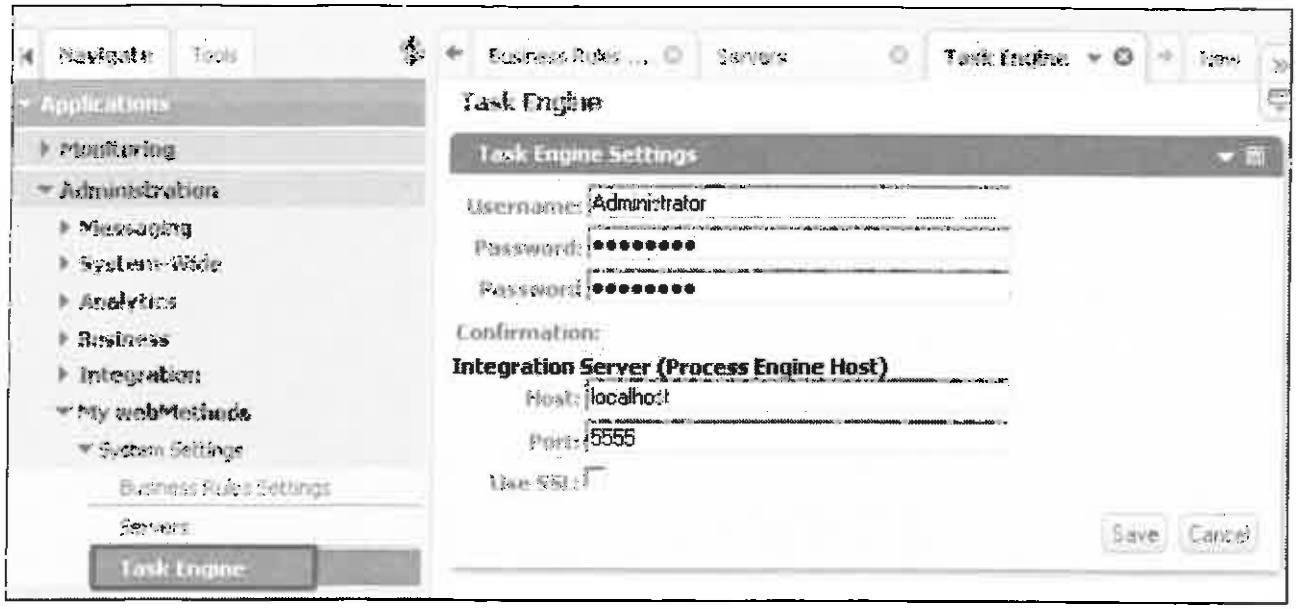
6. Select the Navigate tab and drill down to **Applications -> Administration -> My webMethods -> System Settings -> Servers**. Confirm the Integration Server and Analytic Engine configurations match the following details:



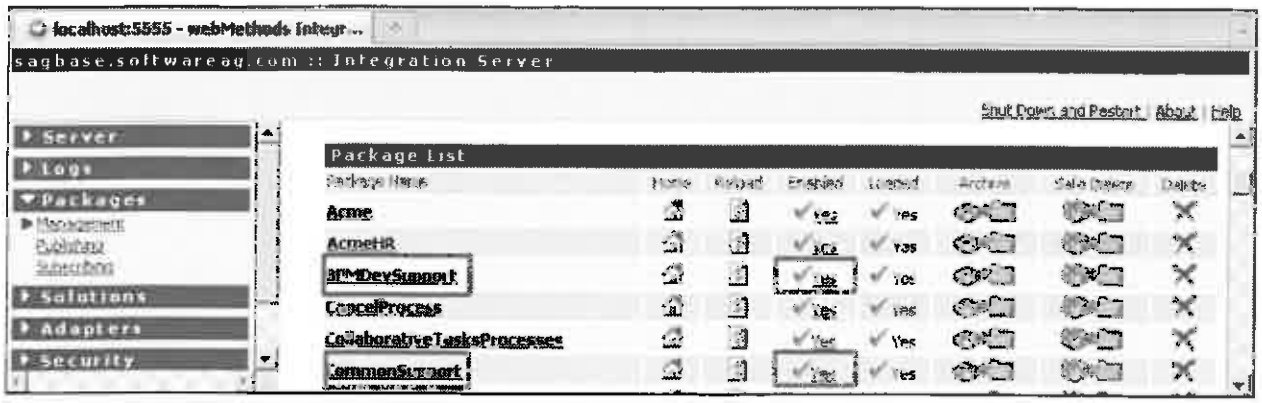
Use **Check Server Status** to check the configuration:



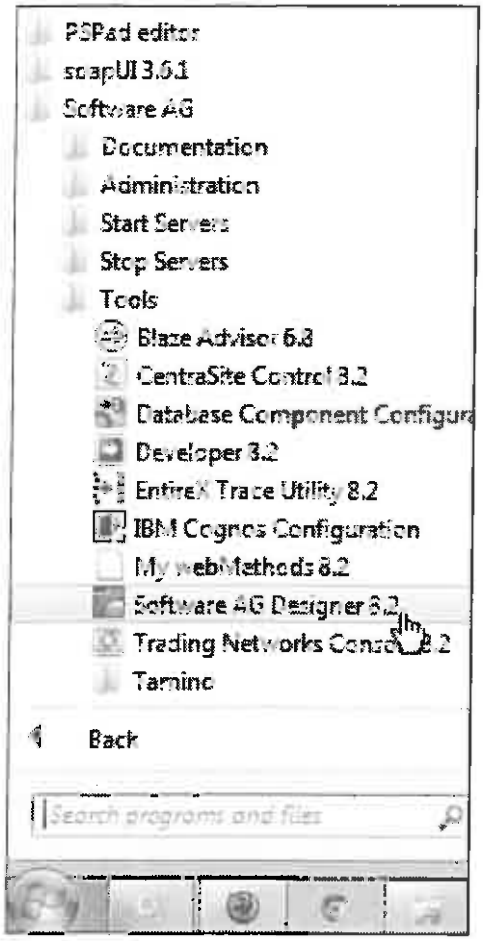
7. Drill down to **Applications -> Administration -> My webMethods -> System Settings -> Task Engine**. Confirm the Task Engine settings match the following details (password is **manage**):



8. Use a browser and URL **http://localhost:5555** to open the Integration Server Administration console. Use **Administrator/manage** for authentication.
9. In the IS Administration console visit **Packages -> Management**. Ensure that the packages **BPMDevSupport** and **CommonSupport** are enabled. If they are disabled, click the appropriate link in the Enabled column to enable them.

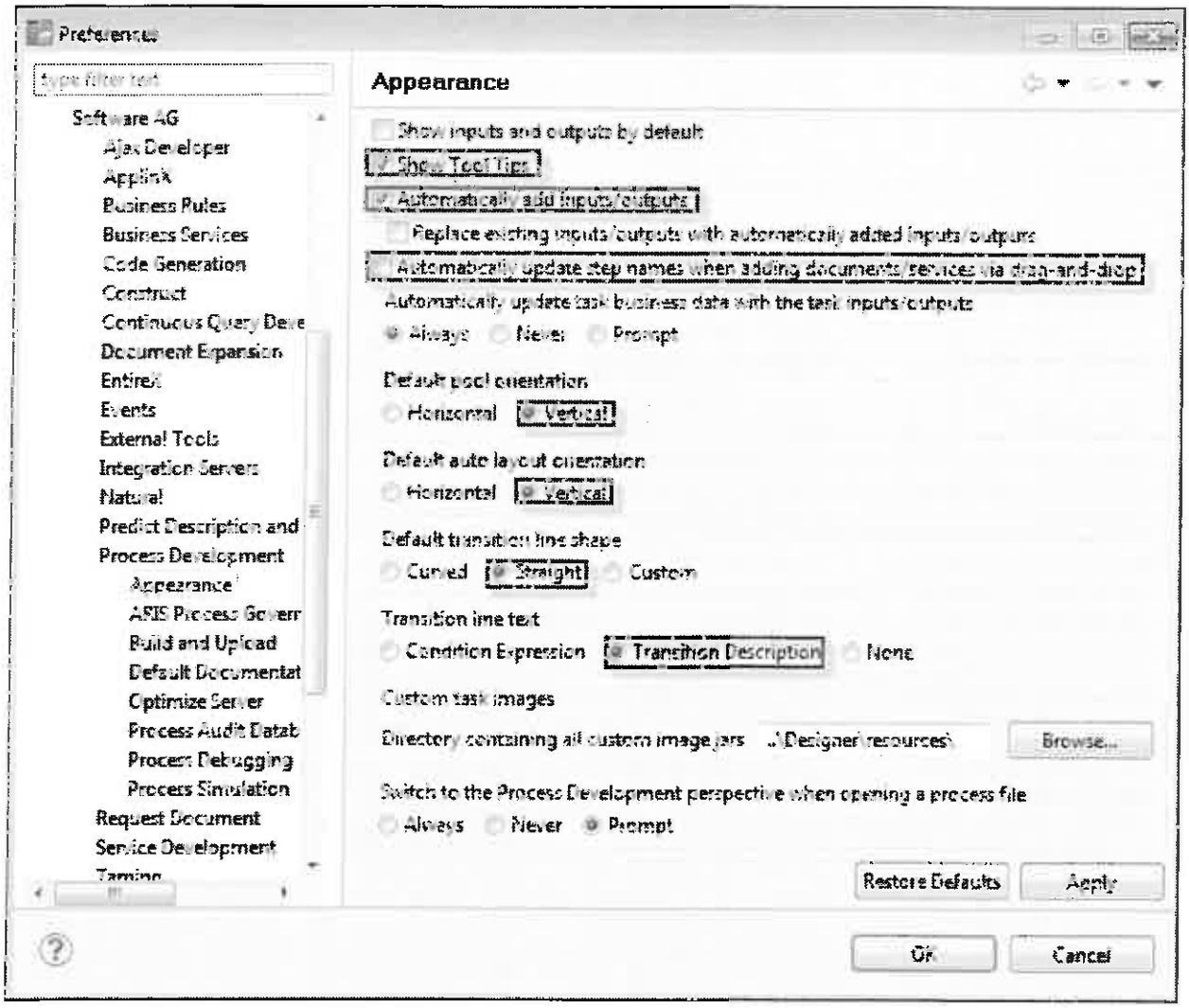


10. Launch Software AG Designer.

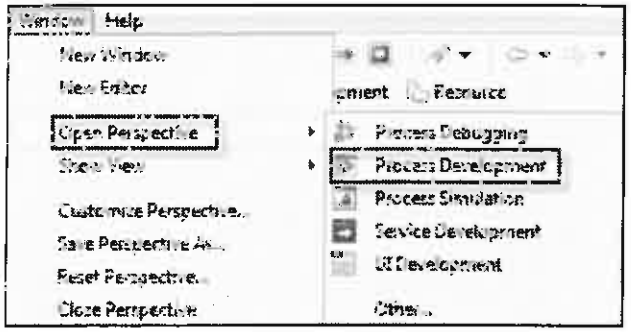


Accept the default workspace directory by clicking the OK button.

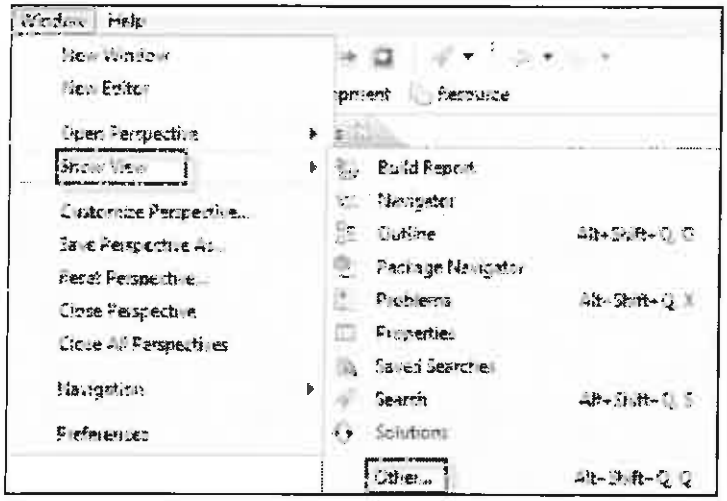
11. In Designer, update the Process Development Appearance preferences by clicking **Window -> Preferences -> Software AG -> Process Development -> Appearance.**
Set up the Appearance Preferences by ensuring that **Vertical** is selected for **Default pool orientation** and **Default auto layout orientation.** Choose **Straight** transition lines as **Default transition line shape** and select **Transition Description** for **Transition line text.**
Also, enable **Show Tool Tips, Automatically add inputs/outputs** and disable **Automatically update step names when adding... .**

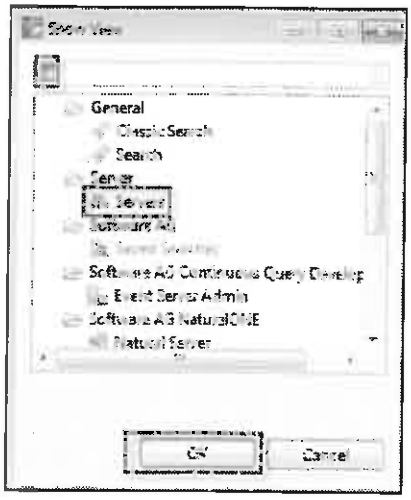


12. If necessary, switch to the **Process Development** perspective:

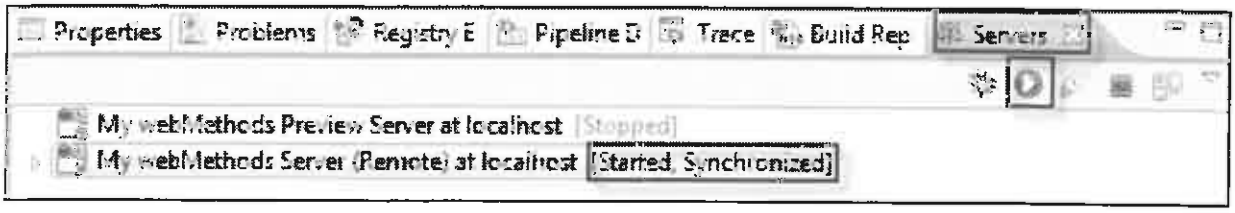


13. Open the **Servers** view, if it is not already open:





14. In the Servers view, two My webMethods Servers are already preconfigured. Select the My webMethods Server marked as **Remote** you have started as a Windows service. Click the green start button. If asked for authentication, provide **Sysadmin/manage** as user credentials. When clicking the green start button, Designer detects that the MWS is running and shows its State as **Started, Synchronized.**



15. In Designer, open the **Package Navigator** view. If not connected yet, connect to your **Default** Integration Server. If asked for authentication, provide **Administrator/manage** as user credentials.
    Ensure that packages **BPMDevSupport** and **CommonSupport** are available for later process development.

Check Your Understanding

1. What is the URL for the Integration Server Administration console?
   Http://localhost:5555

2. Which packages did we enable to be used by the Integration Server for this course?
   Dev Support and Common Support

3. Does the Start Server button in the Servers view always launch a new server instance?
   No

Exercise 2:
Creating an Automated Business Process

Overview

In this exercise, you will create a new business process named **HandleNewOrder** from scratch. You will add pools and BPMN steps to the process. The **HandleNewOrder** process handles purchase orders submitted by customers to the Acme Corp.

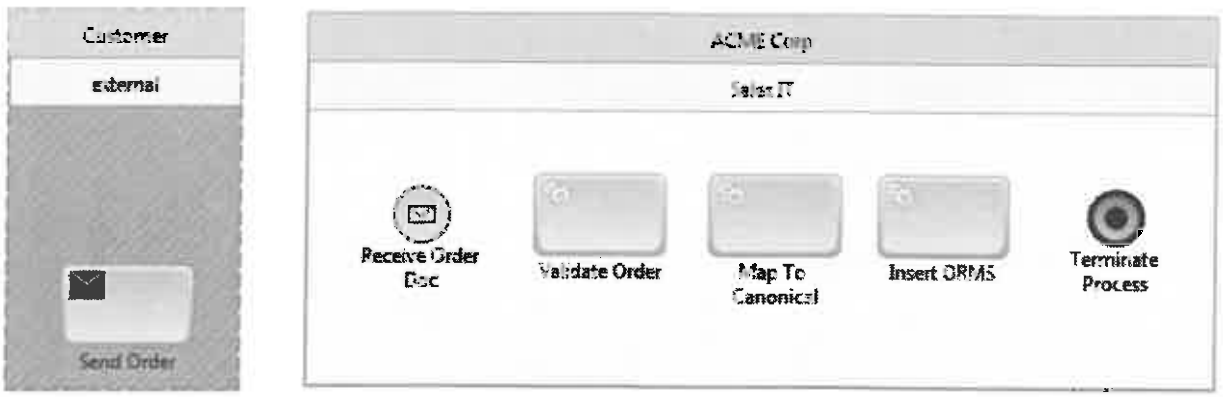
Note:

This process model will be enhanced step by step by the subsequent exercises.

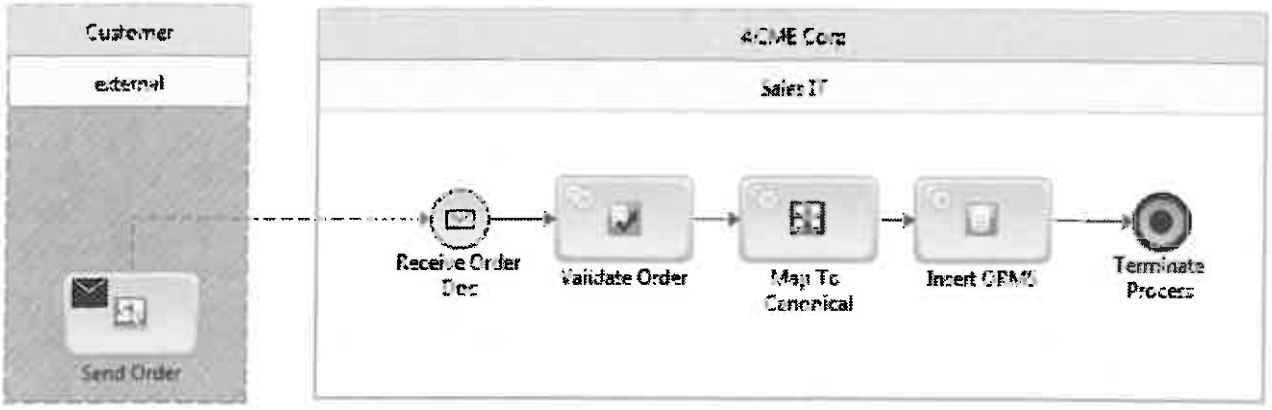
If you are unable to complete all of the steps of any given exercise within the course, refer to the Appendix at the end of this document for instructions how to obtain the solution to an exercise so that subsequent exercises can be completed.

Steps

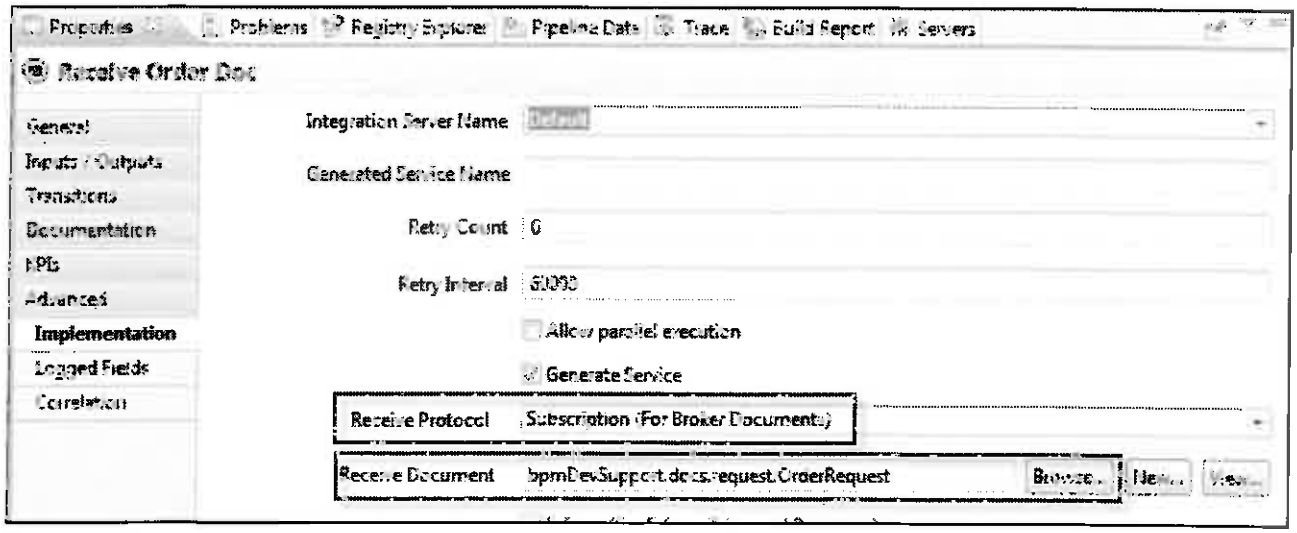
1. Ensure both Integration Server and My webMethods Server are running as Windows services.
2. Launch Software AG Designer and ensure you are in **Process Development** perspective.
3. In the **Solutions** view, right-click on Processes and select **New Process Project**. Name the new project **CorporateProcesses**. Use the default location and select **Finish**.
4. From the Solutions view right-click the CorporateProcesses project and select **New Process**. For the Process Name enter **HandleNewOrder** and select **CorporateProcesses** for the Process Project. Click the **Finish** button.
5. Enable the **Process Developer mode** by clicking the  icon in the menu bar.
6. Using the Palette, add an internal pool **ACME Corp** to your process. Rename its default swimlane to **Sales IT**.
7. Add an external pool called **Customer** to the left. Rename its internal swimlane to **external**. Change the color of the swimlane in the external pool to orange.
8. Add a Start Message Event named **Receive Order Doc** to your internal pool. Add three Service Task Activity steps named **Validate Order**, **Map To Canonical**, and **Insert ORMS** to your internal pool. Add a trailing End Terminate Event named **Terminate Process** behind Insert ORMS. In the external pool, add a Send Task Activity named **Send Order**.
   Your process model should now look like this:



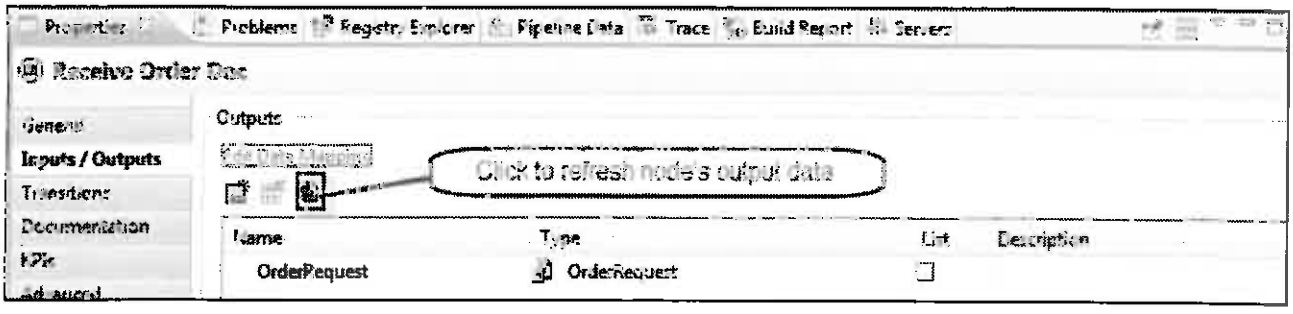
9. Add step images and add transitions to correspond to the following screen shot:



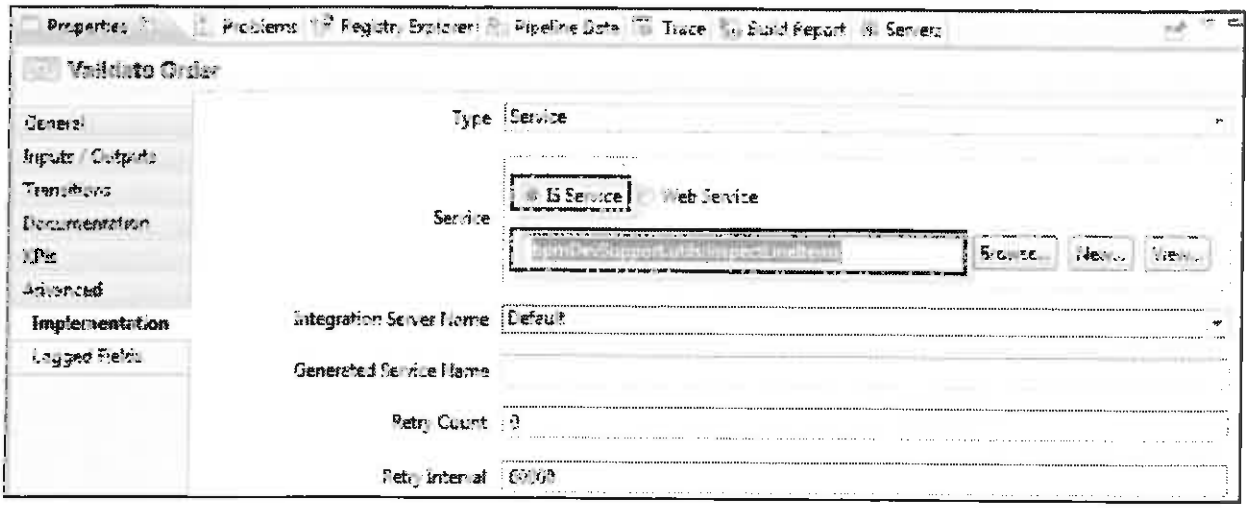
10. Configure the implementation properties of the **Receive Order Doc** step to receive a document of type **bpmDevSupport.docs.request:OrderRequest** using Broker subscription. *Note*: You can either browse for the IS document or use drag the document from the Package Navigator view onto the step in the design canvas.



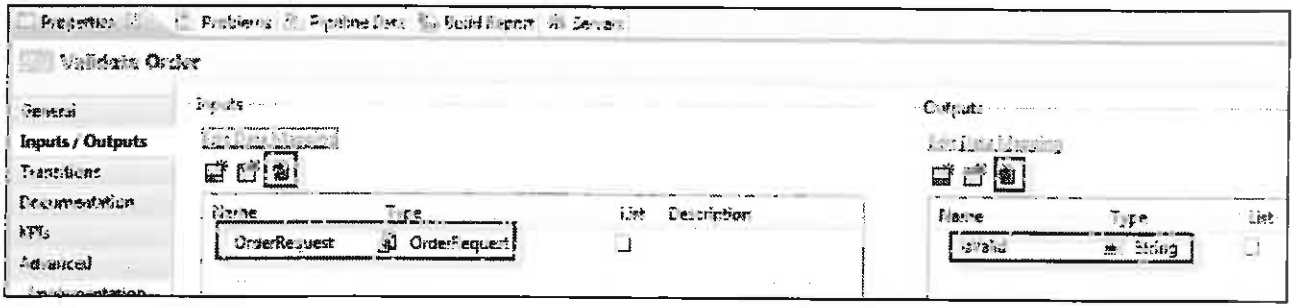
Ensure that the document is assigned as output in the steps Inputs/Outputs properties.



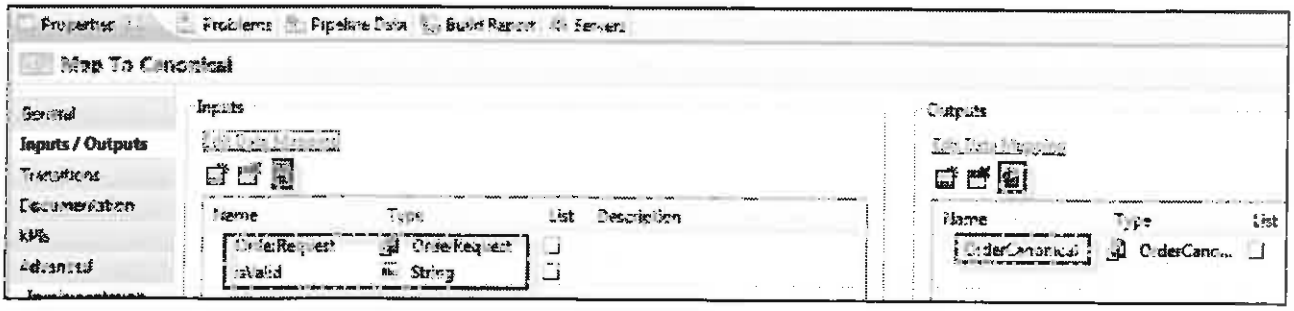
11. Setup the Service Task Activity **Validate Order** to invoke the provided IS service **bpmDevSupport.utils:inspectLineItems.**
To do so, drag and drop the IS service from the Package Navigator view onto the step in the design canvas. This will adjust the implementation type and service name in the Implementation tab of the Validate Order step properties:



Then select **Inputs/Outputs** in the steps Properties view. In the Inputs section, ensure that the inputs were set automatically (if not, click the Refresh button to refresh the inputs from the service signature). You should also make sure that the outputs were set automatically (if not, click the Refresh button to refresh the outputs from the service signature). Input and output data should look like this:

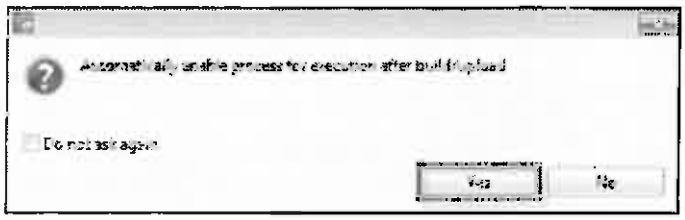


12. Next, setup step **Map To Canonical** to invoke the provided IS service **bpmDevSupport.maps:OrderRequestToCanonical**. Check that inputs and outputs were set automatically, otherwise refresh them from the service signature.
The Inputs/Outputs tab within the Properties view should look like this:

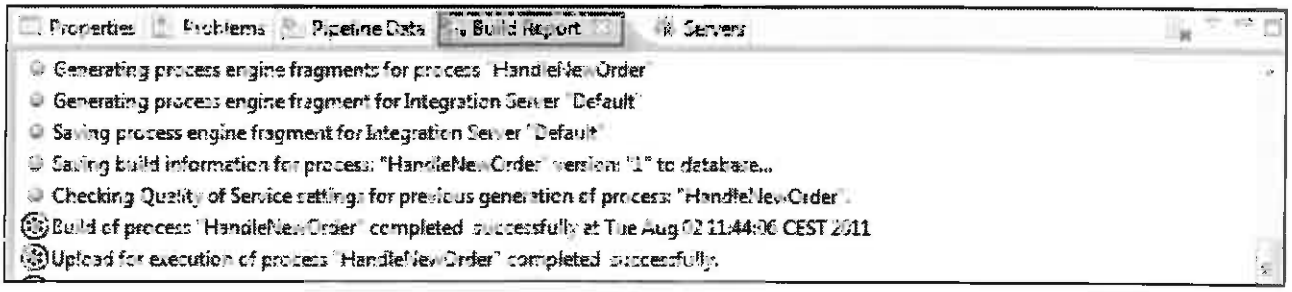


13. Finally setup the step **Insert ORMS** to invoke the provided IS service **bpmDevSupport.utils:insertOrderCanonical**.
    After checking/refreshing inputs and outputs, the step should have a reference to the OrderCanonical document as input; there are no outputs for this step.

14. Click to Build and Upload the project (*Note*: To make the build button enabled, you have to click on the tab of the opened process). When asked for saving or enabling the process for execution, always confirm with Yes.



15. Use the Build Report view to ensure the project builds successfully with no errors:



Check Your Understanding

1. What was the purpose of adding the external pool?
   Represent non-organic processes.

2. What was the purpose of the "Receive Order Doc" step?
   Receives the order from external source and initiated the process

3. After the "Terminate Process" event is executed, what happens to the process instance?
   The process closes till called again

Exercise 3:
Generate Process Documentation

Overview

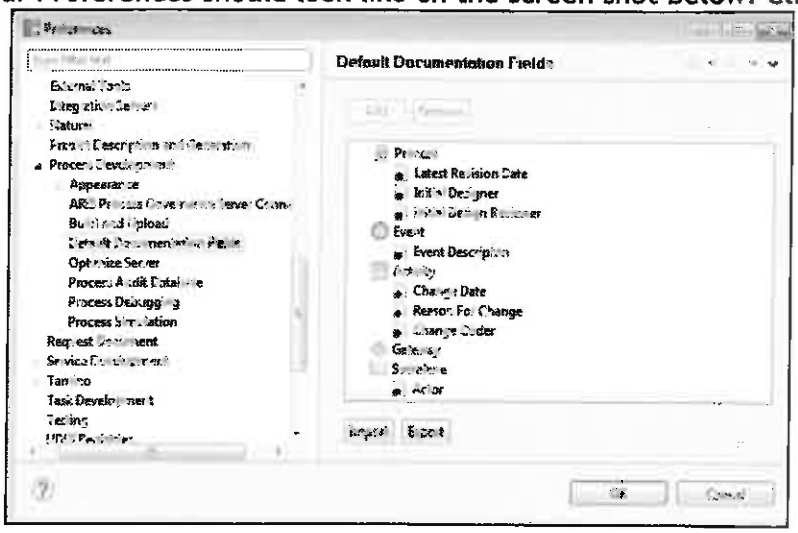
In this exercise, you will document your HandleNewOrder process and create a technical documentation report. This can be used to talk to Business Analysts.

Steps

1. Ensure both Integration Server and My webMethods Server are running as Windows services.
2. Launch Software AG Designer.
3. Open Software AG Designer preferences. Drill down to **Software AG -> Process Development -> Default Documentation Fields.** Add the following documentation fields to the categories Process, Event, Activity, and Swimlane:

Category	Documentation Fields
Process	Latest Revision Date
	Initial Designer
	Initial Design Reviewer
Event	Event Description
Activity	Change Date
	Reason For Change
	Change Coder
Swimlane	Actor

Your Preferences should look like on the screen shot below. Click **OK** to save your entries.



4. Open the Properties view of your **HandleNewOrder** process. Use the **Documentation** tab to document the process with the following details:

Documentation Field	Value for the Field
Latest Revision Date	*"Current Date"*
Initial Designer	John Smith
Initial Design Reviewer	Bill Campbell

Additionally, visit the **General** tab to add the Description text **Receives and saves a new order** to your process model.

5. For each step in the HandleNewOrder process, set the **Description** property as shown in the following table:

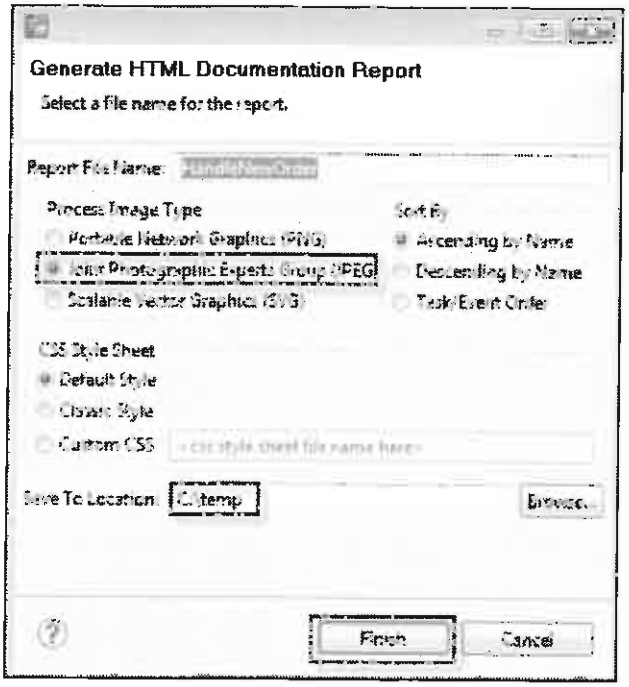
Step	Description
Receive Order Doc	Receives OrderRequest document
Validate Order	Validates purchase order
Map to Canonical	Maps OrderRequest to OrderCanonical
Insert ORMS	Inserts canonical order into database
Terminate Process	Terminates process with status "Completed"

6. Open the **Inputs/Outputs** tab of the **Validate Order** step. Add the following descriptions to the input/output fields:

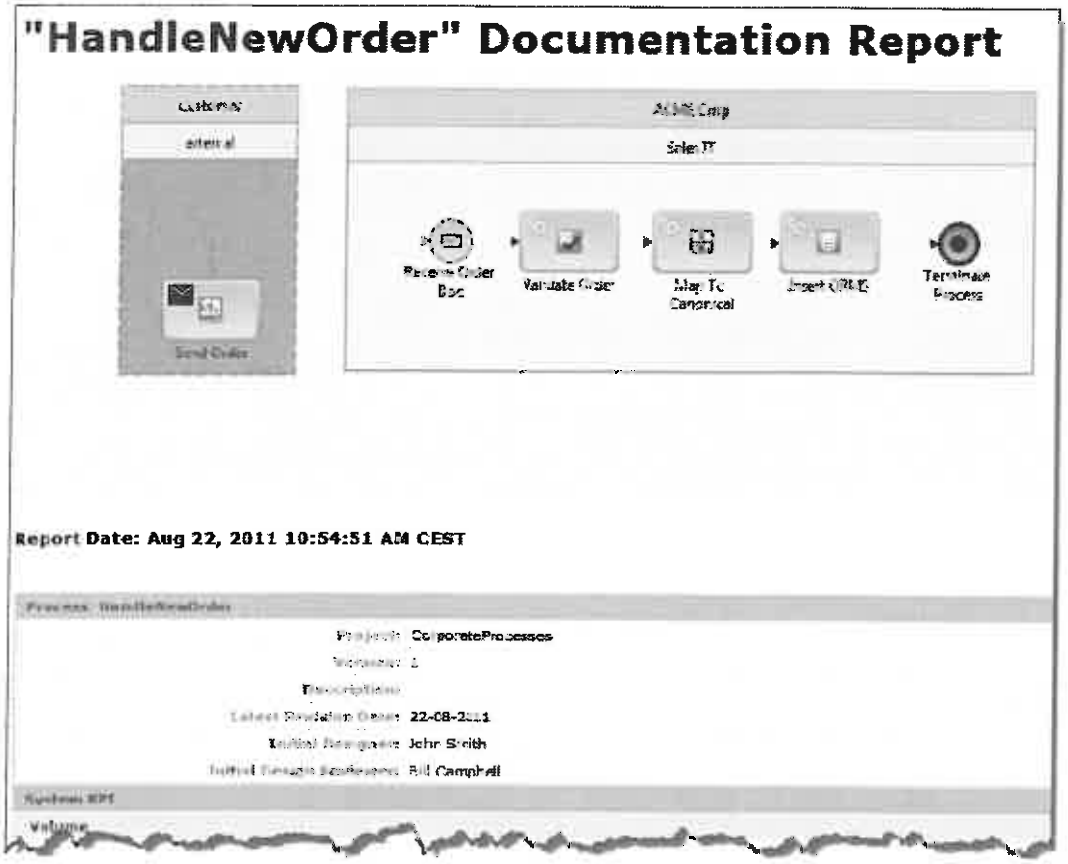
Field name	Input/Output	Description
OrderRequest	input	Purchase order document
isValid	output	true or false

7. Save your process.

8. Select **Generate Documentation Report...** from the File menu to generate an HTML documentation report for the HandleNewOrder process. Set the **Process Image Type** to **JPEG**, and browse for the Save To Location **C:\temp**:



9. Open your documentation report stored **C:\temp\HandleNewOrder.html**. Browse the different sections of the report.



Check Your Understanding

1. What is the format of the generated documentation?

2. Where or when would the generated documentation be used?

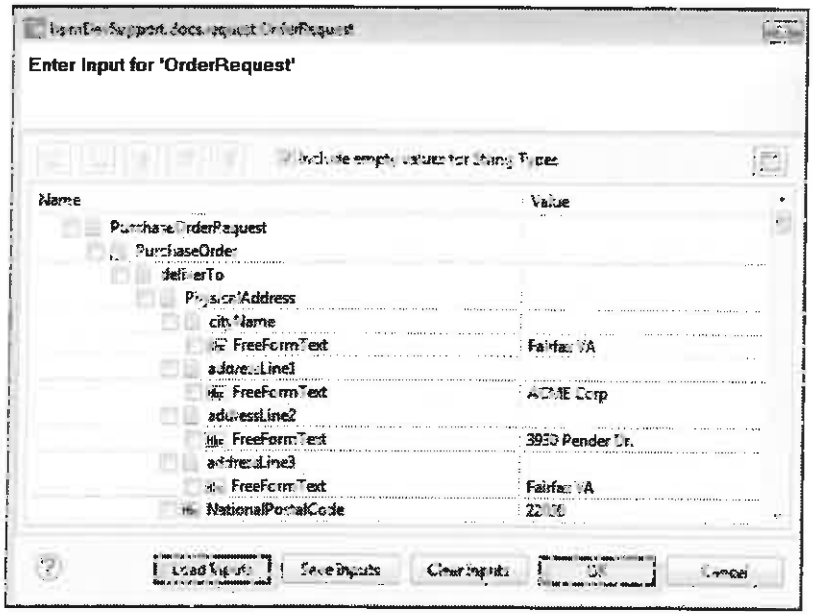
Exercise 4:
Process Debugging

Overview

After creating and building the HandleNewOrder process in the previous exercises, you will now test and step through a process using the built-in Debugger in Software AG Designer.
To double-check that the data is persisted in the database you will also inspect the related database tables for new order entries.

Steps

1. Ensure both Integration Server and My webMethods Server are running as Windows services.
2. Launch SoftwareAG Designer and ensure you are in **Process Debugging** perspective.
3. After building and uploading the **HandleNewOrder** process, click the **Debug Selected Process** icon to start a debugging session.
   Note: If the Debug Selected Process button is not enabled, click anywhere inside the canvas of the **HandleNewOrder** process model.
4. For the IS input document, load the provided file ***<workshop_dir>*\Exercise4\Resources\Ex4_input.txt** as input data
   Note: When you load the test data for debugging, feel free to update the data, including the date fields with current dates/times. It's not mandatory but your test data will appear more current. If you change any data, remember the changes you made so that you can find it in the server log when testing.
   Click **OK**.

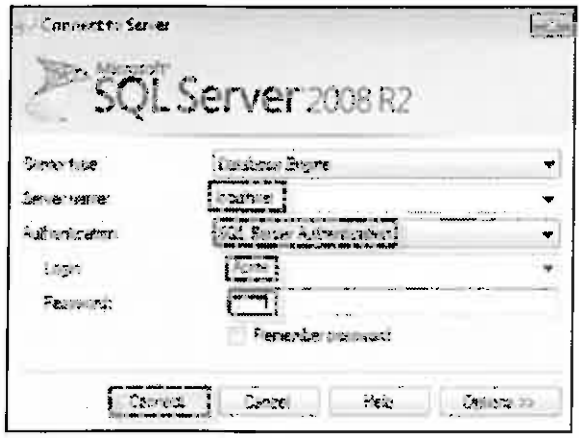


5. Step through the process until its end.
   The provided order is valid; the process should complete successfully and should insert the order into the database.

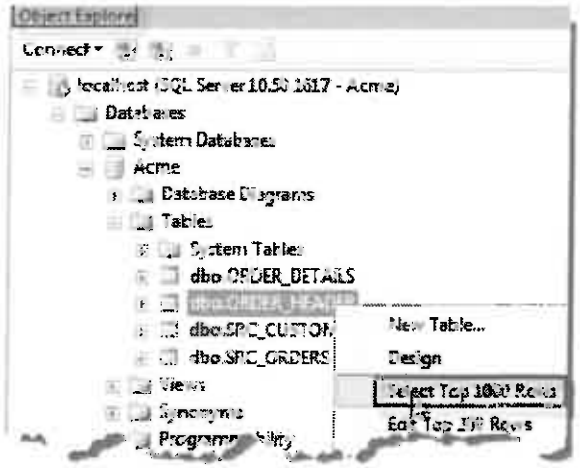
6. Ensure the order was entered successfully into the Acme database:

   a) Start SQL Server Management Studio Express by navigating to **Start -> All Programs -> Microsoft SQL Server 2008 R2-> SQL Server Management Studio.**

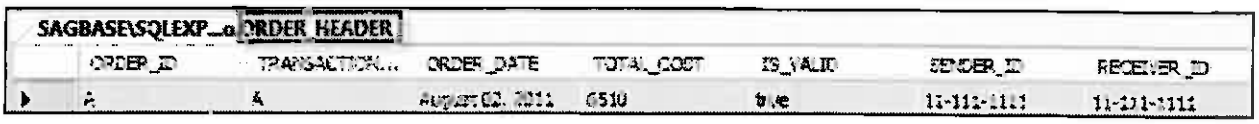
   b) For authentication choose **localhost** as server name, select **SQL Server Authentication**, and provide **Acme/manage** as user credentials:

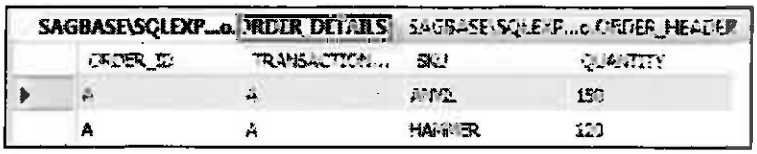


   c) Under the Object Explorer, navigate to **localhost -> Databases -> Acme -> Tables.** To view the contents of the **dbo.ORDER_HEADER** or **dbo.ORDER_DETAILS** table, right-click the table and click **Select Top 1000 Rows:**



   d) Verify that your order has been stored in both tables:





Check Your Understanding

1. Why did you load the file <*workshop_dir*>\Exercise4A\Resources\Ex4A_input.txt?
   To provide data to the process for execution

2. What did performing the SQL table lookup prove?
   That the information was properly stored.

This page intentionally left blank.

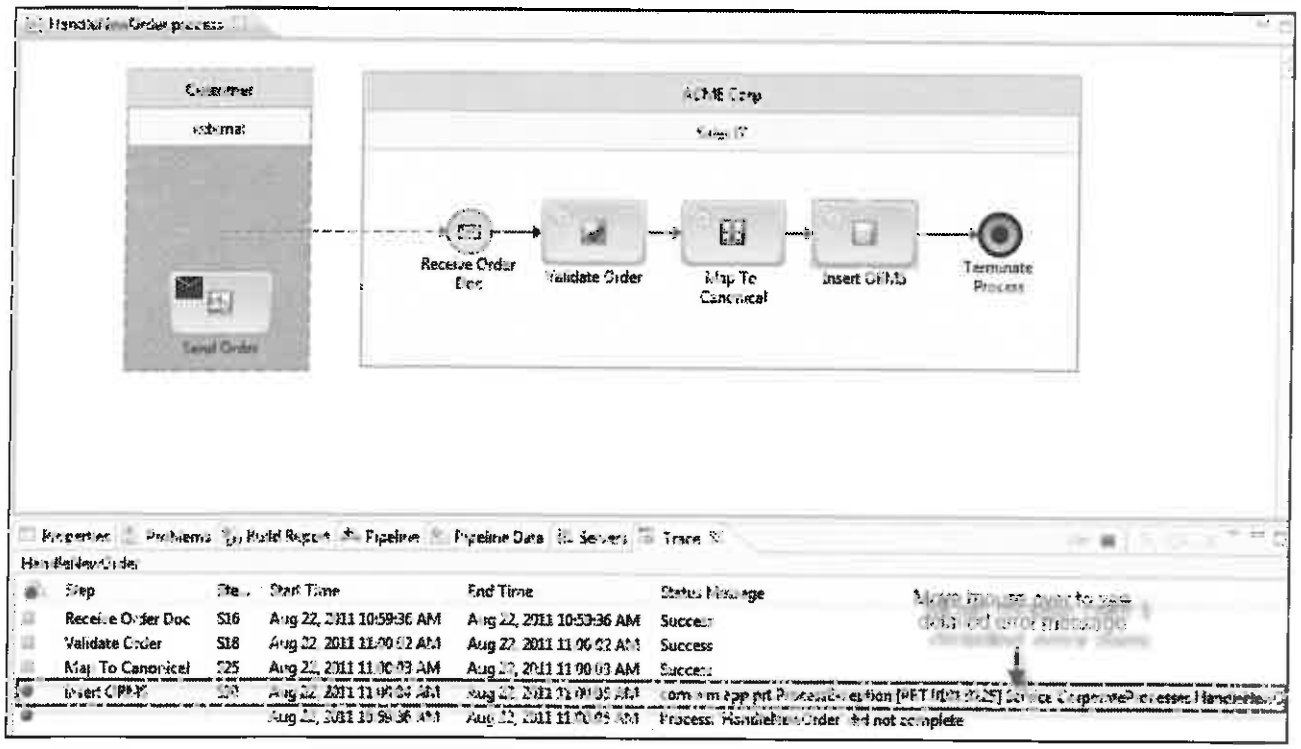
Exercise 5:
Advanced Process Debugging

Overview

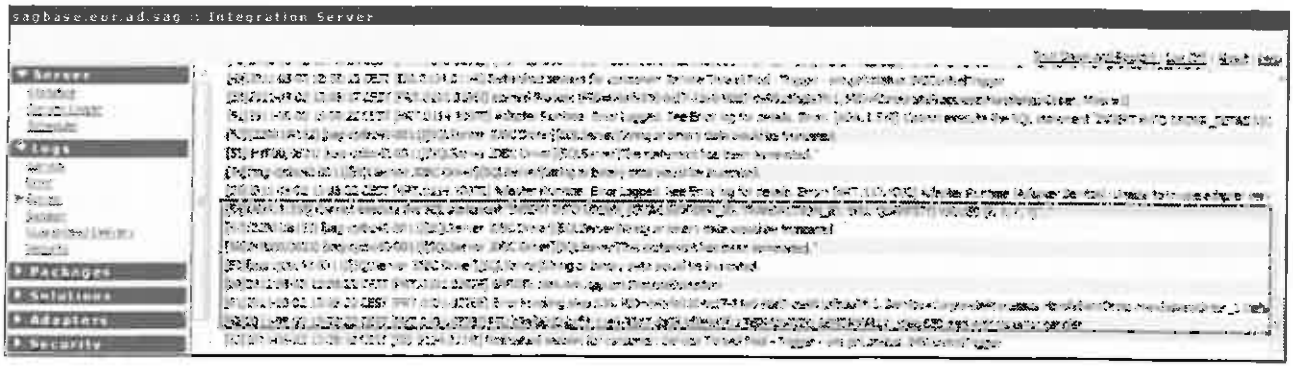
In this exercise, you will again test and debug the HandleNewOrder process using the Debugger contained in Software AG Designer.
In this exercise, an invalid order will be used for debugging.

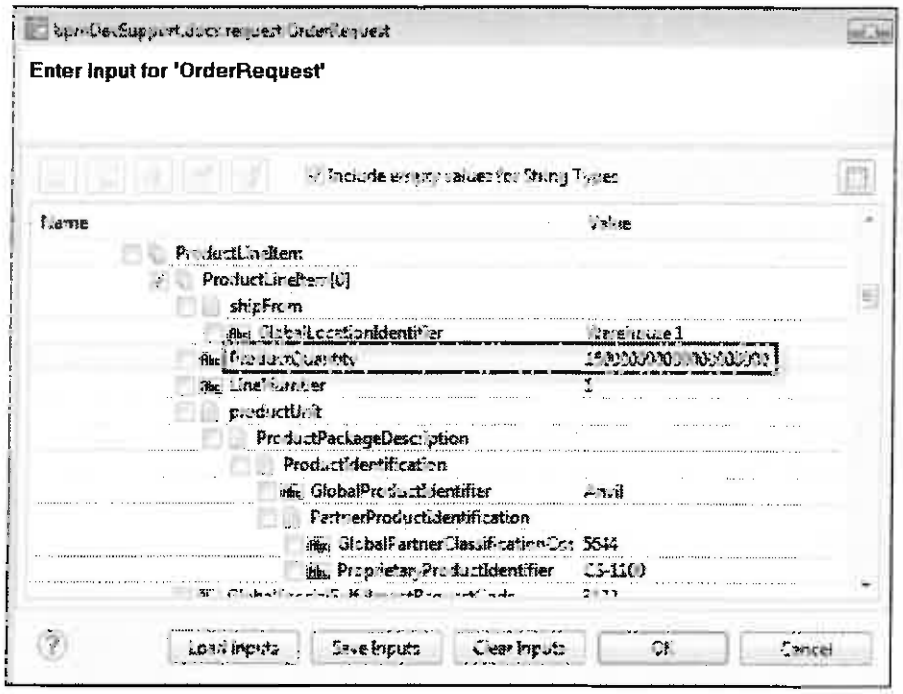
Steps

1. Ensure both Integration Server and My webMethods Server are running as Windows services.
2. Launch Software AG Designer and ensure you are in **Process Debugging** perspective.
3. After building and uploading the HandleNewOrder process, click the **Debug Selected Process** icon to start a debugging session.
4. For the IS input document, load the provided document from the file ***<workshop_dir>*\Exercise5\Resources\Ex5_input.txt**. Step through the process. The provided order contains bad data; the process should stop with an error:



5. Open the Integration Server's **server.log** file to investigate the error. The server.log file can be found in the folder **C:\SoftwareAG\IntegrationServer\logs**.
   As an alternative, you can inspect the **server.log** file by using the IS Administration console (http://localhost:5555).





Check Your Understanding

1. Which log did you use to show the exception details?
   server.log out of IS Administration

2. In the Trace view, how can you easily identify that the process instance failed?
   The point of failure is highlighted.

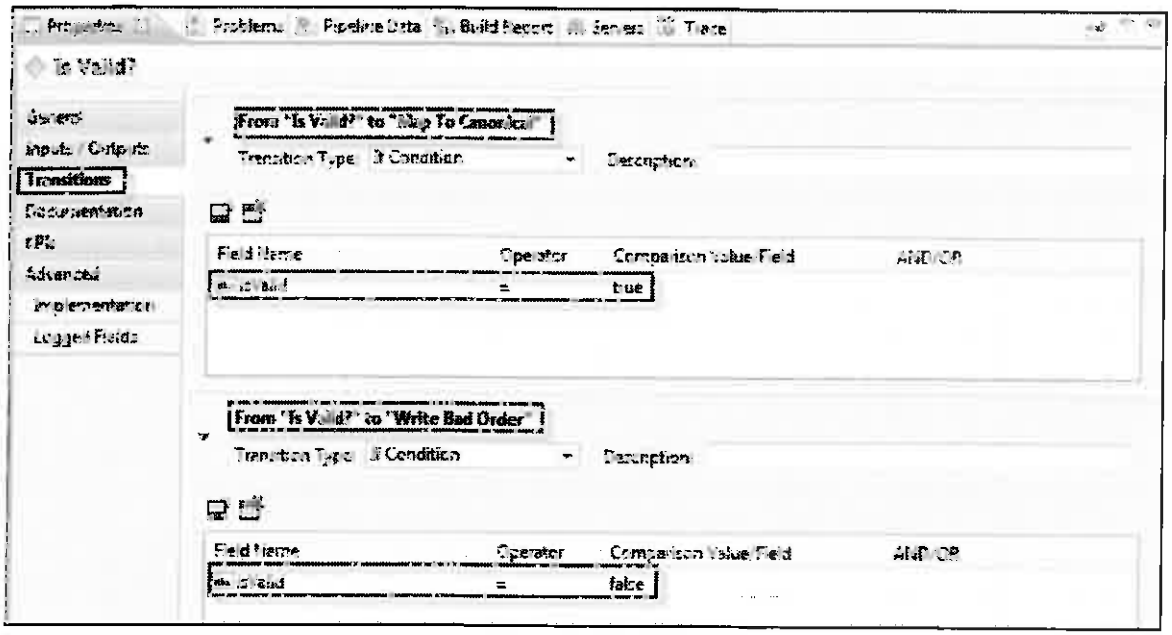
Exercise 6:
Adjust the Flow of Data

Overview

In this exercise, you will add a Gateway with conditional transitions to handle invalid orders. Moreover, you will test the process in various ways: by debugging, publishing an IS document, and starting a process instance from a web form.

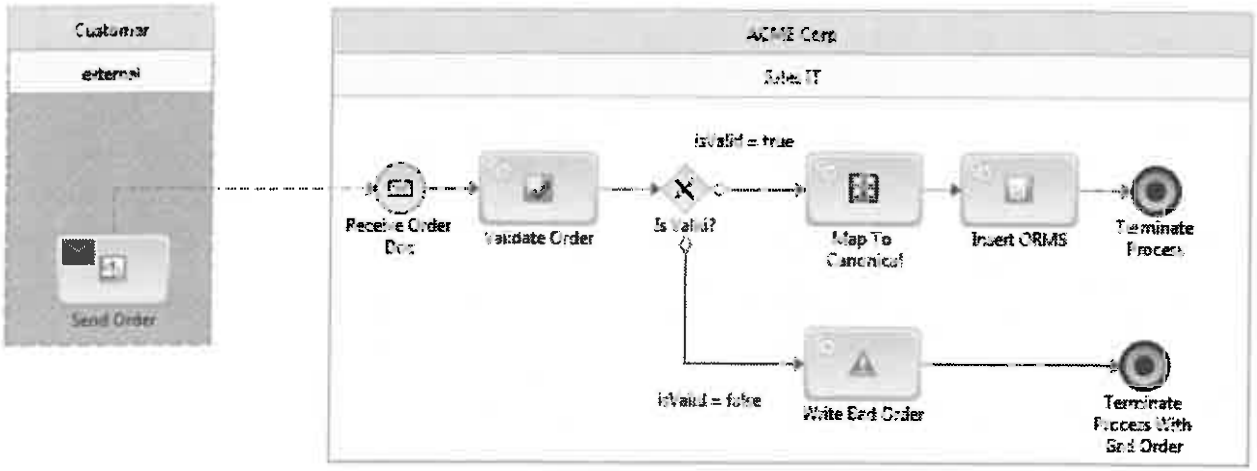
Steps

1. Ensure both Integration Server and My webMethods Server are running as Windows services.
2. Launch Software AG Designer and ensure you are in the **Process Development** perspective.
3. Remove the transition between **Validate Order** and **Map To Canonical.**
4. Insert an Exclusive Gateway and position it between **Validate Order** and **Map To Canonical.** Rename its label to **Is Valid?**
5. Draw a transition from the **Validate Order** step to the **Is Valid?** Gateway.
6. Add a new Service Task Activity called **Write Bad Order** to write an invalid incoming order to the file system. Set up the **Write Bad Order** step to invoke the IS service **bpmDevSupport.utils:writeDocToFile**. Remember to check the inputs and outputs from the Input/Output tab of the Properties View (click icon, if necessary).
7. Draw an outgoing transition from the **Is Valid?** step to the **Map To Canonical** step. Draw a second outgoing transition from the **Is Valid?** step to the **Write Bad Order** step.
8. Select the Transitions tab in the Is Valid? step properties. Add the following transition conditions:
   - For the transition from **Is Valid?** to **Map to Canonical**: if isValid = true
   - For the transition from **Is Valid?** to **Write Bad Order**: if isValid = false



9. Add another trailing End Terminate Event named **Terminate Process With Bad Order** to the right of the **Write Bad Order** step. Draw a transition from the **Write Bad Order** step to the **Terminate Process With Bad Order** event.

10. Add an image to the **Write Bad Order** step. The HandleNewOrder process should now correspond to the following image:



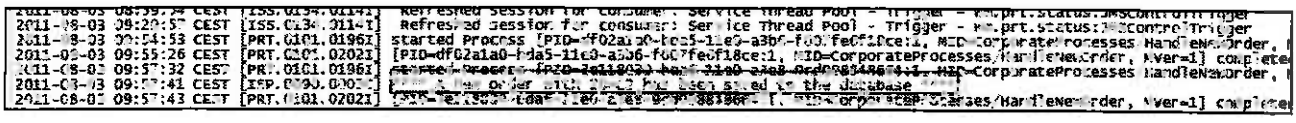
11. After building and uploading the HandleNewOrder process, click the **Debug Selected Process** icon to start a debug session.

12. For the IS input document, load the provided file ***<workshop_dir>*\Exercise6\Resources\Ex6_input1.txt**. Use the Debugger to step through the process. The order is invalid; the process should execute the route **Receive Order Doc -> Validate Order -> Is Valid? -> Write Bad Order -> Terminate Process With Bad Order.**

13. Examine the log file created in the folder C:\temp.

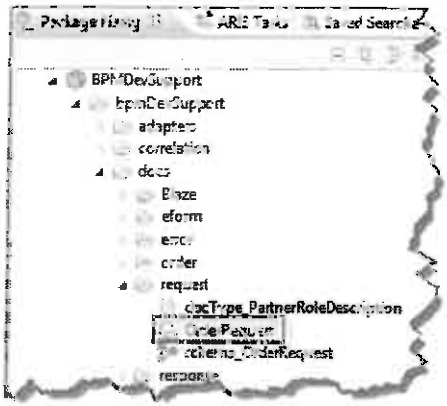
14. Start another debugging session, saving the current process if required, and load the input document from the file ***<workshop_dir>*\Exercise6\Resources\Ex6_input2.txt**.
Step through the process. The order is valid; the process should execute the route **Receive Order Doc -> Validate Order -> Is Valid? -> Map To Canonical -> Insert ORMS -> Terminate Process.**

15. To confirm your valid order has been received, ensure the message "**** A new order with ID=C2 has been saved to the database ****" is displayed in the Integration Server's **server.log** file.
To do so, open the IS **server.log** file by using an editor or by using the IS Administration console.

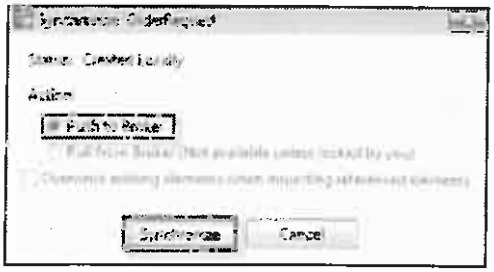


16. Start another **HandleNewOrder** process instance by publishing an OrderRequest document to the Broker:

    a) Locate the **OrderRequest** document in the Package Navigator view. If asked for authentication, provide **Administrator/manage** as user credentials:



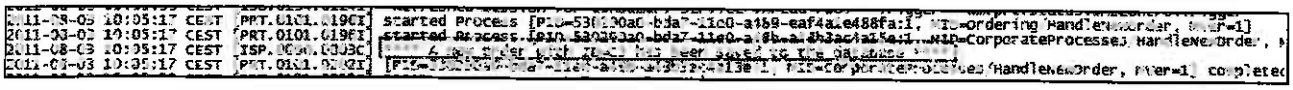
    b) To publish an input document without using the Debugger, ensure the document type has been synchronized with the Broker in advance. Right-click **OrderRequest** in the Package Navigator view and choose **Sync Document Type** from its context menu to synchronize the document with the Broker. On the Synchronize panel select **Push** as the Action and click **Synchronize**:



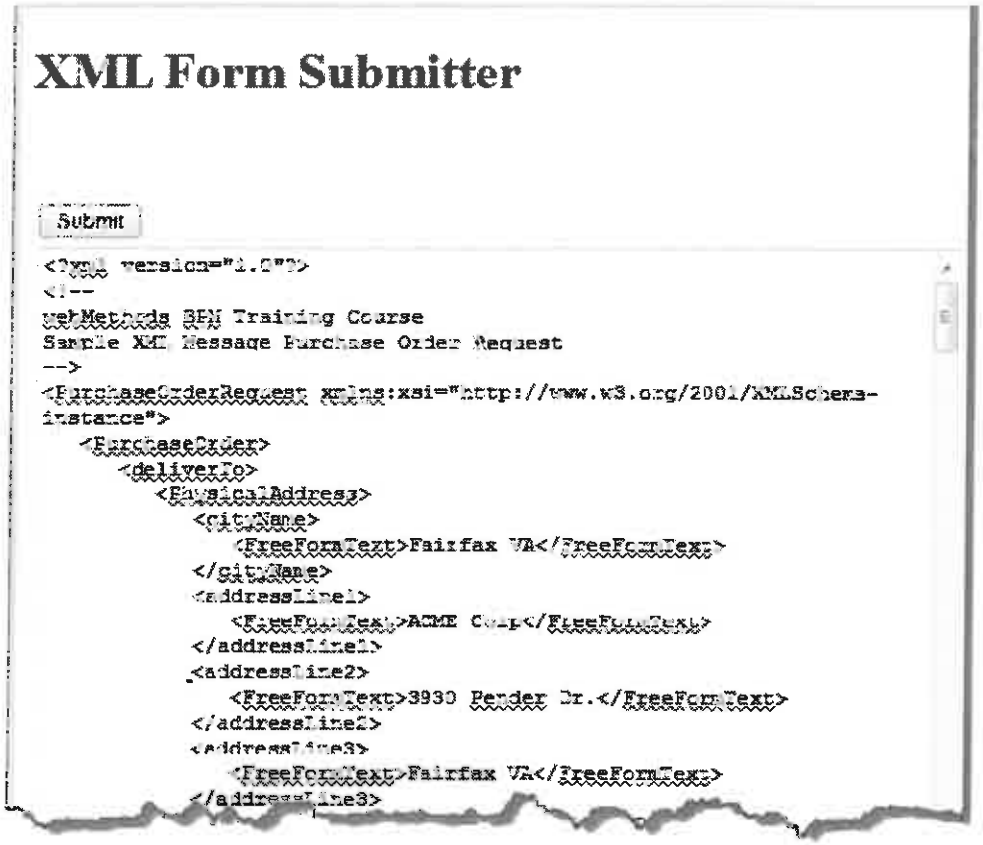
    c) Right-click on **OrderRequest** in the Package Navigator view and select **Run As -> Publishable Document** from the context menu.

    d) For the IS document to be published, load the provided input file ***<workshop_dir>*\Exercise6\Resources\Ex6_input3.txt** and click **OK** to publish to the Broker.

17. To confirm your order has been received, ensure the message "**** A new order with ID=C3 has been saved to the database ****" is contained in the Integration Server's **server.log** file. To do so, open the IS **server.log** file by using an editor or by using the IS Administration console:



18. Finally test your process by starting a **HandleNewOrder** process instance using a web form via a browser. To do so, double-click the provided web from ***<workshop_dir>*\Exercise6\Resources\Ex6_submit.html**. Review the order in the text area and click the **Submit** button. If prompted for IS authentication use **Administrator/manage**.



19. To confirm your order has been received, ensure the message "**** A new order with ID=C4 has been saved to the database ****" contained in the Integration Server's **server.log** file. To do so, open the IS **server.log** file by using an editor or by using the IS Administration console.

Check Your Understanding

1. Which step sets the value of the isValid variable?
   validate order.
2. Which step generated the log file in the c:\temp directory? Write Bad Order.
   ~~Receive Order & Insert ORMS & Terminate all onto log.~~
3. At a high level, describe what happens when the submit button is clicked on the Ex6_submit.html form.
   The webform submits an order to Receive Order Doc, initiating the process.

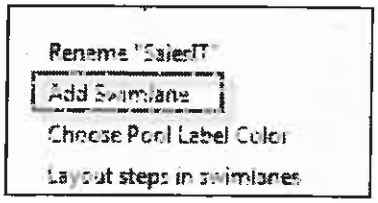
Exercise 7:
Inserting User Interaction

Overview

In this exercise, you will add user interaction by adding a User Task Activity to the **HandleNewOrder** process. A corresponding User Task UI will be generated from scratch and improved in subsequent exercises.
The User Task Activity will be invoked if the submitted order is invalid. In this case and a new user task instance will be created and assigned to the sales department for review.

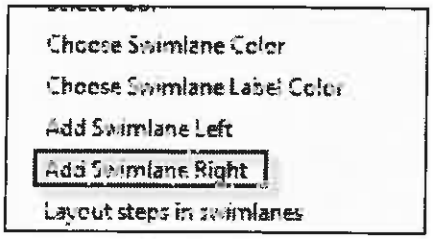
Steps

1. Ensure both Integration Server and My webMethods Server are running as Windows services.
2. Launch Software AG Designer and ensure you are in **Process Development** perspective.
3. Right-click the internal pool and select **Add Swimlane** from the popup menu to add a new swimlane:



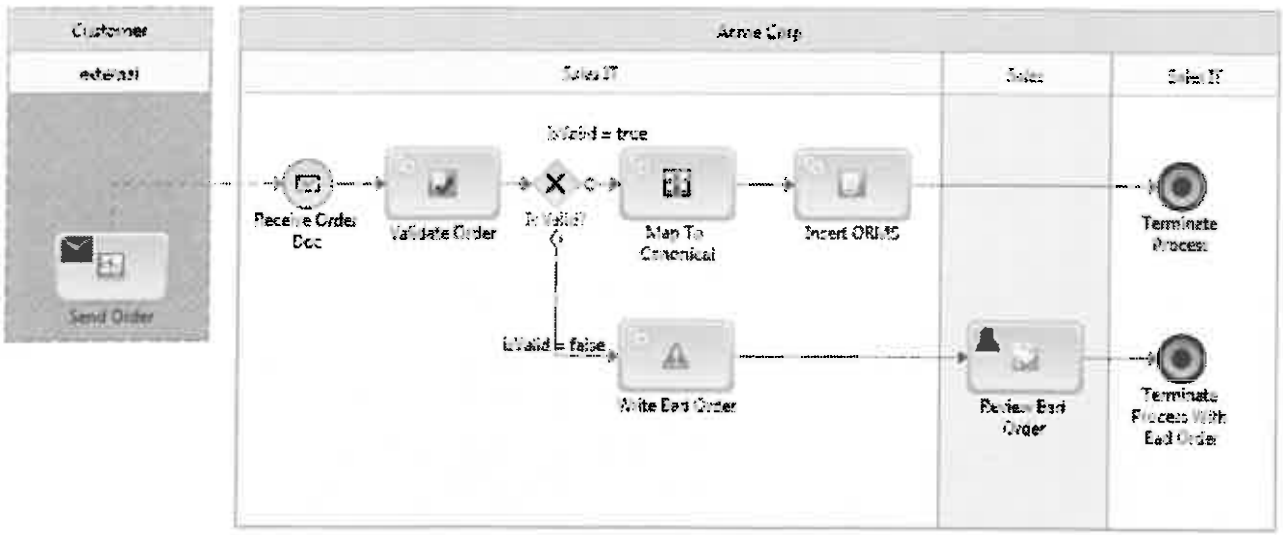
   Name the new swimlane **Sales** and change it's color to turquoise.

4. Click to highlight the new **Sales** swimlane. Right-click swimlane and select **Add Swimlane Right** from the popup menu to add a new swimlane:

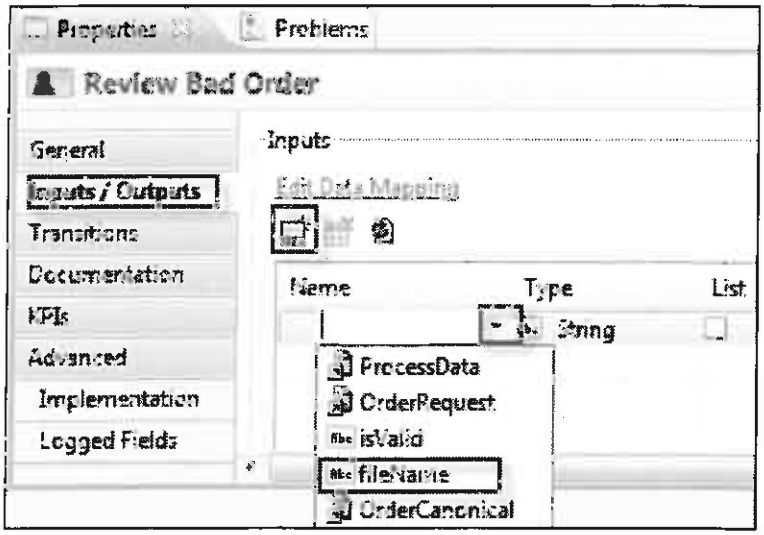


   Name the new swimlane **Sales IT** and choose the same color as for the other existing Sales IT swimlane.

5. Remove the transition from **Write Bad Order** to **Terminate Process With Bad Order.**
6. Move both End Terminate Events from the Sales IT swimlane on the left to the Sales IT swimlane on the right.
7. Add a new User Task Activity to the **Sales** swimlane. Name the step **Review Bad Order.** Add a transition from the **Write Bad Order** step to the new **Review Bad Order** step. Then add a transition from the **Review Bad Order** step to the **Terminate Process With Bad Order** event.
8. Add a image to the **Review Bad Order** step. Your **HandleNewOrder** process should now correspond to the following image:



9. Open the Properties view of the **Review Bad Order** step. Using the Inputs/Outputs tab, click the Create new input button in the Inputs section and select **fileName** of type string as an input for step **Review Bad Order**.
   Note: Variable **fileName** has been added to the pipeline as the return parameter of the previous Write Bad Order step.

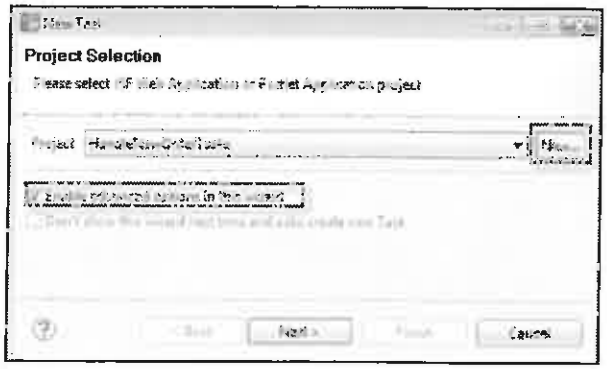


10. Save your changes.

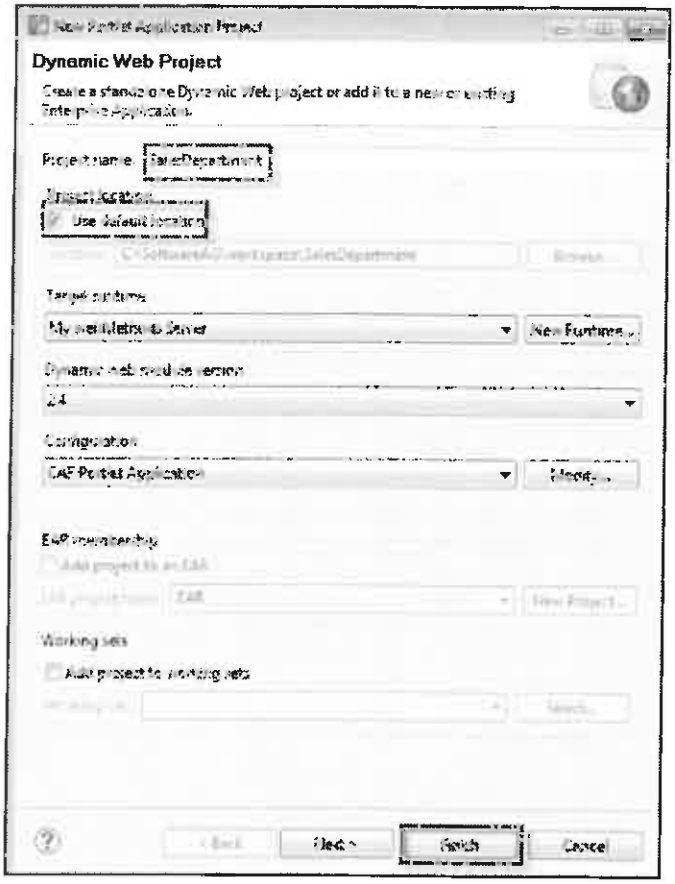
11. Create a new User Task implementation for the **Review Bad Order** step:

    a) Right-click the **Review Bad Order** step in the design canvas and choose **Create Implementation -> New User Task...** from the context menu.

    b) In the Project Selection wizard, check the box beside "Enable advanced options in this wizard". Then click **New** to create a new CAF project.
    Important: After clicking on any button in the wizards, wait for the progress bar in the lower right corner of Designer to fully complete (disappear) before continuing.

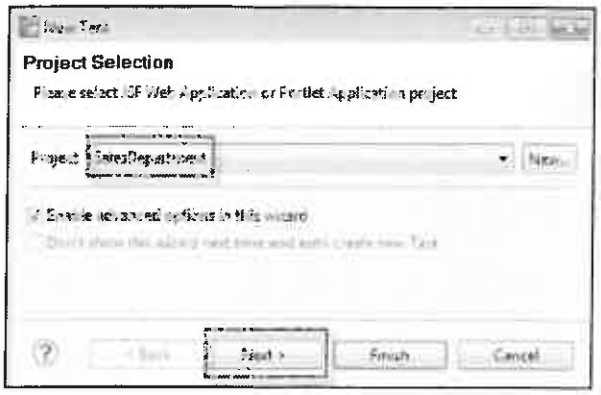


    c) In the New Portlet Application Project wizard, specify the CAF Portlet Application project name as **SalesDepartment**. Keep all the other default values and click **Finish**:

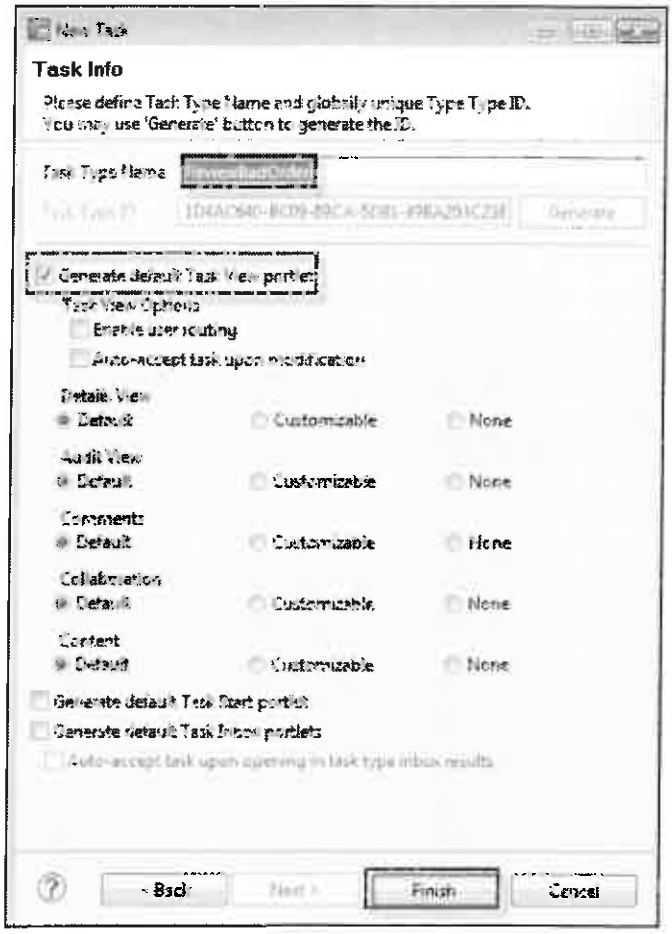


    d) If asked, accept to switch to the **UI Development** perspective.

e) When you are returned to the New Task wizard, confirm that **SalesDepartment** is the project name and click **Next:**

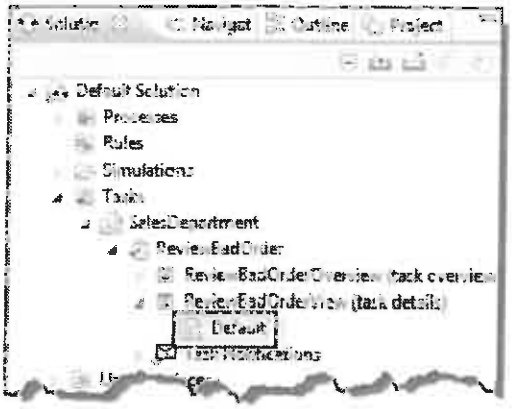


f) In the the subsequest dialog remove the spaces between the Task Type Name, so it reads: **ReviewBadOrder**. Check **Generate default Task View portlet.** Click **Finish** to complete the New Task wizard.



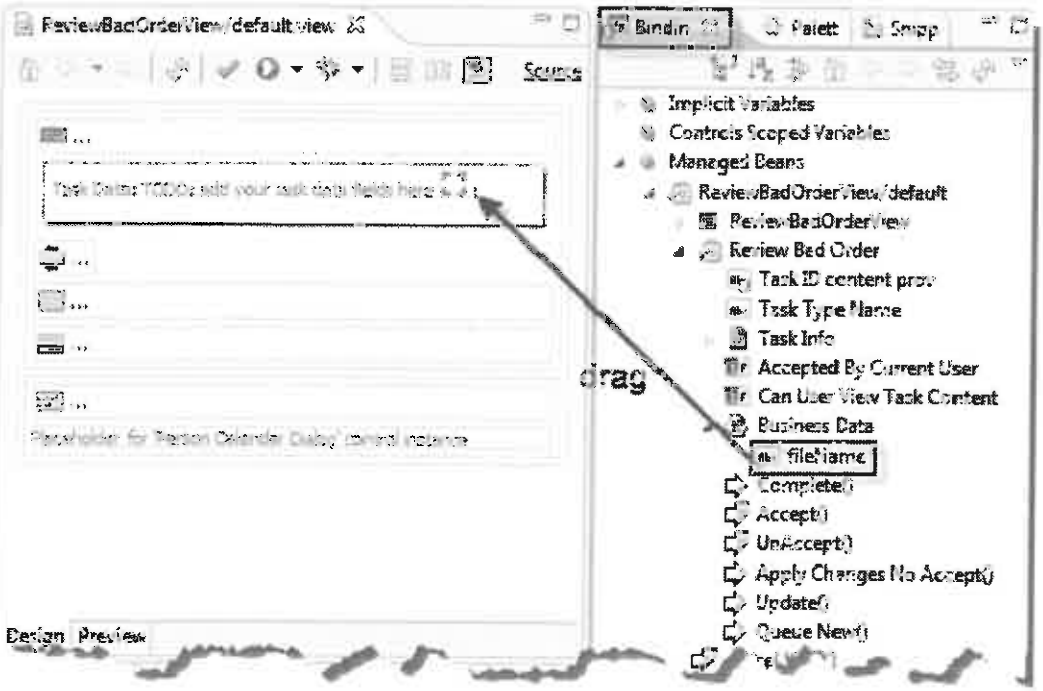
12. Customize the generated default Task UI:

a) Find the generated **Default** view in the Solutions view under **Tasks/SalesDepartment/ReviewBadOrder/ReviewBadOrderView.**



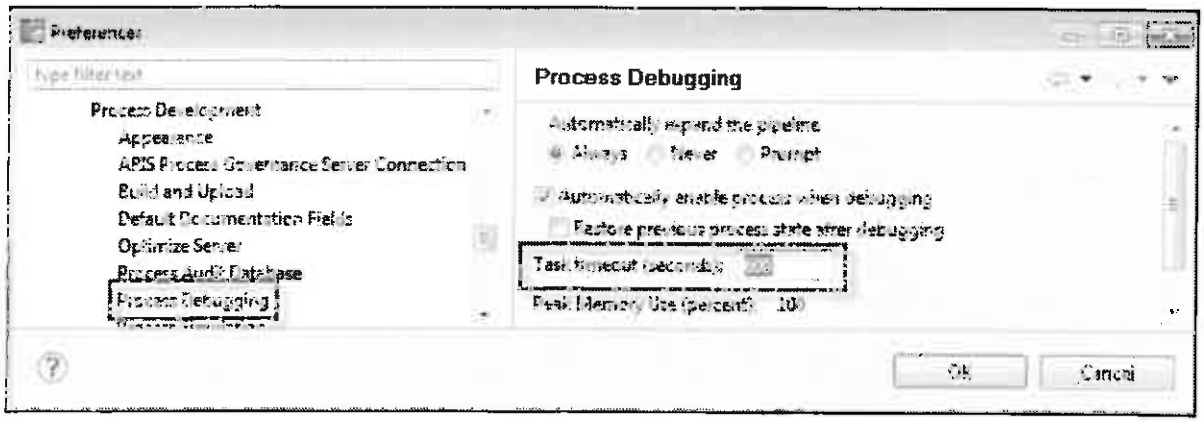
b) Double-click it to open.

c) From the Bindings view, drag the **fileName** variable (found under: **Managed Beans/ReviewBadOrderView/default/Review Bad Order/Business Data**) and drop it inside the rectangle that says "Task Data: TODO: add your task data fields here" of the **ReviewBadOrderView/default.view** (do NOT drop it INSIDE the square box to the right of the word "here")
If you don't see **fileName** in the Bindings view, click the refresh button at the top of the bindings view first.

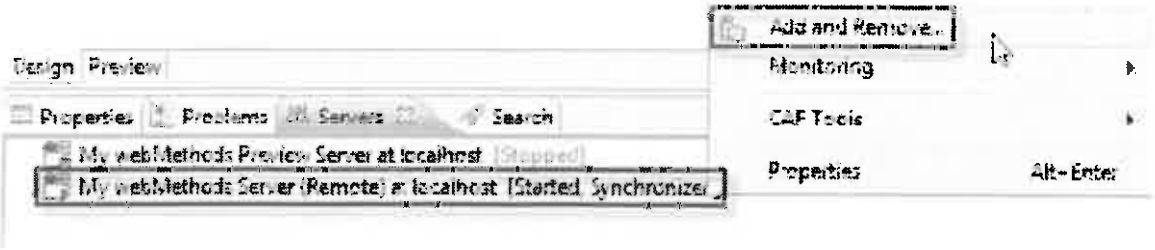


13. Save your changes and switch back to **Process Development** perspective.

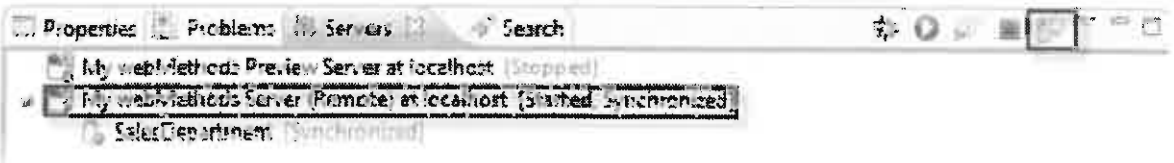
14. To avoid timeouts during debugging, navigate to **Window -> Preferences... -> Software AG -> Process Development -> Process Debugging** and adjust the Task Timeout to **600** seconds.



15. Open the Servers view. Make sure that the status for **My webMethods Server (Remote) at localhost**" is **[Started]**. If it is not started, click on this server and click the Start icon . This will establish connectivity to the My webMethods Server on port 8585.
Right-click the server and add the **SalesDepartment** project to the My webMethods Server. If prompted for Authentication, use **SysAdmin/manage**.

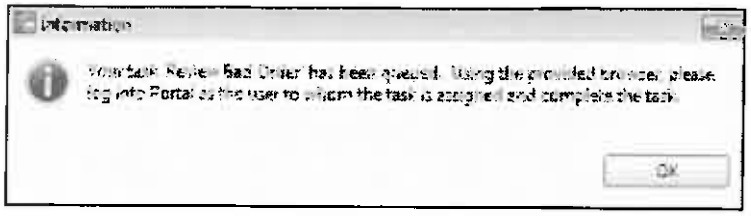


Click on the **Publish to the server** icon to upload and synchronize the **SalesDepartment** project.



16. If not opened yet, open your **HandleNewOrder** process in the process editor.
Build and upload your process model.
Note: Ignore a warning during the build and upload that says it did not deploy task....

17. Click the **Debug Selected Process** icon to start a new process debugging session. For the IS input document, load the provided file ***<workshop_dir>*\Exercise7\Resources\Ex7_input.txt**. Step through the process. The provided order is invalid; the process should execute the route **Receive Order Doc -> Validate Order -> Is Valid? -> Write Bad Order -> Review Bad Order**. Step **Write Bad Order** should log the invalid order in the C:\temp folder. Step **Review Bad Order** should create a new User Task instance:



18. A browser page will open inside of Designer so that you can login to My webMethods. Login using **Administrator/manage**. The Task List Management page will open automatically. Locate your User Task instance and click on the corresponding Task ID value to open it:

19. **Accept** and **Complete** the User Task through the opened Task Details page. Ensure you can see the **fileName** variable in the task data view.

20. In Designer, go back to the Trace view and click the "step over" button to complete the process debugging session normally.

Check Your Understanding

1. Why did you declare fileName as an input to step Review Bad Order?

   To receive the output from Write Bad Order

2. Why did you change the Process Debugger preferences, Task Timeout to 600 seconds?

   To avoid time-outs during debugging

3. Does the User Task Review Bad Order run in MWS, or in an Integration Server?

   MWS

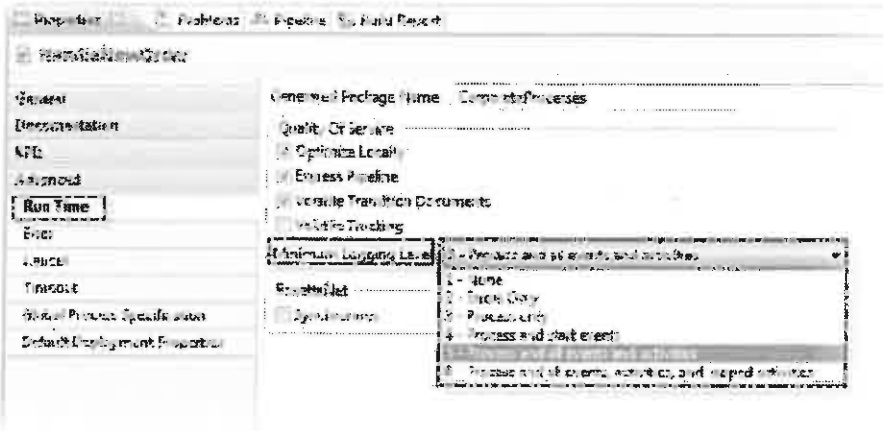
Exercise 8:
Monitor and Resubmit Processes

Overview

In this exercise, you will configure process logging for the **HandleNewOrder** process. Additionally, you will enable process steps for resubmission.

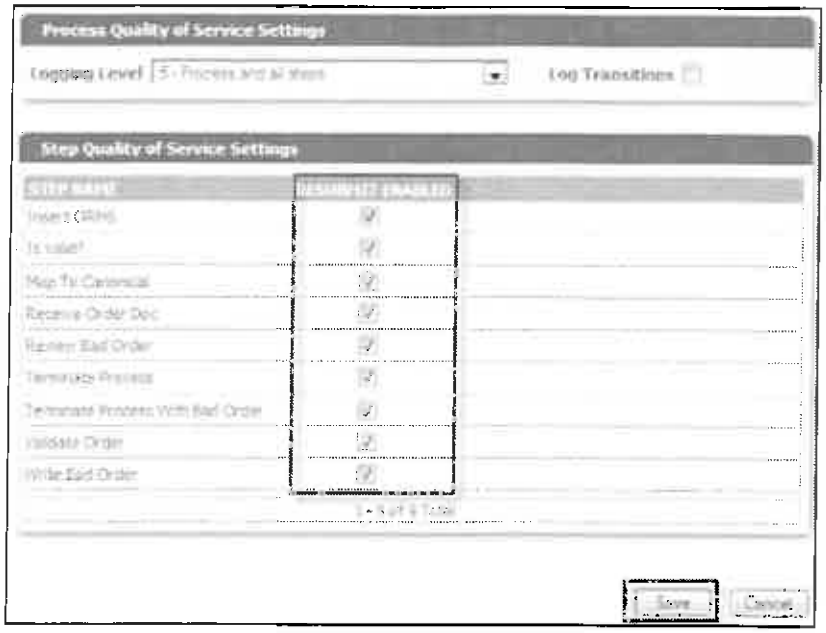
Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Software AG Designer and ensure you are in the **Process Development** perspective.
3. Open the **HandleNewOrder** process and visit its Properties view. Using the Run Time tab, ensure the **Minimum Logging Level** is set to **5**.



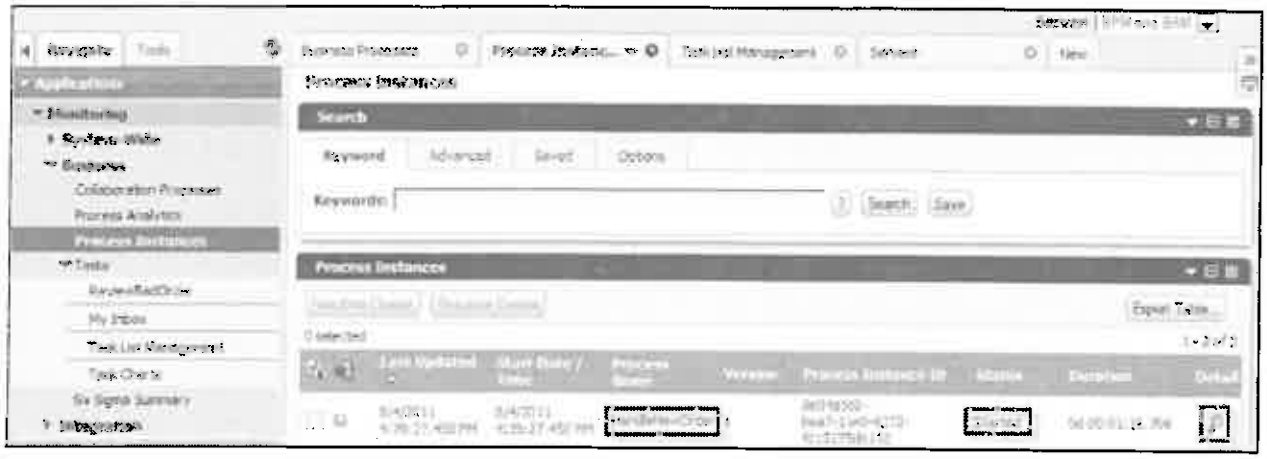
4. Click to Build and Upload the HandleNewOrder process.
5. Use a browser to login to My webMethods as **Administrator/manage**.
6. Drill down to the **Applications** -> **Administration** -> **Business** -> **Business Processes** page, click to edit the configuration details for the **HandleNewOrder** process.

7. Enable all of the process steps to be resubmit-enabled and save your changes.

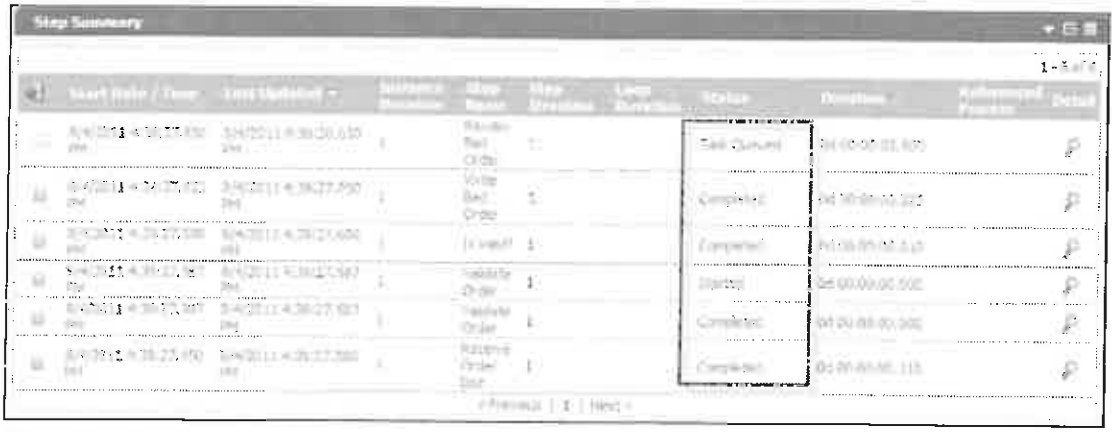


8. Start the **HandleNewOrder** process using a browser by double-clicking the web page: ***<workshop_dir>*\Exercise8\Resources\Ex8_submit.html**
Review the order in the text area. Note that the order has two items, the first item with quantity equal to -9 and the second item with a quantity equal to 120. Click the submit button. If prompted, login to IS as **Administrator/manage.**

9. Using My webMethods, drill down to **Applications -> Monitoring -> Business -> Process Instances**. Look for a new process instance of **HandleNewOrder** with a **Started** status.

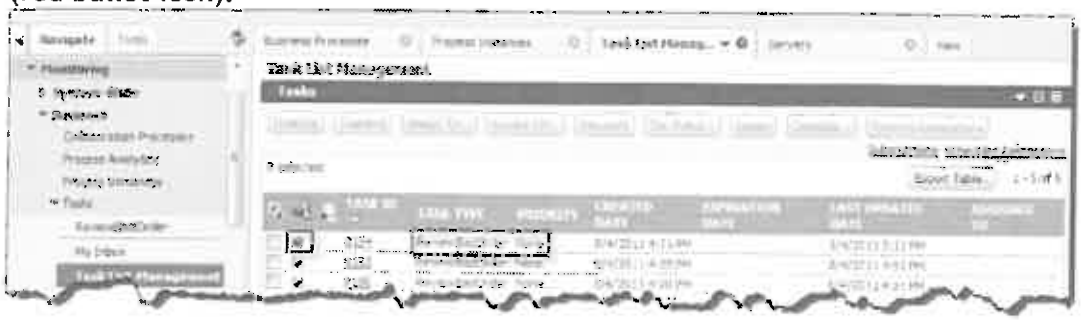


10. Click to view the details of the process instance. In the instance details, under Step Summary, ensure that steps **Receive Order Doc**, **Validate Order**, **Is Valid?**, and **Write Bad Order** have a **Completed** status. However, **Review Bad Order** has a **Task Queued** status:



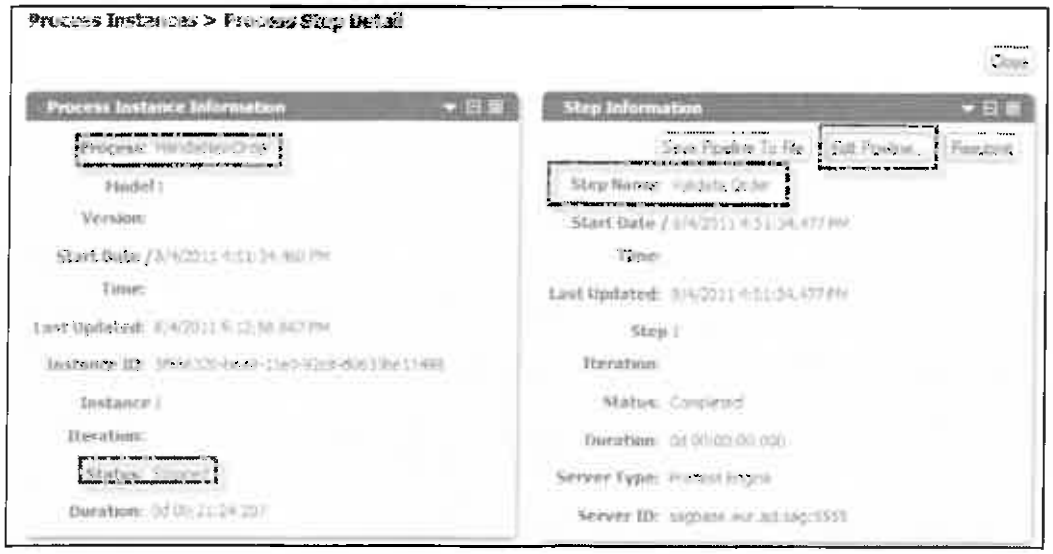
11. **Stop** the HandleNewOrder process from its process instance page in My WebMethods:

12. In My webMethods, navigate to **Applications -> Monitoring -> Business -> Tasks -> Task List Management**. Note an User Task instance of type **ReviewBadOrder** with a **Canceled** status (red bullet icon).

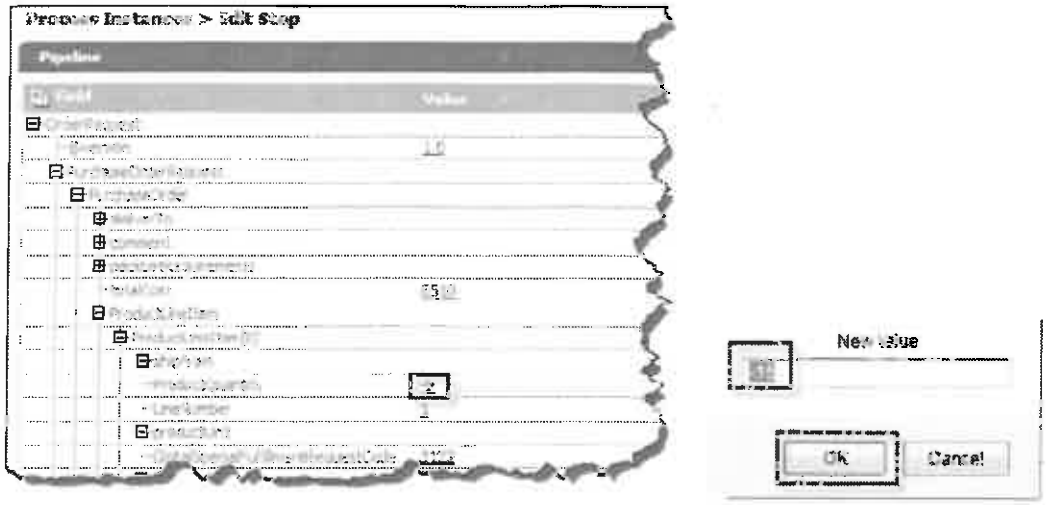


13. Switch back to **Applications -> Monitoring -> Business -> Process Instances**. Review the status of your process instance. Status should be **Stopped**.

14. Click to view the details of the process instance. Drill down to view the details of the step **Validate Order**.



15. Edit the pipeline for the step **Validate Order** so that the quantity of the first ProductLineItem is **-15** instead of -9:



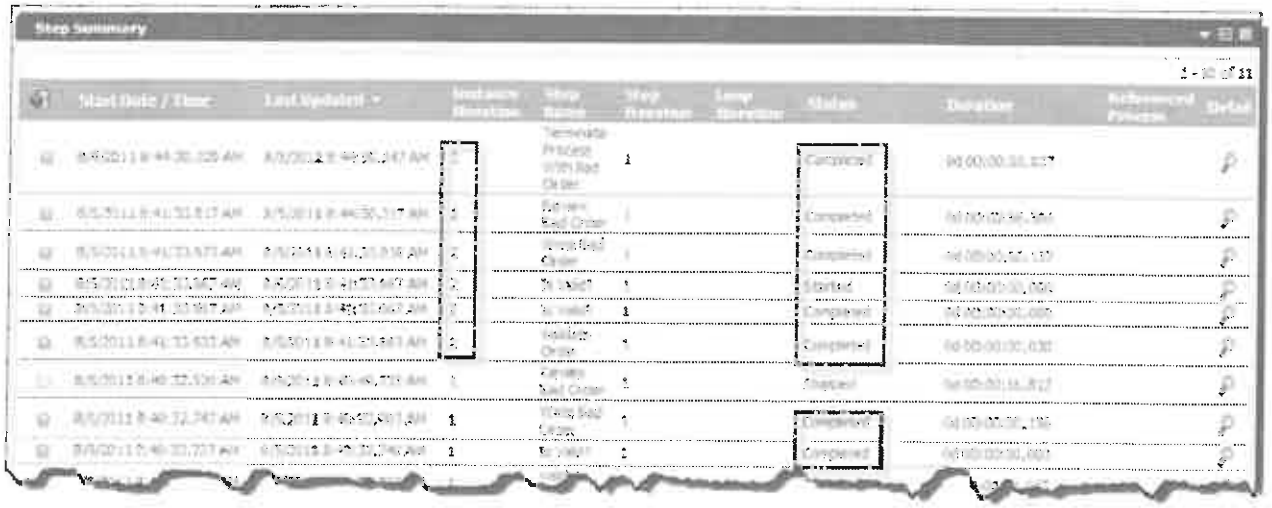
Save your change.

16. Resubmit the **Validate Order** step:

17. Back in the Process Instance Details, in the Step Summary table, ensure that the steps **Validate Order**, **Is Valid?**, **Write Bad Order**, and **Review Bad Order** have been resubmitted.

18. Re-visit the Task List Management page and ensure a new User Task instance has been created. Open the User Task instance. Select **Accept** and **Complete** to complete the User Task.

19. Refresh the Process Instance Detail page, and ensure that all steps are marked **Completed:**



Check Your Understanding

1. The logging level was set to 5 in Designer. Could you have decreased the logging level in My webMethods? Yes.

2. Can a process step be enabled for resubmission in Designer?

   No

3. In step 12 you located a canceled User Task. What forced this User Task to be marked as canceled? It was stopped in step 11.

This page intentionally left blank.

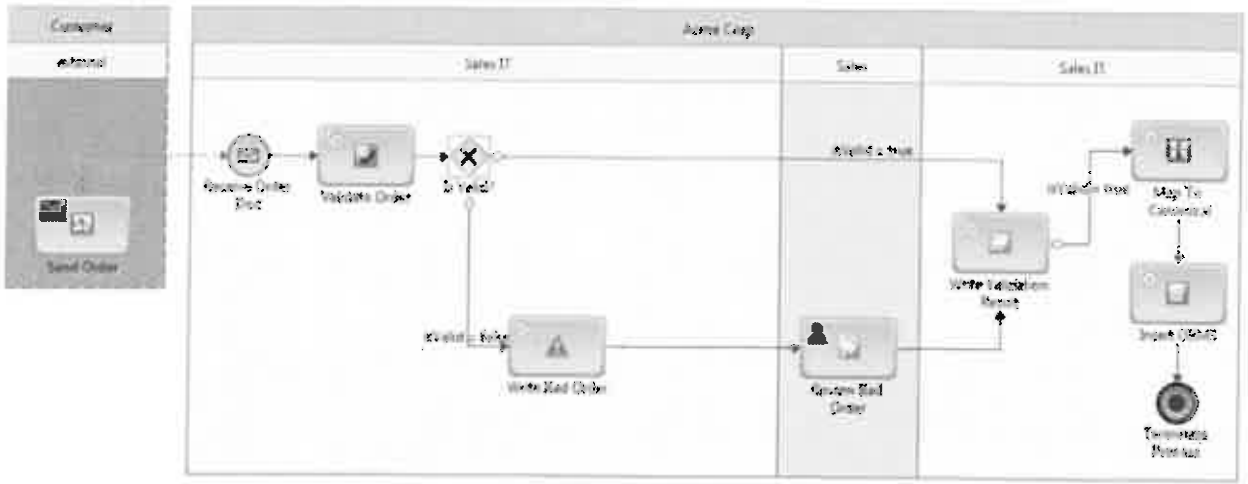
Exercise 9:
Adding a Subprocess

Overview

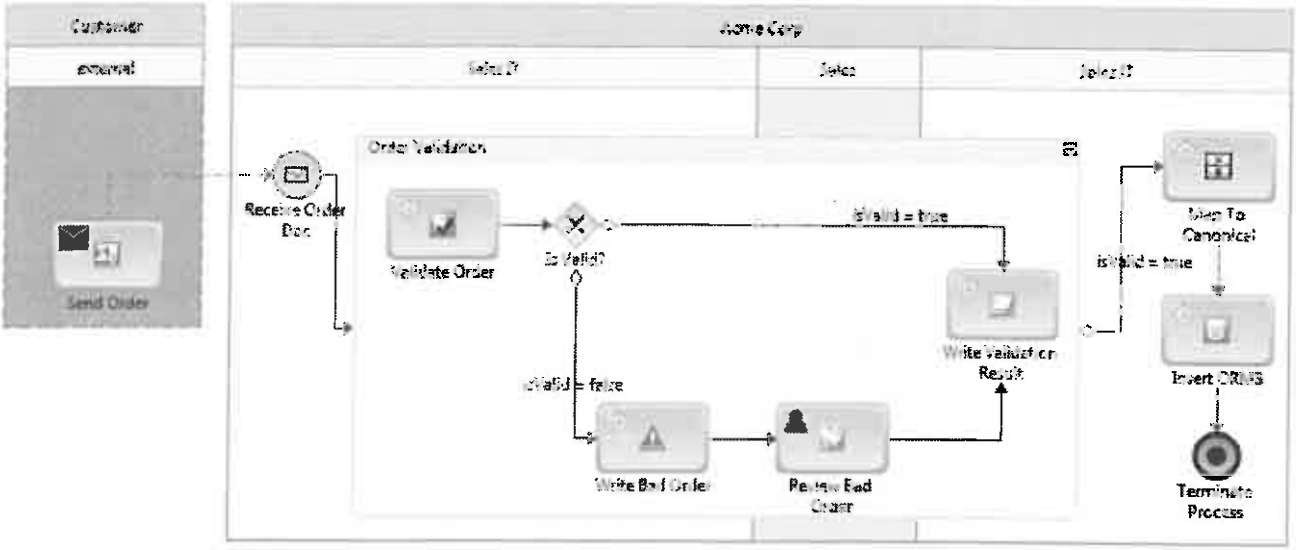
In this exercise, you will add a subprocess to the **HandleNewOrder** Process to encapsulate several process steps. The subprocess can be expanded to show the hidden steps for technical users or it can be collapsed to hide the technical complexity and to improve the readability of your process model.

Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Software AG Designer and ensure you are in the **Process Development** perspective.
3. Open the **HandleNewOrder** process.
4. Remove the **Terminate Process with Bad Order** event.
5. You will need to resize the Sales IT swimlane on the right side. To do this click on the internal pool Acme Corp, and drag the pool to the right. This will make the swimlane wider.
6. Drag steps **Map To Canonical** and **Insert ORMS** from the Sales IT swimlane on the left to the Sales IT swimlane on the right.
7. Delete the transition from **Is Valid?** To **Map To Canonical.**
8. Add a Service Task Activity named **Write Validation Result** into the Sales IT pool at the right. Set up the **Write Validation Result** step to invoke the IS service **bpmDevSupport.utils:writeDocToFile.** Remember to check the inputs and outputs from the Input/Output tab of the Properties View (click icon, if necessary).
9. Draw a conditional transition from **Is Valid?** to **Write Validation Result** that is only used if the field **isValid** contains **true.** Draw a second transition from **Review Bad Order** to **Write Validation Result.** Draw another conditional transition from **Write Validation Result** to **Map To Canonical** that is only used if the field **isValid** contains **true.** Finally set the Join type of step **Write Validation Result** to **Unsynchronized Or** and add a step image to correspond to the following screen shot:



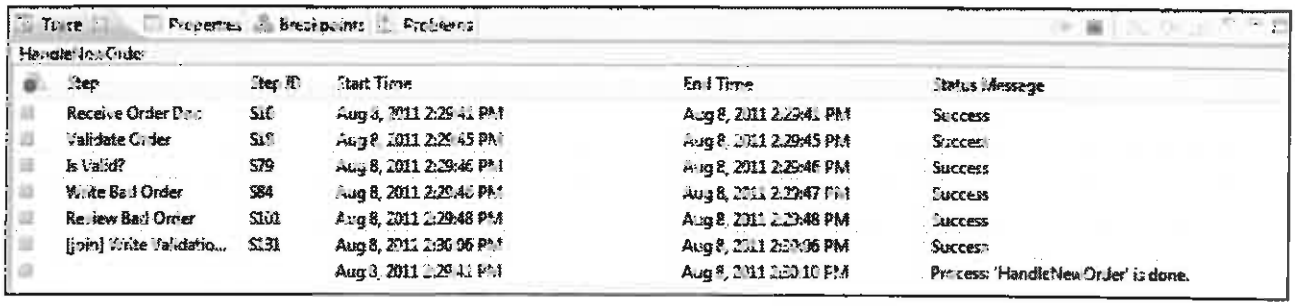
10. Select the steps **Validate Order**, **Is Valid?**, **Write Bad Order**, **Review Bad Order**, and **Write Validation Result** and collapse them to a new Subprocess. Rename the Subprocess to **Order Validation** and click its + Activity Marker to expand it. Your process model should now correspond to the following image:



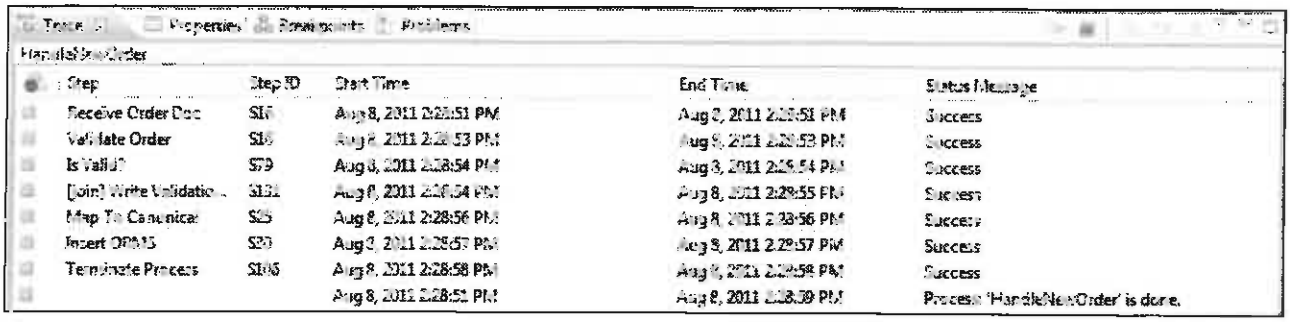
11. Save, build and upload the **HandleNewOrder** process.

12. Click the **Debug Selected Process** icon to start a debugging session. For the IS input document, load an invalid order from the file ***<workshop_dir>*\Exercise9\Resources\Ex9_input1.txt**. Step through the process.

13. Debugging the process will create a new User Task. Use the My webMethods Task List Management page to select, **Accept**, and **Complete** the task. After the User Task is completed make sure you debug the process until it completes. The process should terminate successfully by executing the route **Receive Order Doc -> Validate Order -> Is Valid? -> Write Bad Order -> Review Bad Order -> Write Validation Result:**



14. Start another debugging session. Load a valid order from the file *<workshop_dir>*\Exercise9\Resources\Ex9_input2.txt. Step through the process. The process should terminate successfully by executing the route: **Receive Order Doc -> Validate Order -> IS Valid? -> Write Validation Result -> Map to Canonical -> insert ORMS -> Terminate Process.**



Step	Step ID	Start Time	End Time	Status Message
Receive Order Doc	S17	Aug 8, 2011 2:28:51 PM	Aug 8, 2011 2:28:51 PM	Success
Validate Order	S18	Aug 8, 2011 2:28:53 PM	Aug 8, 2011 2:28:53 PM	Success
Is Valid?	S19	Aug 8, 2011 2:28:54 PM	Aug 8, 2011 2:28:54 PM	Success
[join] Write Validatio...	S132	Aug 8, 2011 2:28:54 PM	Aug 8, 2011 2:28:55 PM	Success
Map To Canonical	S25	Aug 8, 2011 2:28:56 PM	Aug 8, 2011 2:28:56 PM	Success
Insert ORMS	S20	Aug 8, 2011 2:28:57 PM	Aug 8, 2011 2:28:57 PM	Success
Terminate Process	S106	Aug 8, 2011 2:28:58 PM	Aug 8, 2011 2:28:58 PM	Success
		Aug 8, 2011 2:28:51 PM	Aug 8, 2011 2:28:59 PM	Process 'HandleNewOrder' is done.

Check Your Understanding

1. Is the subprocess reusable? yes

2. If the order is invalid, will the process instance leave the subprocess and execute the Map To Canonical step? No

3. In case of an invalid order, when does the process terminate?

   Write Validation Result

This page intentionally left blank.

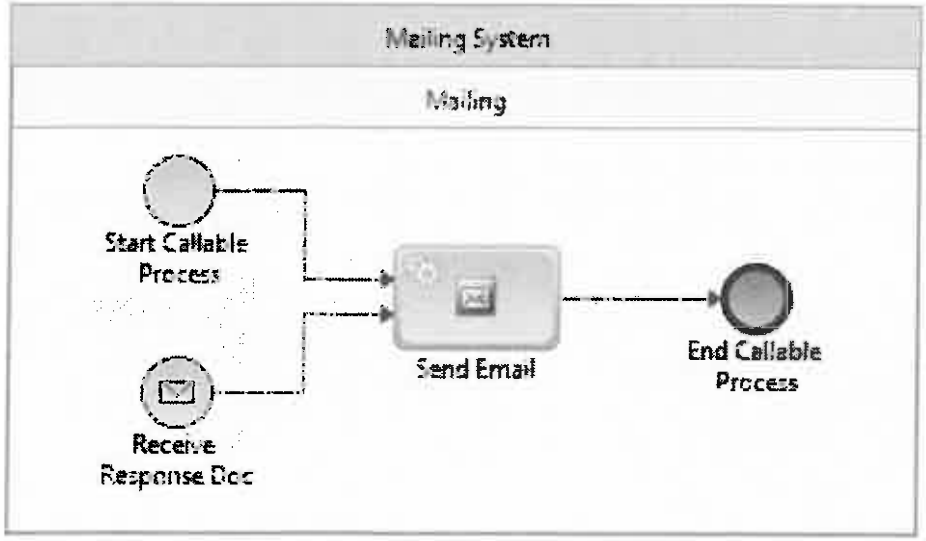
Exercise 10:
Process Invocation via Call Activities

Overview

In this exercise, you will add a new process to your **CorporateProcesses** project. The new process called **NotifyCustomer** is used to notify the customer whether the order is approved or rejected. You will invoke the process model in your **HandleNewOrder** process multiple times using Call Activities. This introduces modularization and reuse.
The parent process invokes **NotifyCustomer** as BPMN Callable Process as well as webMethods Referenced Process. To support this and to allow standalone debugging, you will implement the **NotifyCustomer** process as a "hybrid" process with two Start Events.

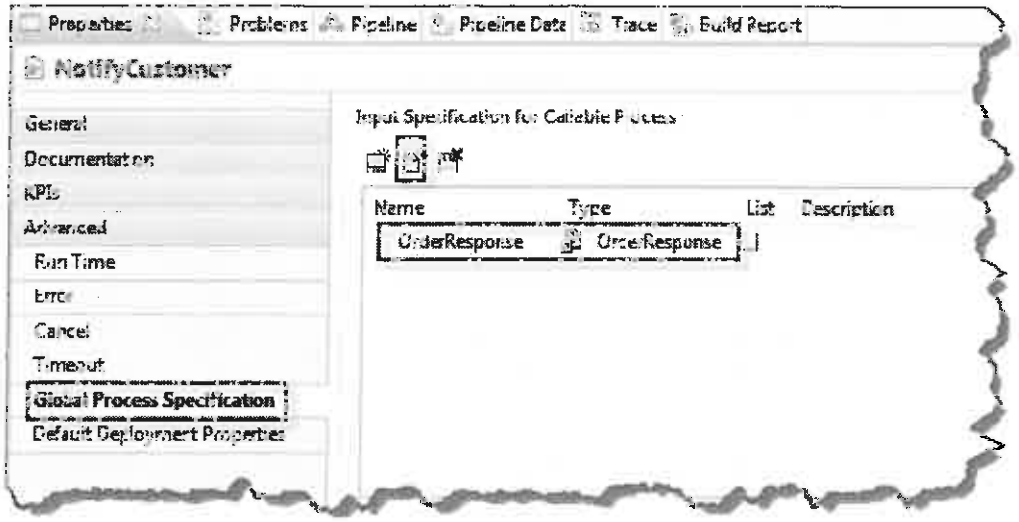
Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Software AG Designer and ensure you are in the **Process Development** perspective.
3. Create a new process in your **CorporateProcesses** project. Name your process **NotifyCustomer.**
   Add an internal pool with default swimlane, a Start None Event, a Start Message Event, a Service Task Activity, and a End None Event. Add transitions to the process, adjust all labels and customize the image of the Service Task Activity to correspond to the following image:

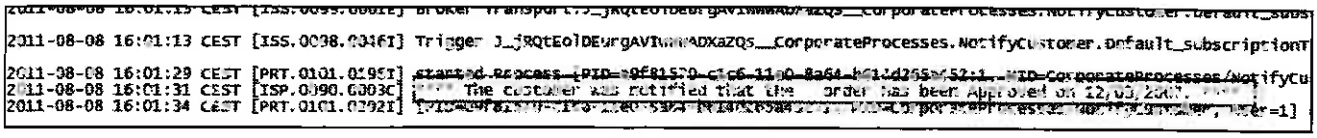


4. Set up the Start Message Event **Receive Response Doc** to receive a document of type **bpmDevSupport.docs.response:OrderResponse**. Remember to refresh the event outputs.
5. Set up the Service Task Activity **Send Email** step to invoke the IS Service **bpmDevSupport.utils:notifyCustomer**. Remember to refresh the step inputs and outputs.
   Note: The service **notifyCustomer** service does not return any output.

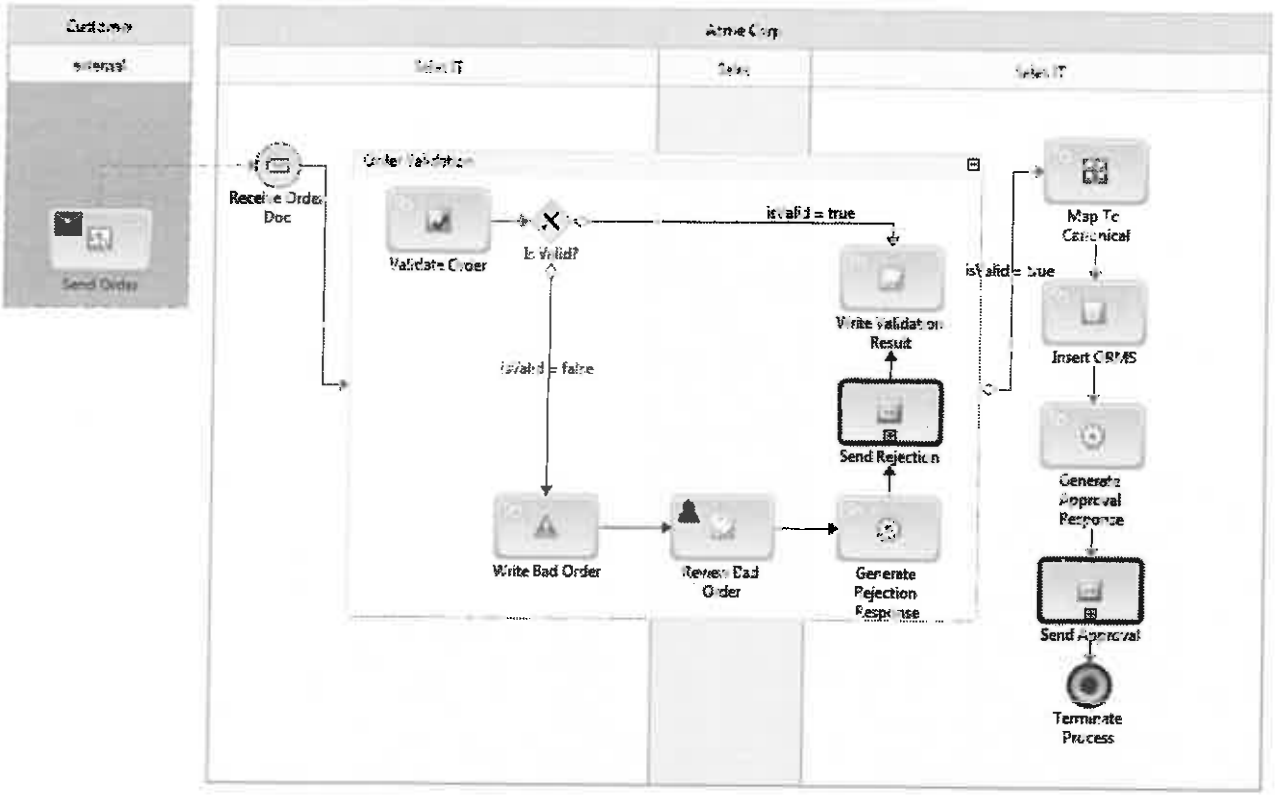
6. In the Global Process Specification of your process model, create a document reference of type **bpmDevSupport.docs.response:OrderResponse** as input specification:



7. Set the Join type of step **Send Email** to **Unsynchronized OR**.
8. Save, build and upload the **NotifyCustomer** process.
9. To test the **NotifyCustomer** process, click the **Debug Selected Process** icon to start a debugging session.
   For the IS input document, load an OrderResponse document from the file ***<workshop_dir>*\Exercise10\Resources\Ex10_input1.txt**. Step through the process. Ensure you can see a message similar to " **** The customer was notified that the order has been Approved..." in the IS Server **server.log** file. To do so, open the IS **server.log** file by using an editor or by using the IS Administration console:



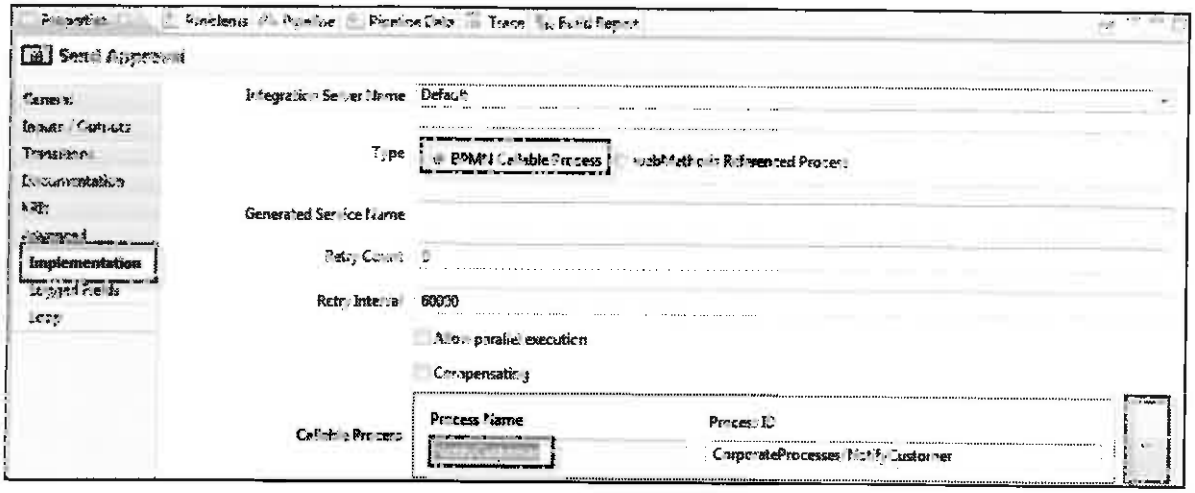
10. Open the **HandleNewOrder** process. Add two Service Task Activities and two Call Activities to the process. Name the Service Task Activities **Generate Rejection Response** and **Generate Approval Response.** Name the Call Activities **Send Rejection** and **Send Approval.** Adjust transitions and step images so that the **HandleNewOrder** process now corresponds to the following image:



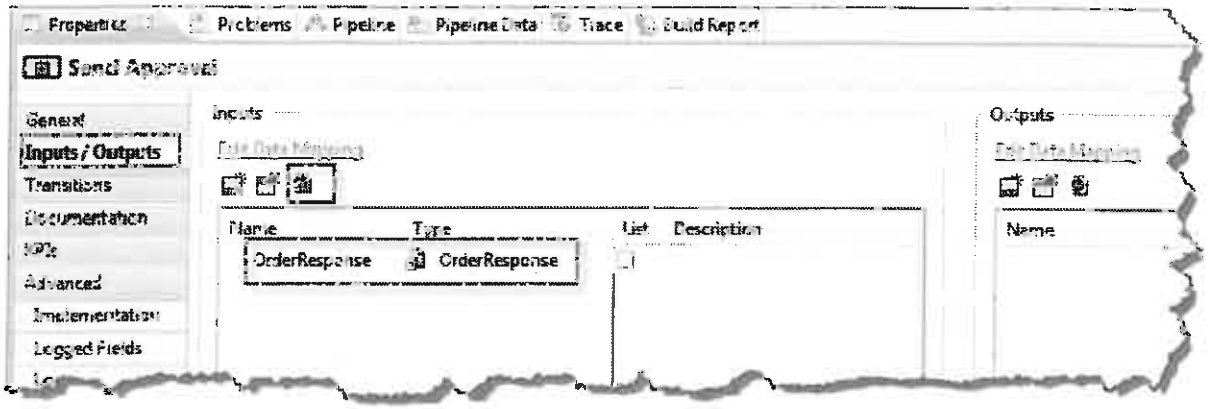
11. Reset the join condition of step **Write Validation Result** step to the Join Type **Unsynchronized Or.**

12. Setup Service Task Activities **Generate Rejection Response** and **Generate Approval Response** to invoke the IS service **bpmDevSupport.utils:generateResponse.** Refresh their inputs and outputs.

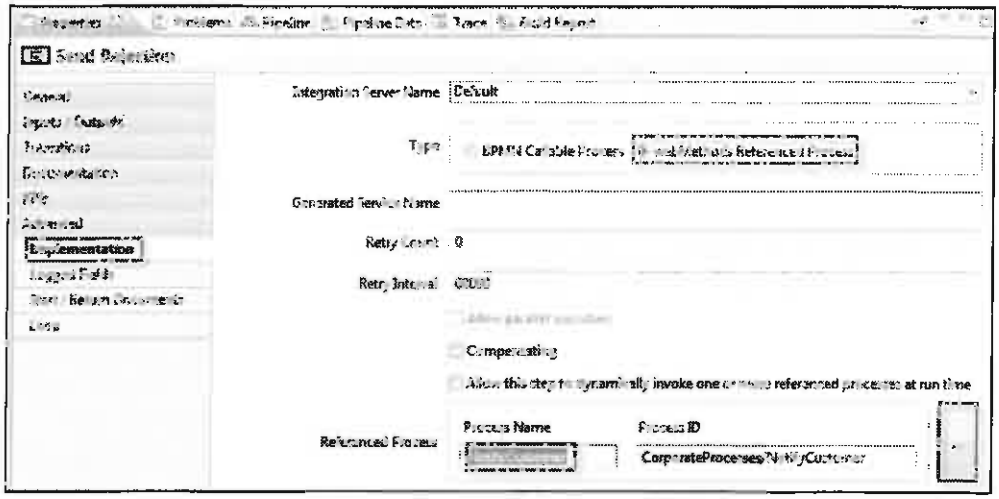
13. Set up Call Activity **Send Approval** to invoke the process **NotifyCustomer** as **BPMN Callable Process:**



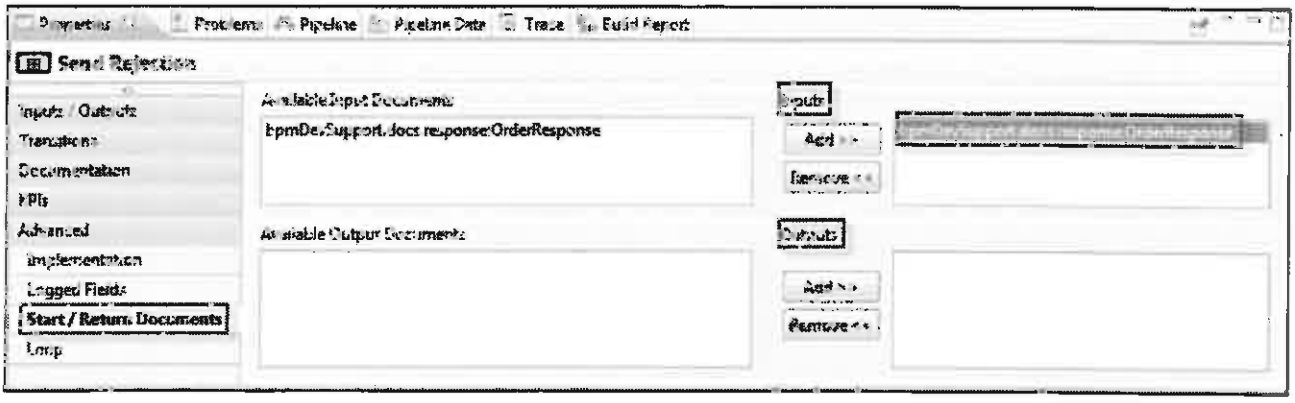
On the Inputs/Outputs tab, refresh the Inputs and Outputs of the **Send Approval** Call Activity. Note that they are retrieved from the Global Process Specification of your invoked **NotifyCustomer** process.



14. Set up Call Activity **Send Rejection** to invoke the process **NotifyCustomer** as **webMethods Referenced Process:**



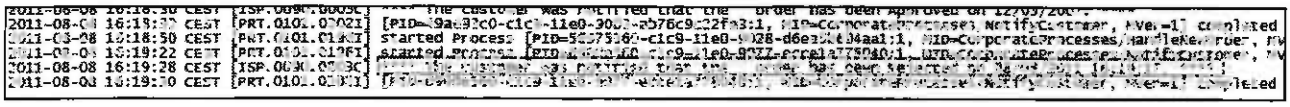
Use the **Start/Return Document** tab to ensure that the Start document corresponds to the document to be received (Inputs) by your referenced **NotifyCustomer** process.
Note: **NotifyCustomer** does not publish any return document, so the return document (Outputs) has to be empty.

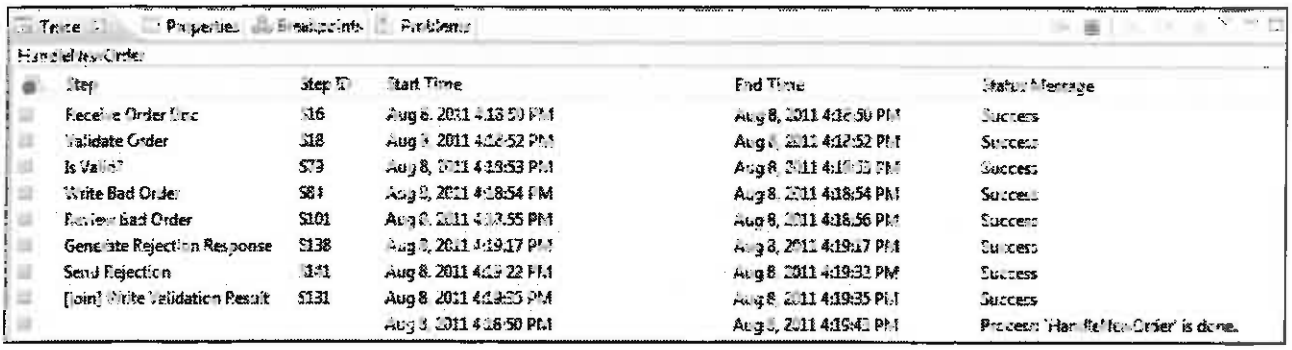


Finally refresh the Inputs and Outputs of the **Send Rejection** Call Activity.

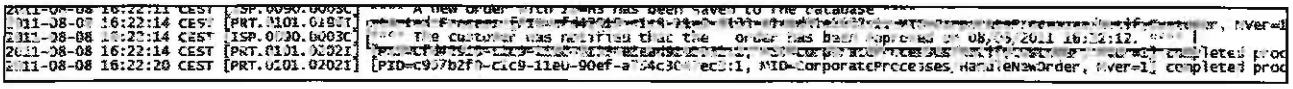
15. Save, build and upload your **HandleNewOrder** process.

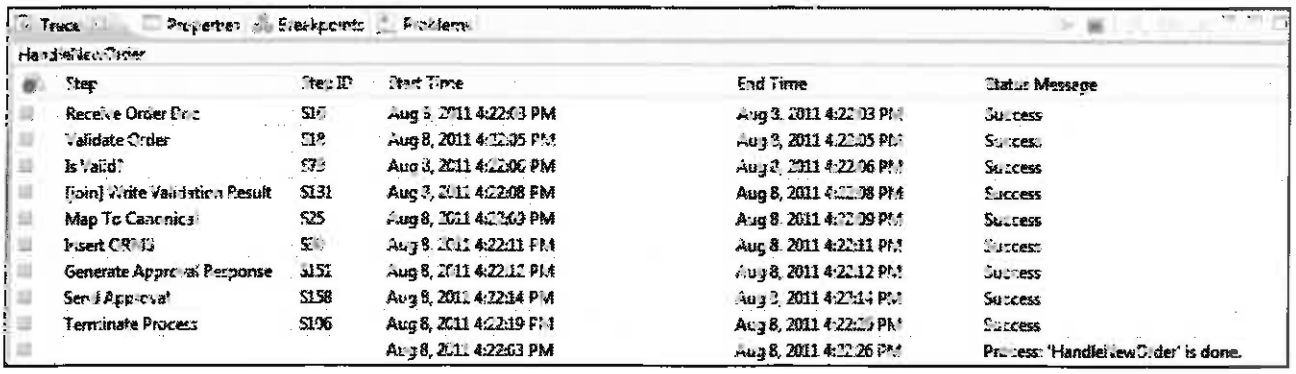
16. Start a debugging session. For the input document, load an invalid order from the file ***<workshop_dir>*\Exercise10\Resources\Ex10_input2.txt**. Step through the process, **Accept/Complete** the User Task instance in the Task List Management page and then finish stepping through the process.
Ensure you can see a message similar to " **** The customer was notified that the order has been Rejected ..." in the IS Server **server.log** file:





17. Start another debugging session. For the input document, load a valid order from the file ***<workshop_dir>*\Exercise10\Resources\Ex10_input3.txt**. Step through the process.
Ensure you can see a message similar to " **** The customer was notified that the order has been Approved ..." in the IS Server **server.log** file:





Check Your Understanding

1. Is the process invoked by your Send Approval Call Activity reusable?

   Yes

2. Process NotifyCustomer is invoked twice in the parent (HandleNewOrder) process. It produces different results based on where it is executed from in the parent. What causes the different results to be generated?

   The value of isValid

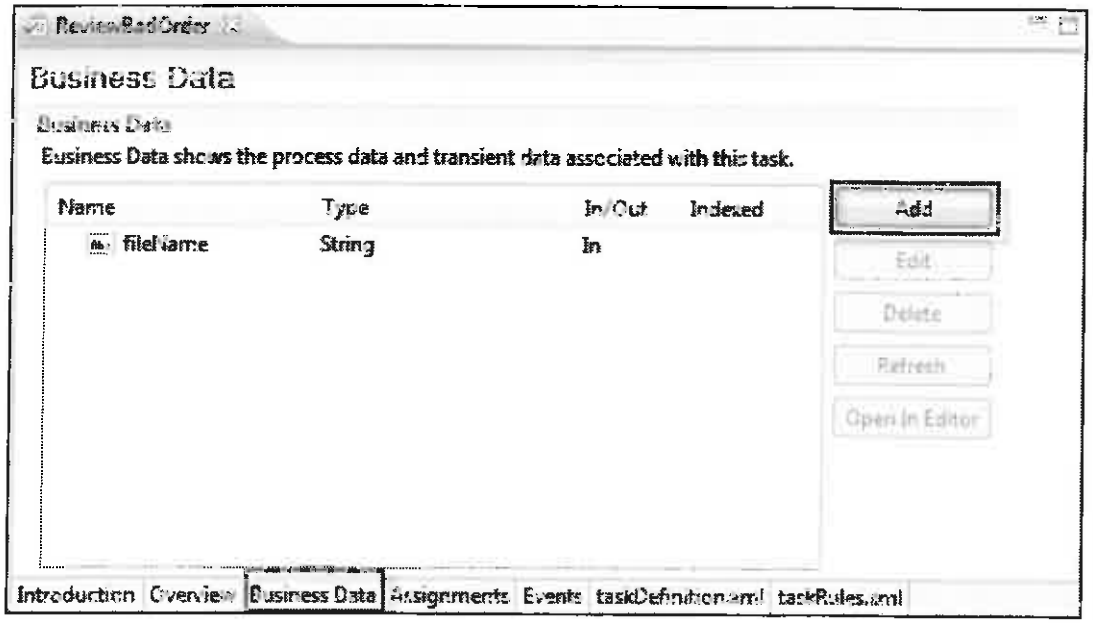
Exercise 11:
Re-factor User Task Inputs and Outputs

Overview

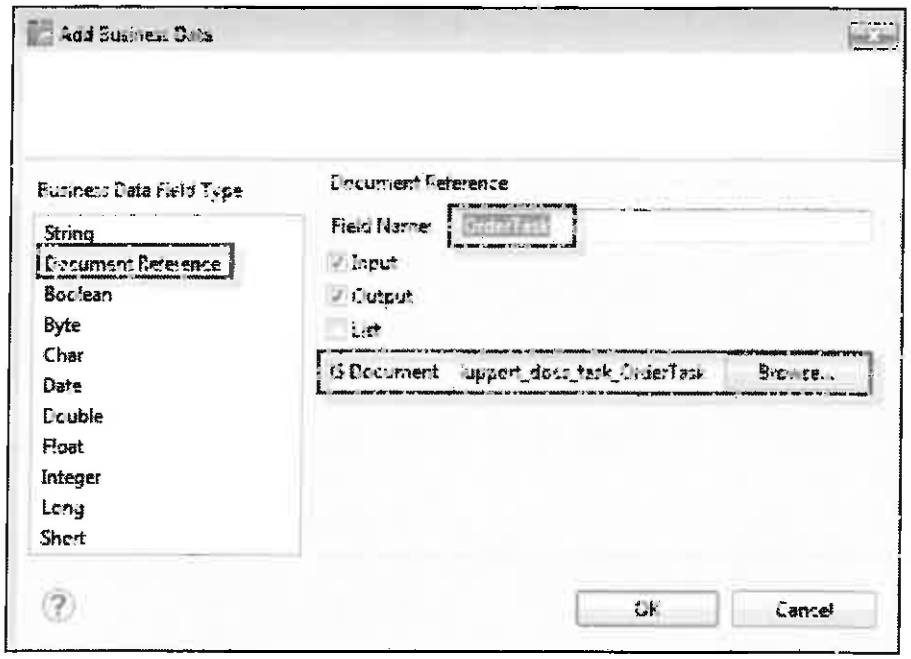
In this exercise, you will modify the **ReviewBadOrder** User Task to display more realistic business data and to return a revalidation flag. You will enhance the **HandleNewOrder** process to react on the revalidation decision made by the user in the User Task UI. You will also have to adjust the inputs and outputs of the corresponding User Task Activity in the HandleNewOrder process so that they match the business data expected in the User Task.

Steps

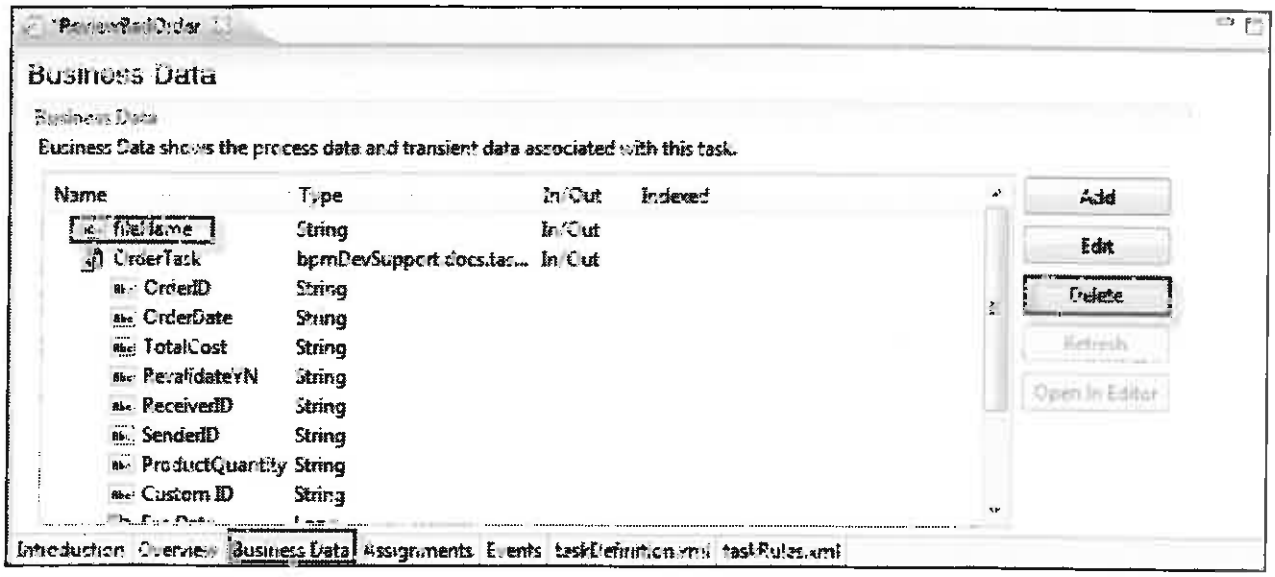
1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Software AG Designer and ensure you are in **UI Development** perspective.
3. Using the Solutions view, open the **ReviewBadOrder** User Task by double-clicking **Tasks/SalesDepartment/ReviewBadOrder**. Select the Business Data tab in the Task editor to modify the ReviewBadOrder User Task's Business Data in the following way:
   a) Add a document reference of type **bpmDevSupport.docs.task:OrderTask** to the Business Data as an In/Out field. To do so, click on **Add** or use drag and drop from the Package Navigator:



Click the Browse button and select **bpmDevSupport.docs.task:OrderTask.** Make sure the value of Field Name changes to **OrderTask:**

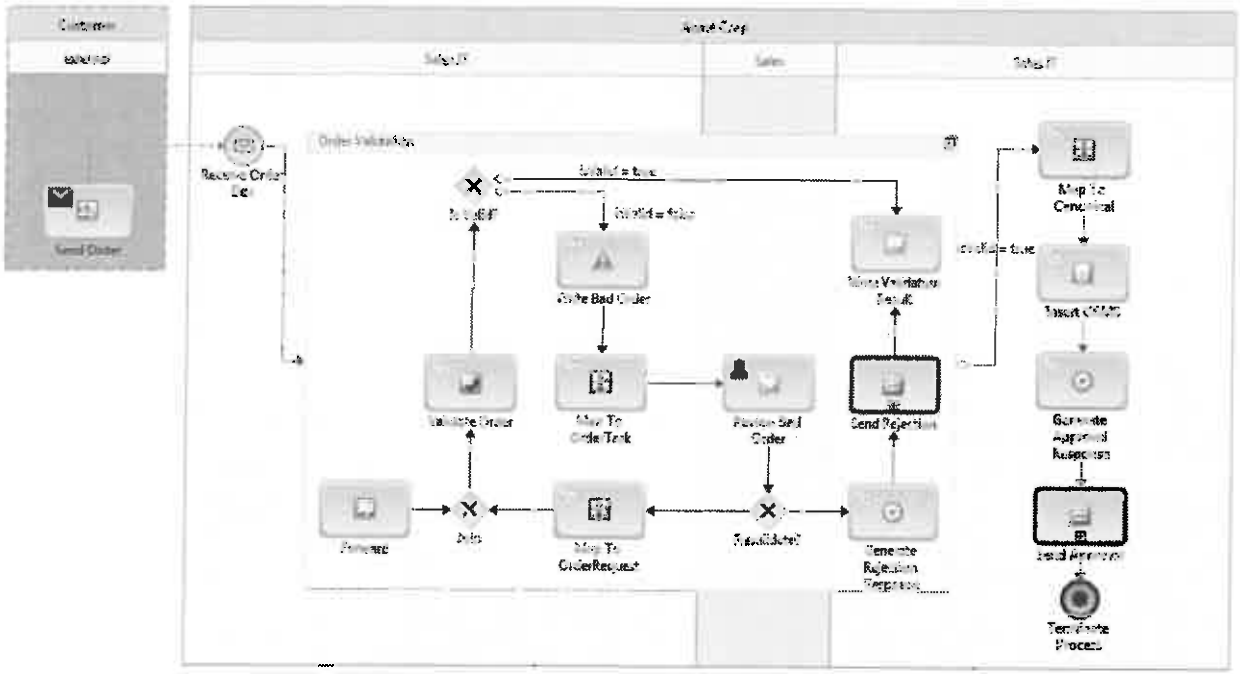


b) Delete the existing **fileName** string variable from the **task** Business Data:

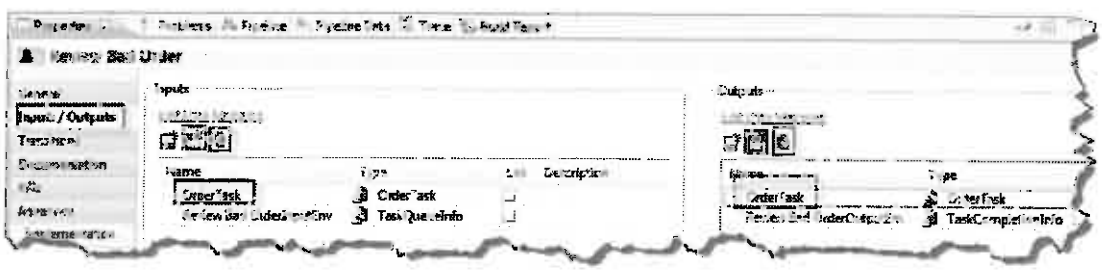


c) Save your changes.

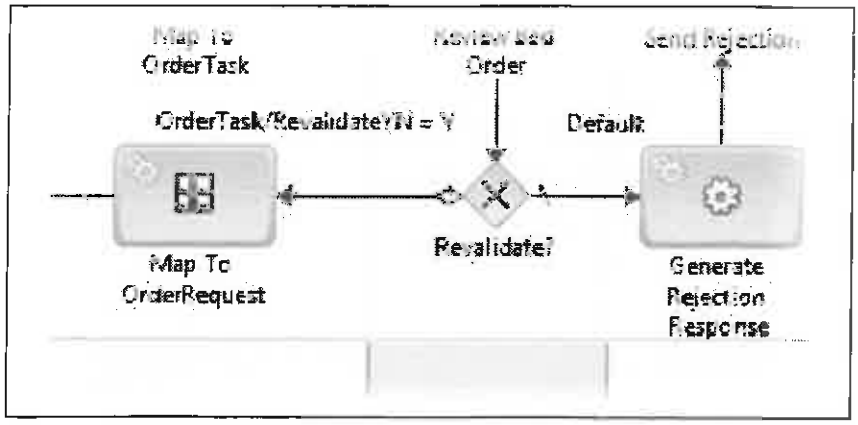
4. Switch to the **Process Development** perspective and open the **HandleNewOrder** process. Add two Service Task Activities to the process. Name the steps **Map To OrderRequest** and **Map To OrderTask.** Also add an Abstract Task Activity labeled as **Forward.**
Now add two Exclusive Gateways: the first one is to split the path and should be labeled **Revalidate** and the second one is to join two paths and should be labeled **Join.**
Finally, provide transitions and step images so that your HandleNewOrder corresponds to the following image:



5. Set up Service Task Activity **Map To OrderRequest** to invoke the IS service **bpmDevSupport.maps:OrderTaskToOrderRequest.** Refresh the input and output data for the step.
6. Set up Service Task Activity **Map To OrderTask** to invoke the IS service **bpmDevSupport.maps:OrderRequestToOrderTask**. Refresh the input and output data for the step.
7. The Abstract Task Activity **Forward** acts as a unique entry point of the entire subprocess. It forwards the incoming order to the Gateway step labeled as **Join.** This Gateway joins between newly received orders and orders to be revalidated. Ensure that the Join type of the **Join** Gateway is an **Unsynchronized OR.**
8. Refresh the inputs and the outputs of the **Review Bad Order** User Task Activity in the Properties view. Ensure the step has **OrderTask** as an input and as an output. Remove field **fileName.**

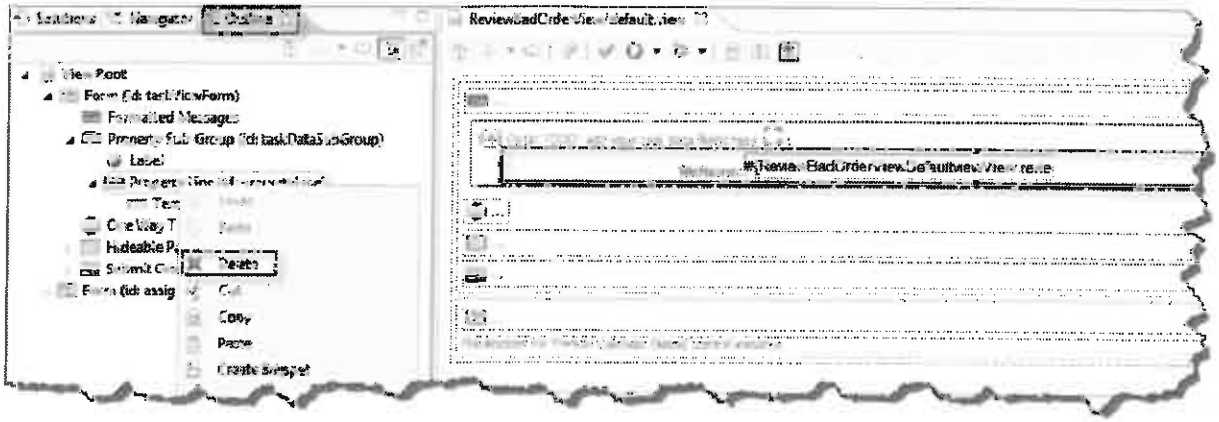


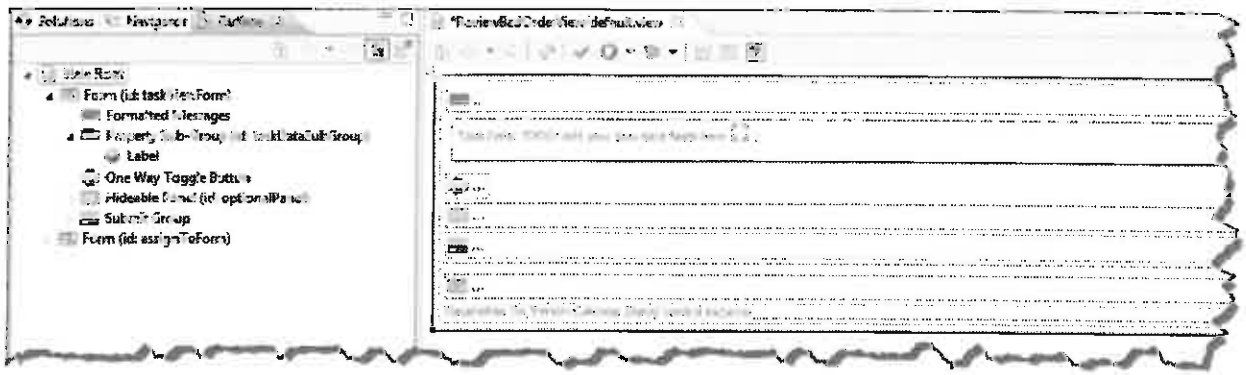
9. Add the condition **"OrderTask/RevalidateYN = Y"** to the transition from **Revalidate?** to **Map To OrderRequest.**
   Change the transition from **Revalidate?** to **Generate Rejection Response** to become the **Default** transition:



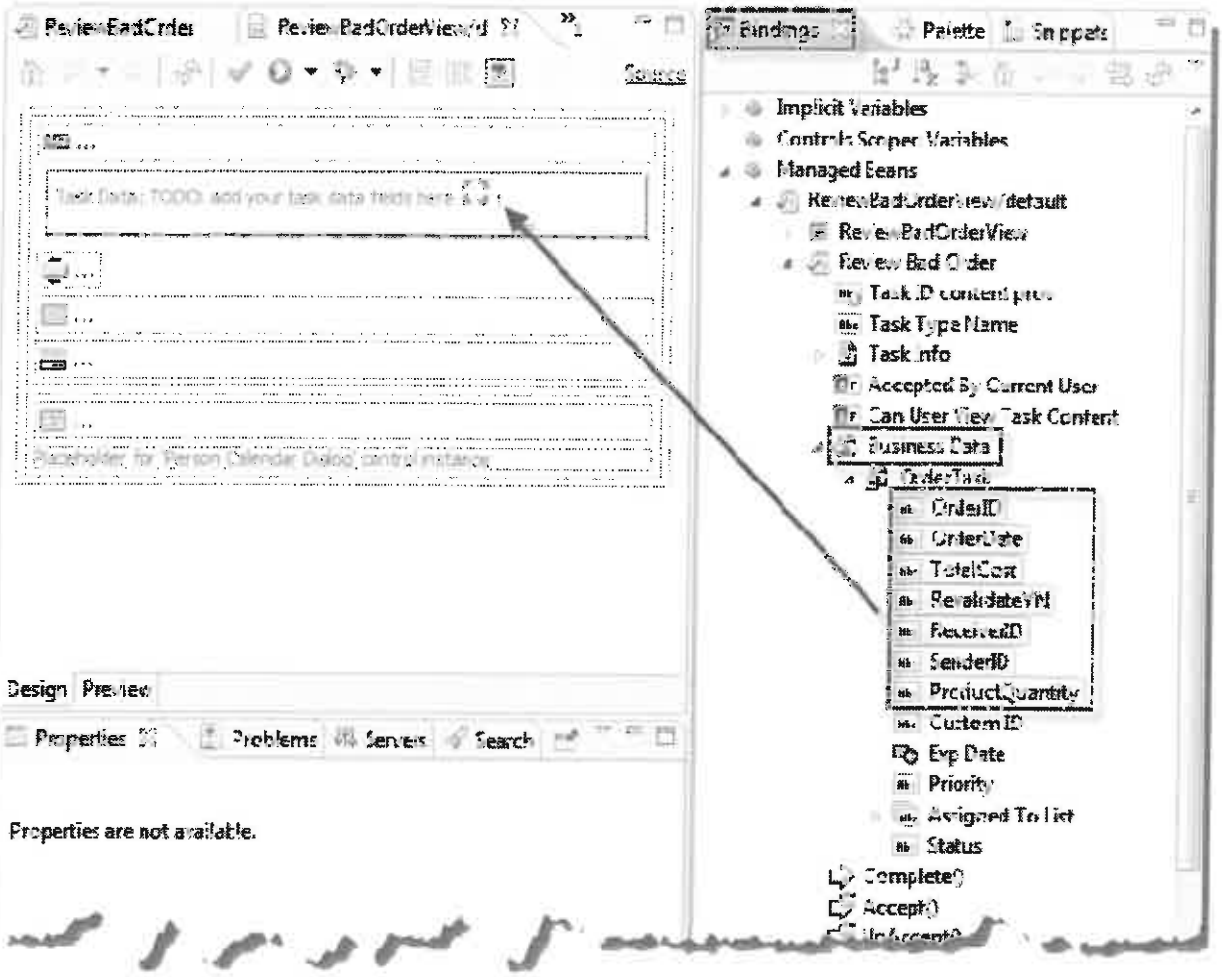
10. Save your changes.

11. Switch back to the **UI Development** perspective.
    Using the Solutions view, drill down to **Tasks** -> **SalesDepartment** -> **ReviewBadOrder** -> **ReviewBadOrderView** -> **Default** to open the **default.view**. Delete the property line which holds the variable **fileName.**
    Note: Deleting can also be performed by using the Outline view.

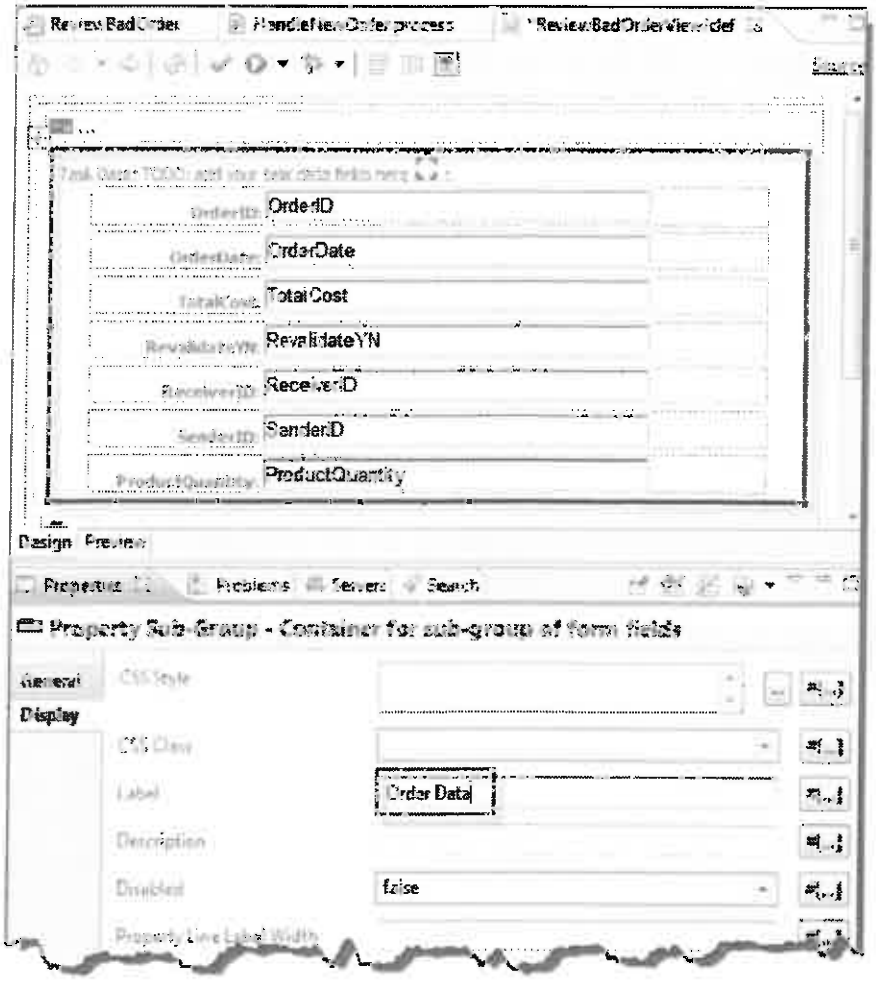




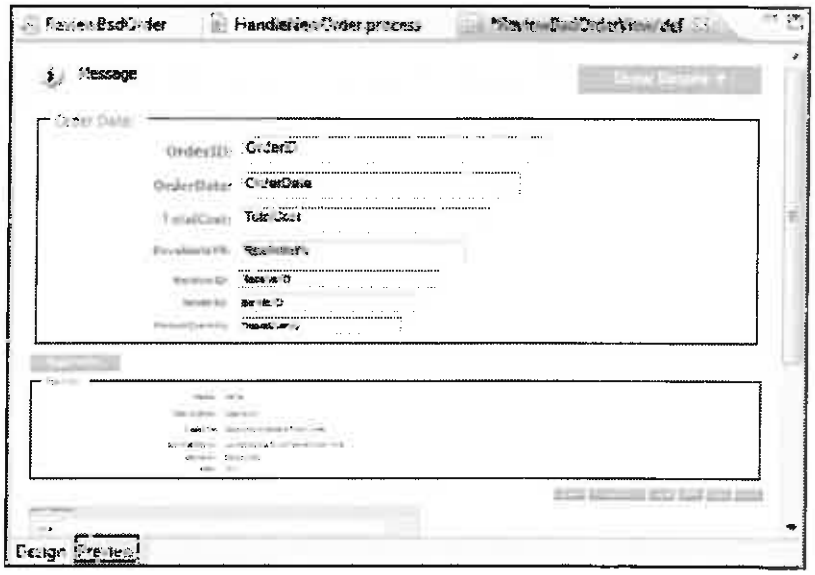
12. In the Bindings view, from within the **Managed Beans/.../Review Bad Order/Business Data/OrderTask** document, drag and drop the fields **OrderID** through **ProductQuantity** to the **TaskData** section in the file **ReviewBadOrderView/default.view.** Use the shift key to select all the fields, then drag and drop them into the view:



13. Highlight the box around the Task Data and select **Display** in the Properties view. Give your task data a more user friendly label by entering **Order Data** in the **Label** field.

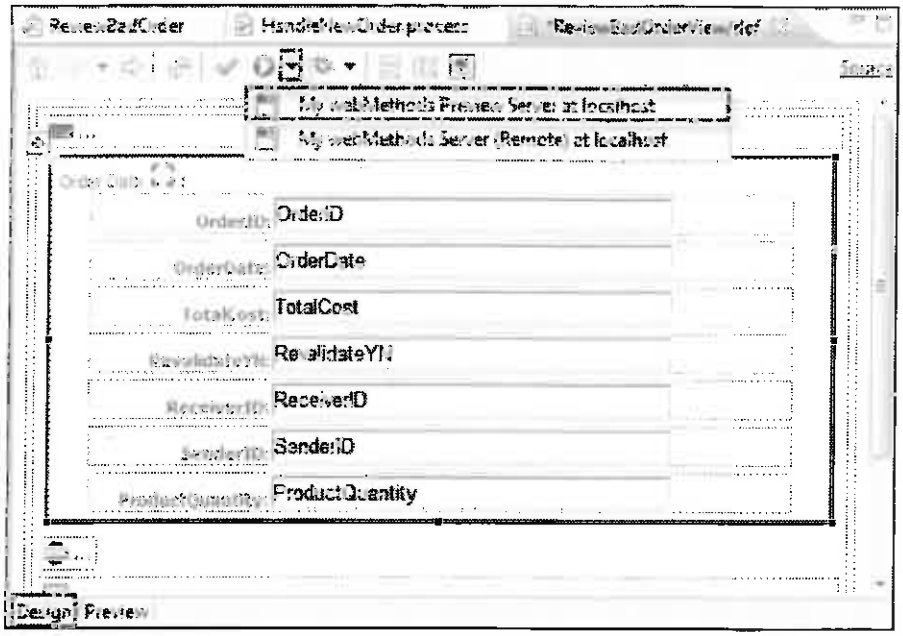


14. Save your changes. Select the Preview tab at the bottom of the task details editor to preview the task data presentation of the **ReviewBadOrderView/default.view**:

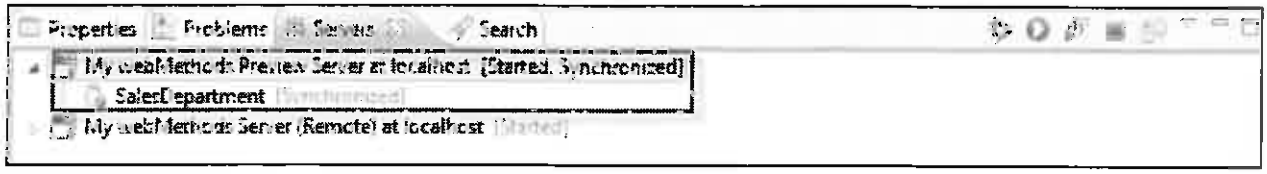


15. Using the Servers view, add and publish the **SalesDepartment** project into **My webMethods Preview Server** (not the remote Server).

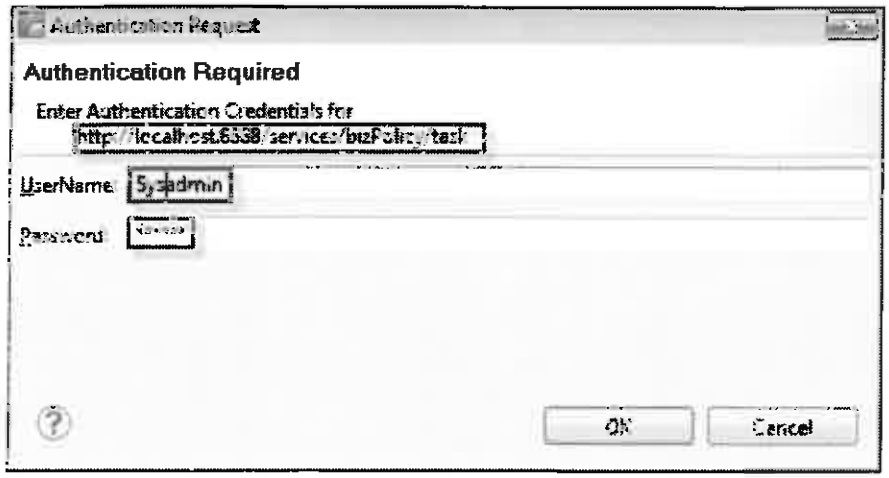
16. Click the Design tab at the bottom of the task details editor. Run the **default.view** of your **ReviewBadOrder** task on the My webMethods Preview Server:



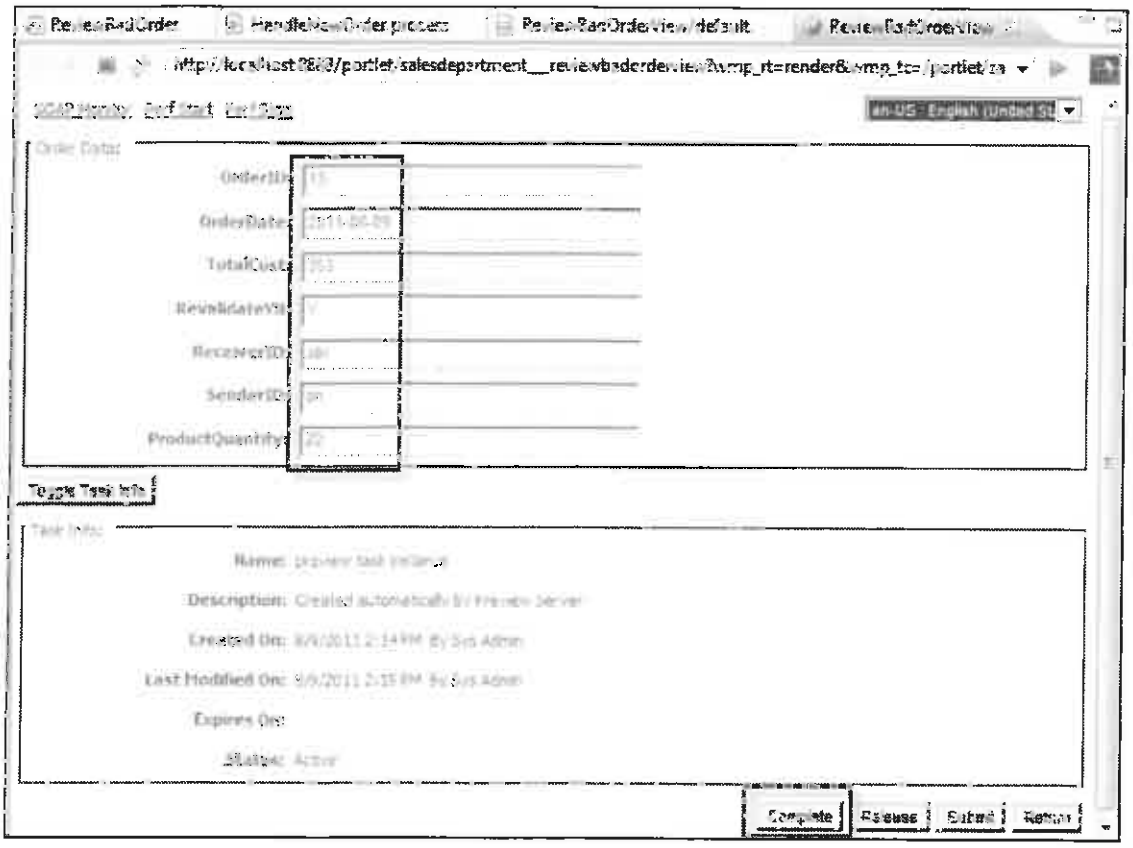
This will bring up the embedded Preview Server listening on port 8888 and publish the SalesDepartment project to this server:



If asked for authentication for the Preview Server, provide **Sysadmin/manage**.
Note: Ensure that it is the Preview Server listening on port 8888.



When ReviewBadOrderView appears in the browser view, first **Accept** the User Task, then enter some test business data, finally **Complete** the User Task:



Finally stop the Preview Server from the Servers view.

17. Using the Servers view, republish the **SalesDepartment** project into My webMethods Server (not the Preview Server):

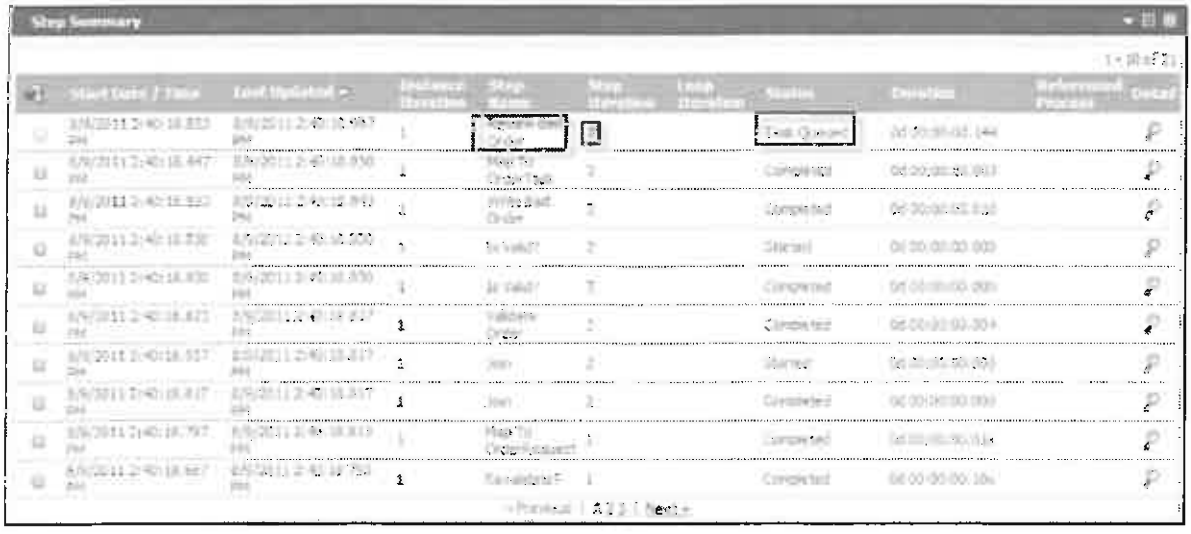
If prompted for Authentication, provide **SysAdmin/manage**. As a result, SalesDepartment should be marked with [Synchronized].

18. Switch back to the **Process Development** perspective.
19. Build and upload the **HandleNewOrder** process.
20. To start the **HandleNewOrder** process using a browser, double-click ***<workshop_dir>*\Exercise11\Resources\Ex11_submit.html**. Review the order in the text area. Note that the order is invalid since the quantity of the first product item is -9. Click the **Submit** button. If asked for IS authentication, provide **Administrator/manage**.
21. Using My webMethods, drill down to **Applications -> Monitoring -> Business -> Process Instances**. Look for a new **HandleNewOrder** process instance with a **Started** status. Click to view the details of the process instance.

22. In the instance details, under Step Summary, ensure that steps **Receive Order Doc**, **Forward**, **Join**, **Validate Order**, **Is Valid?**, **Write Bad Order**, and **Map To OrderTask** have been executed and **Review Bad Order** has a **Task Queued** status.

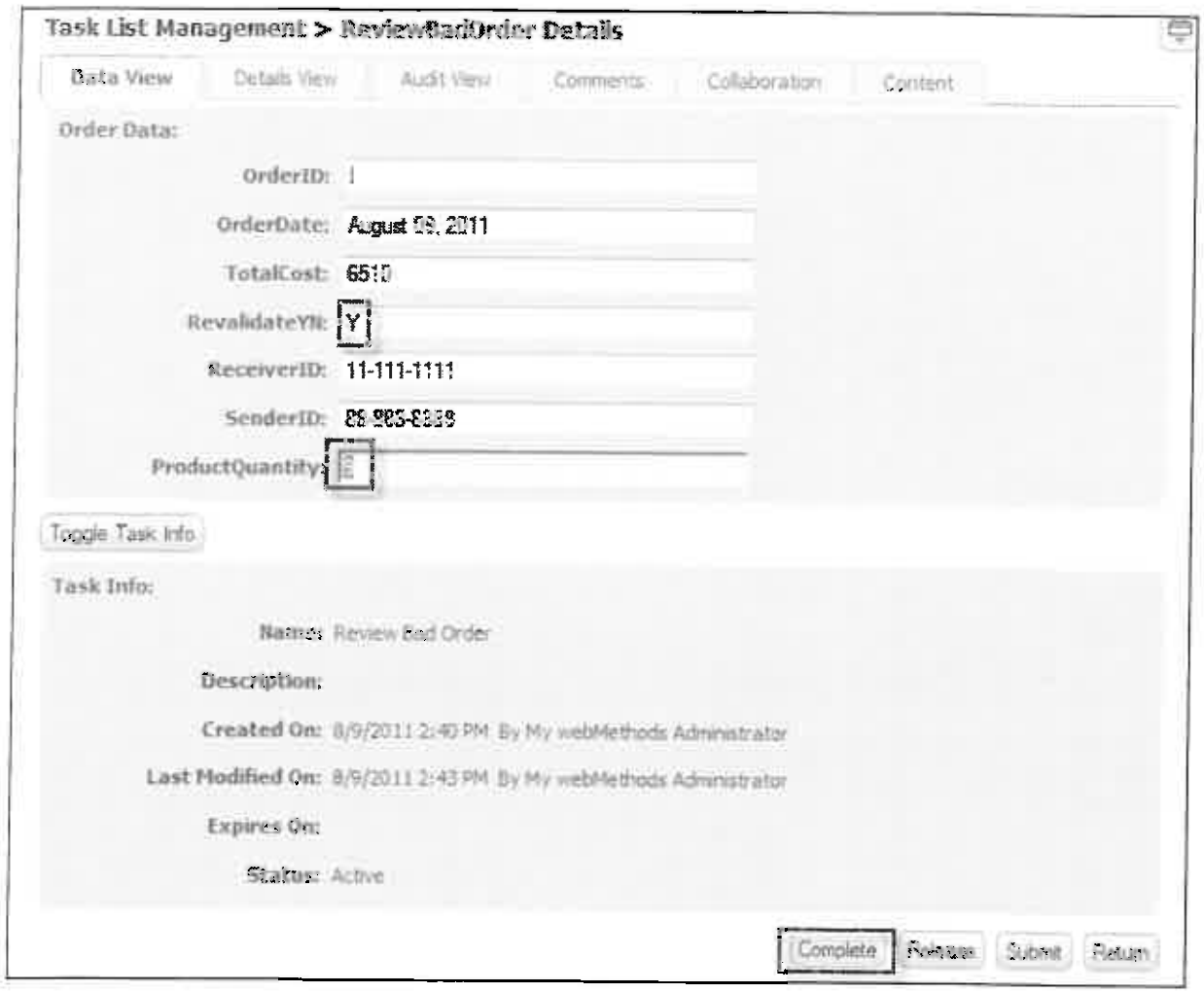
23. Switch to the **Applications** -> **Monitoring** -> **Business** -> **Tasks** -> **Task List Management** page. A new instance of a **Review Bad Order** User Task has been created. Select and **Accept** the User Task. Set field **RevalidateYN** to **Y**, and **Complete** the User Task:

24. Switch back to the **Applications** -> **Monitoring** -> **Business** -> **Process Instances** page and ensure that the order was revalidated. Since the order is still invalid, a new User Task is created and queued:

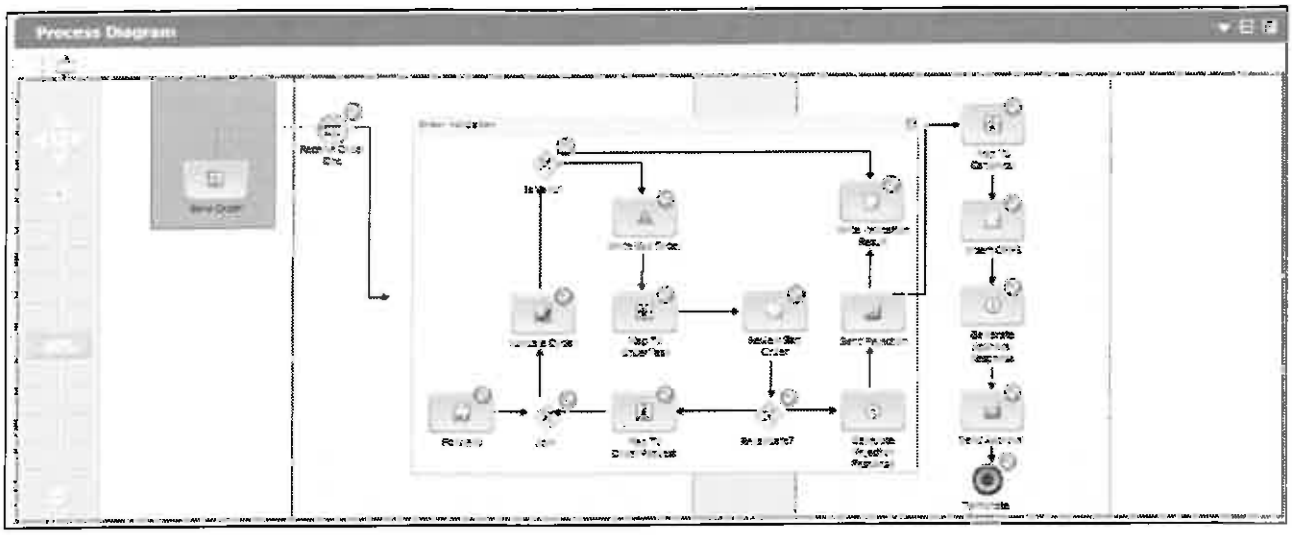


25. Switch to the **Task List Management** page, refresh the list (click Search in the Search Portlet) until the new task is seen:

26. Select and **Accept** the new task, reset the field **RevalidateYN** to **Y**, change the **ProductQuantity** field from **-9** to **9**, and **Complete** the task.



27. Switch back to the **Applications -> Monitoring -> Business -> Process Instances** page. Open the process instance and ensure the order is revalidated, and it is valid. Also, ensure the rest of the steps are executed successfully.



Check Your Understanding

1. The User Task business data was declared as an In/Out parameter. What does this mean at runtime?

2. When interacting with the User Task you set revalidateYN to Y. If you also set the quantity of the product to a positive number, what results did you expect?

3. If you send the process invalid data, what mechanisms could you employ to ensure the process instance does not get stuck in an infinite loop?

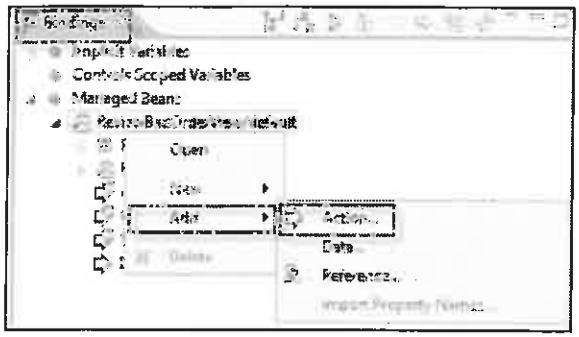
Exercise 12:
Customizing the User Task UI

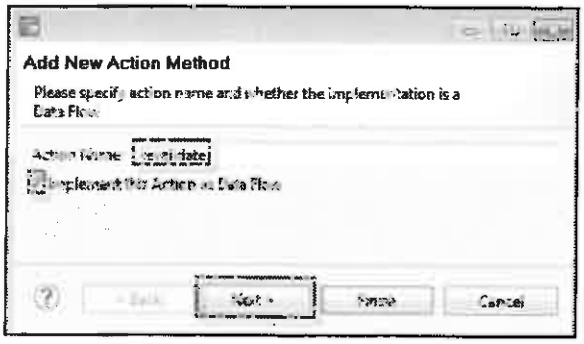
Overview

In this exercise, you will customize the UI (Default view) of your **ReviewBadOrder** User Task. You will change control types, add Action Flows, Command Buttons, and Validators to the view.

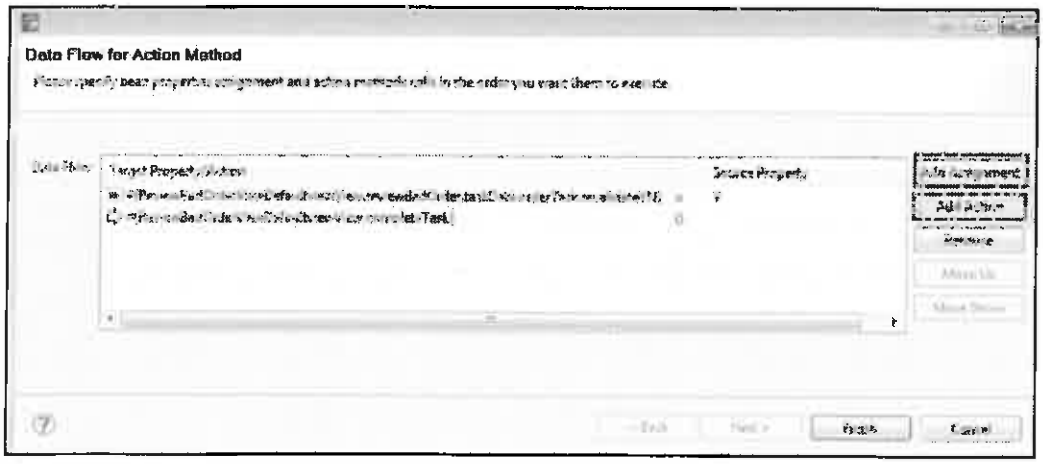
Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Software AG Designer and ensure you are in **UI Development** perspective.
3. Open the file **Tasks/SalesDepartment/ReviewBadOrder/ReviewBadOrderView/Default** from the Solutions view.
4. Open the Bindings view and create a new Action:
   a) Right-click **ReviewBadOrderView/default** and choose **Add -> Action.** Name the Action **revalidate** and implement it as a Data Flow:



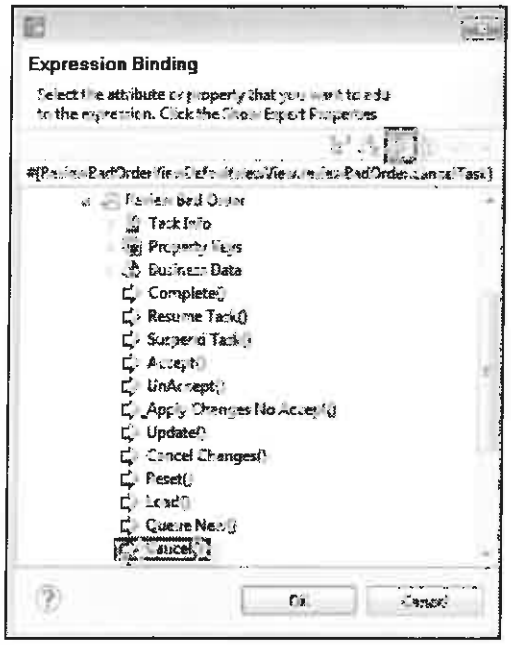


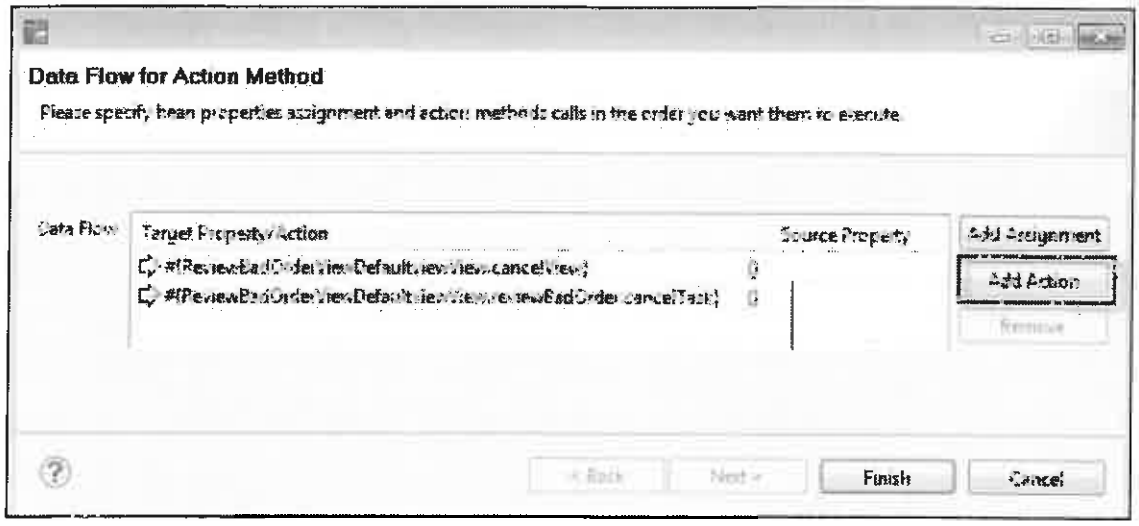
   b) First add an assignment to the Flow Action to set the field **ReviewBadOrderView/default -> Review Bad Order -> Business Data -> OrderTask -> RevalidateYN** to **Y**.
   Then add the existing action **ReviewBadOrderView/default -> Complete Task()** as an Action to the **revalidate** Action Data Flow.



   c) Click **Finish.**

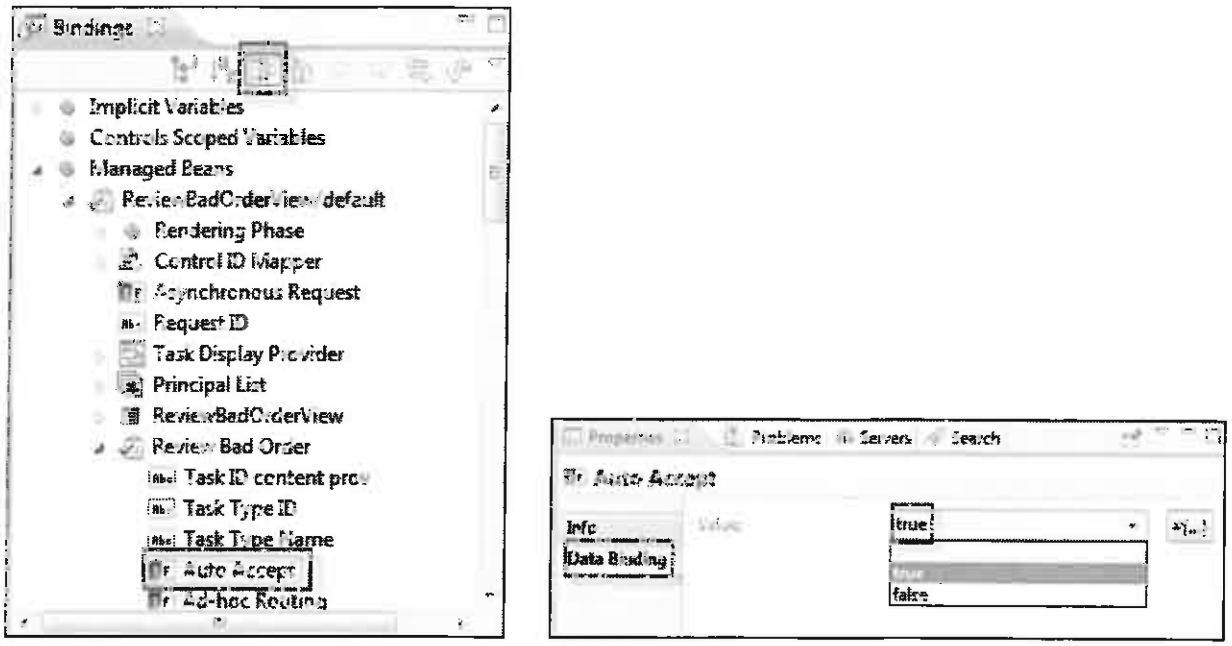
5. In the Bindings view, click the Refresh icon in the upper-right corner.
6. Add another action:
   a) Right-click **ReviewBadOrderView/default** in the Bindings view and add another Action. Name the Action **abort** and implement it as a Data Flow.
   b) Add the existing action **ReviewBadOrderView/default -> cancelView()** as a new Action to the **abort** Action Data Flow.
   c) Add the existing action **ReviewBadOrderView/default -> Review Bad Order -> Cancel()** action as a new Action to the **abort** Action Flow.
   Note: To access the **Cancel()** action mentioned above, click to show the Expert Properties.



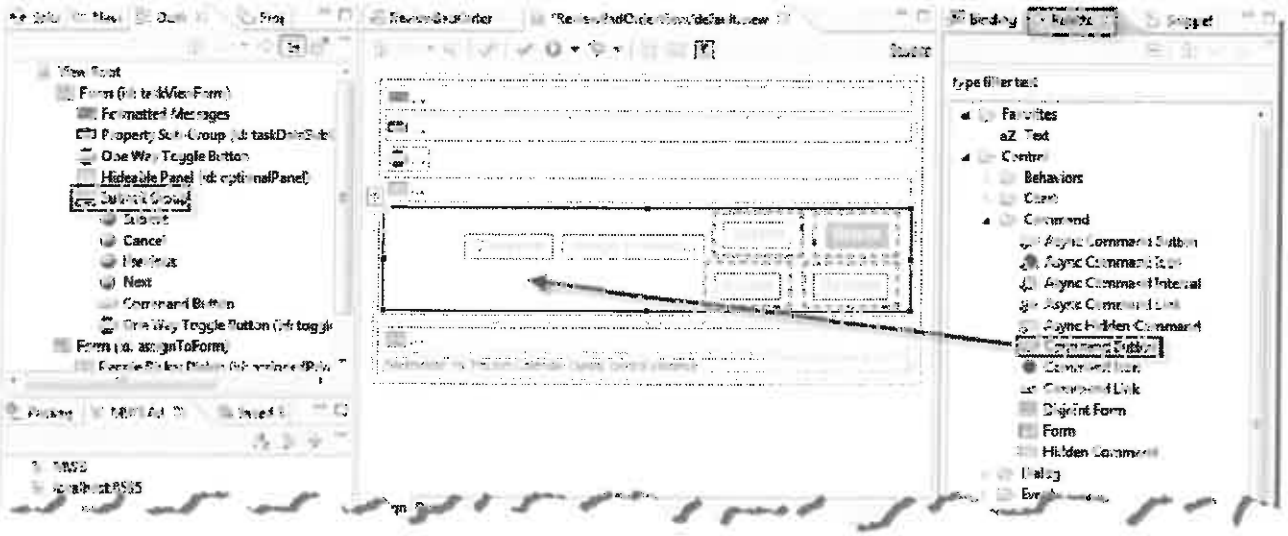


   d) Click **Finish**.

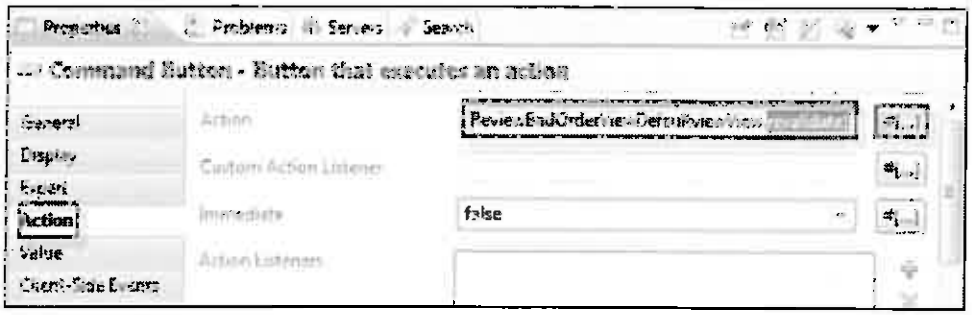
7. The User Task should be accepted automatically when first opened by an assignee. To achieve this, select the property **ReviewBadOrderView/default -> Review Bad Order -> Auto Accept** from the Bindings view. Be sure to click on the toolbar button (Show Expert Properties) at the top of the Bindings view as shown on the screen shot in order to locate the **Auto Accept** property. Modify the value of the **Auto Accept** property by using the Properties view. Choose **true** as new value on the Data Binding tab.



8. From the Palette, drag and drop a Command Button into the Submit Group section.
Note: You can also drag and drop a Command Button to the Outline view.



9. In the Properties view of the added Command Button, change the **Value** field on the Value tab to **Revalidate** and set the **Action** field on the Action tab to the **Revalidate()** Action Data Flow.

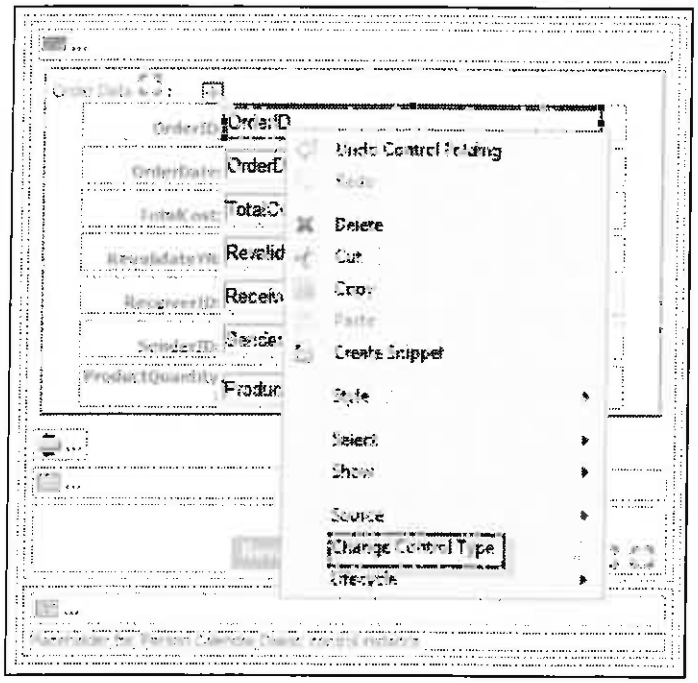


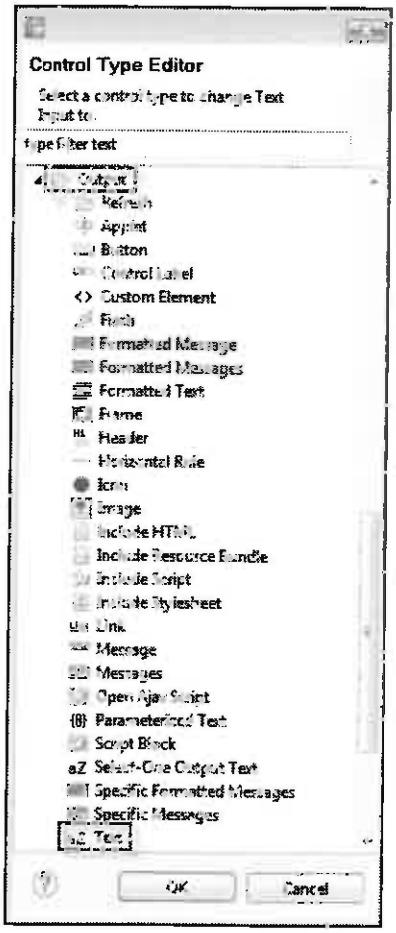
10. Delete the buttons **Complete**, **Assign to Users**, **Submit**, **Accept**, and **Release** from the Submit Button Group section.

11. From the Palette, add another Command Button to the view **ReviewBadOrderView/default.view** in the Group panel to the right of the **Revalidate** button.

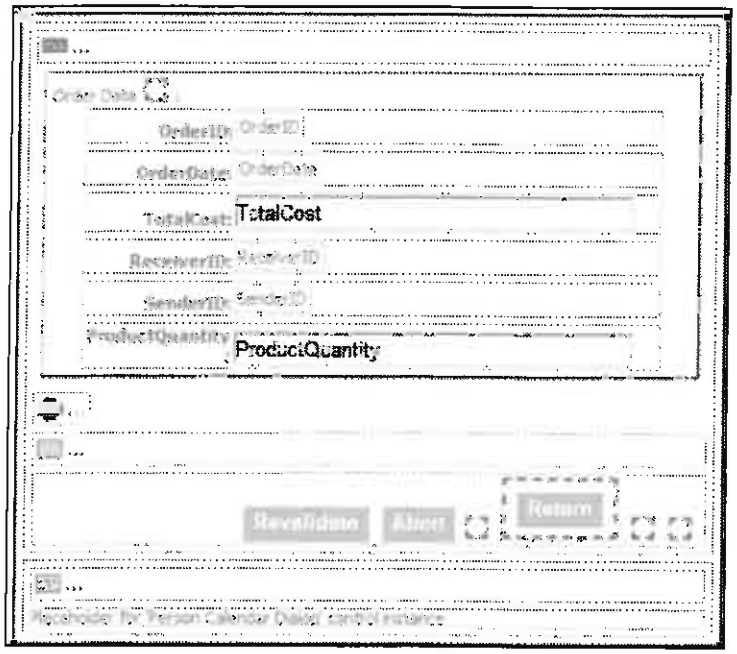
12. In the Properties view of the new Command Button, **set** the **Value** field to **Abort** and set the **Action** to the **Abort()** Action Flow.

13. Change the Output control types for the fields **OrderID**, **OrderDate**, **ReceiverID**, and **SenderID** from Input - Text Input to **Output - aZ Text**.





14. Remove the Property Line with the label **RevalidateYN** including the Text Input control from the view.

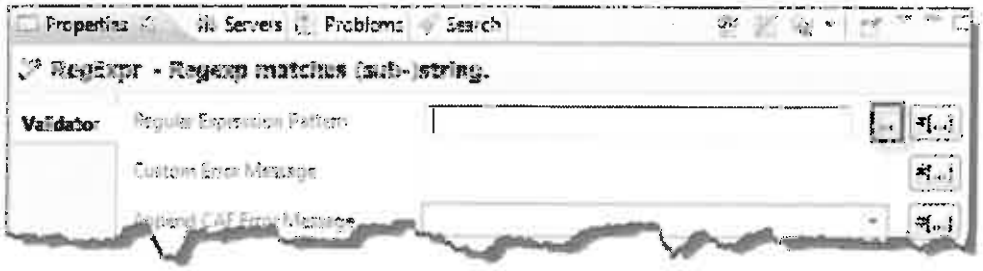


15. Add a Validator to the TotalCost input field:

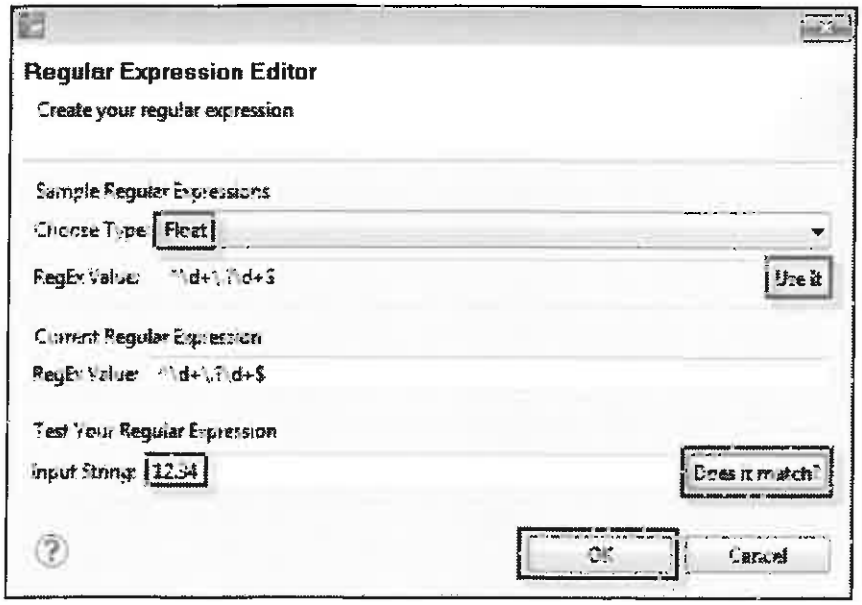
a) Select Text Input control **TotalCost** on the view **ReviewBadOrderView/default.view.**

b) Open the Validation tab in the Properties view. Add a Validator of type **RegExpr**.

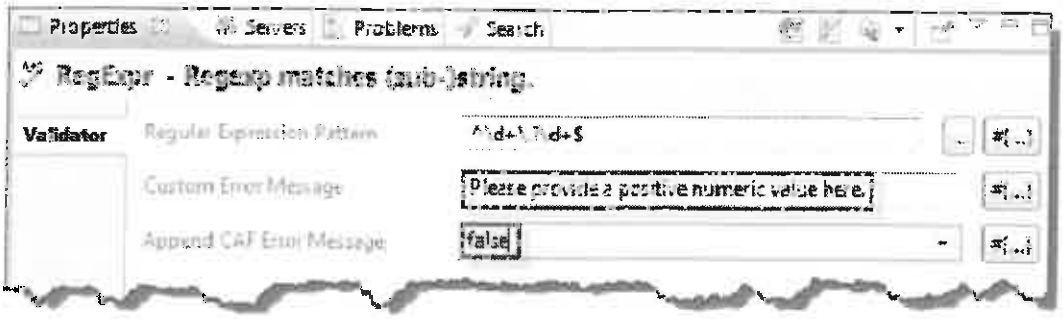
c) Double-click the RegExpr icon in the Validators list. Assign a Regular Expression Pattern by clicking the browse button:



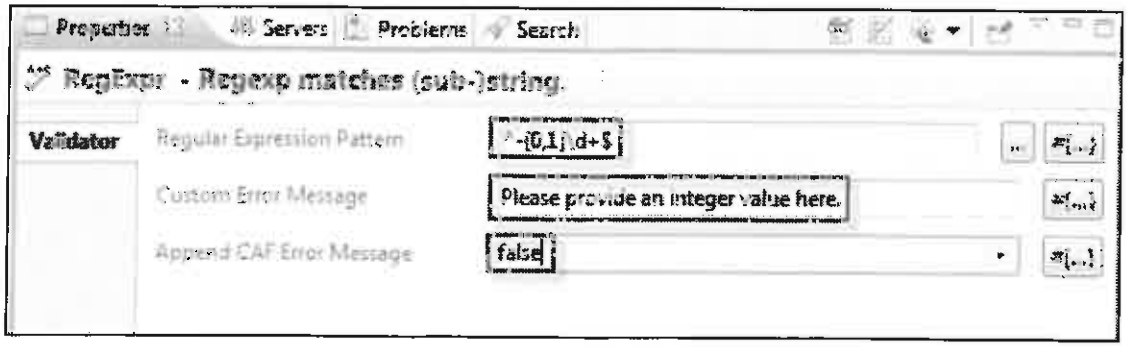
d) In the Regular Expression Editor, select type **Float** and click **Use It**. Test the regular expression in the editor by providing some input strings and clicking **Does it match?** Finally, click **OK** to accept the regular expression.



e) Specify **Please provide a positive numeric value here** as custom error message and disallow CAF error messages to be appended:

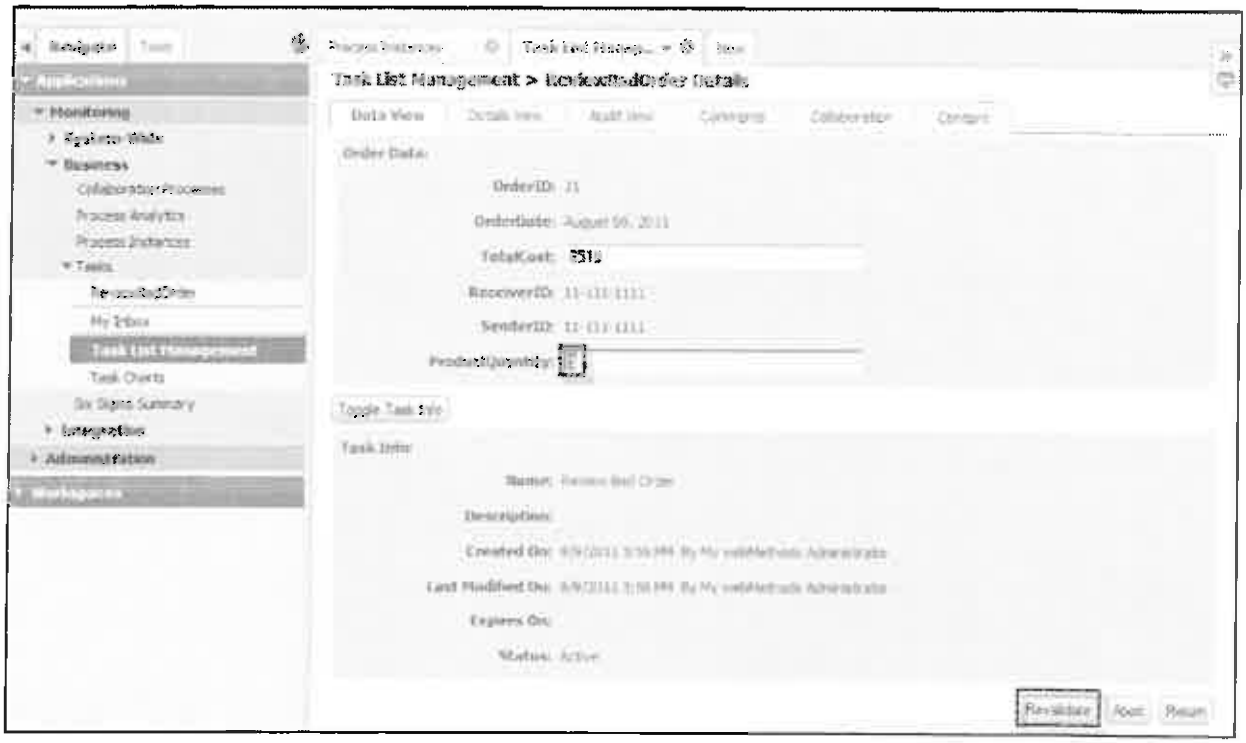


16. In the same way, add another Validator to the ProductQuantity input field:
    a) Select Text Input control **ProductQuantity** on the default view.
    b) Open the Validation tab in the Properties view. Add a Validator of type **RegExpr**.
    c) Double-click the RegExpr icon in the Validators list. Provide **^-{0,1}\d+$** as the Regular Expression Pattern.
    Note: This pattern allows positive and negative integer values.
    Specify **Please provide an integer value here** as custom error message and disallow CAF error messages to be appended:

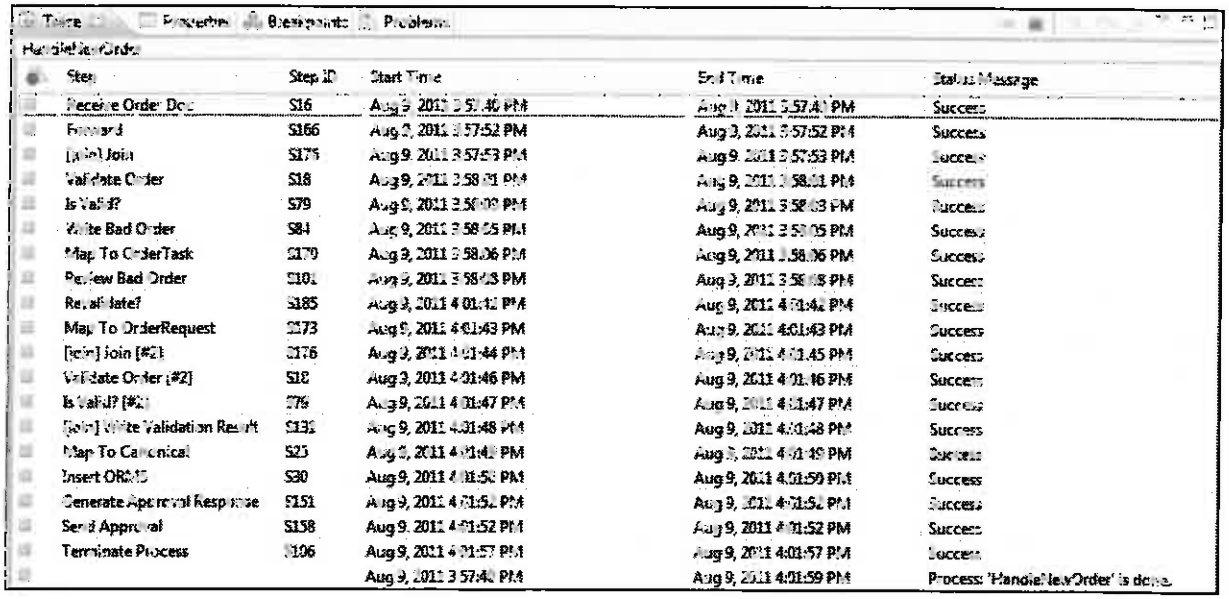


17. Save your changes. Republish the **SalesDepartment** CAF project to the MWS.
18. Switch to the **Process Debugging** perspective and open the **HandleNewOrder** process.
19. Start a debugging session. For the IS input document, load the file ***<workshop_dir>*\Exercise12\Resources\Ex12_input1.txt**. Step through the process.
20. The process should queue a new User Task instance:
    a) Select the User Task instance from the Task List Management page in My webMethods Monitoring. Open the User Task and inspect its Business Data on the Task UI.
    b) To test your Validators, specify a negative numeric value (e.g. -1234.56) for the input field **TotalCost**. Also try to assign a non-numeric value to **ProductQuantity**.

c) Finally reassign **6510** to **TotalCost** and change the **ProductQuantity** from -9 to **9** and click **Revalidate** to complete the User Task.



21. Because of the corrected ProductQuantity, the order is now valid. Ensure the remaining steps execute successfully in the Process Debugger.



Step	Step ID	Start Time	End Time	Status Message
Receive Order Doc	S16	Aug 9, 2011 3:57:46 PM	Aug 9, 2011 3:57:47 PM	Success
Forward	S166	Aug 9, 2011 3:57:52 PM	Aug 9, 2011 3:57:52 PM	Success
[join] Join	S175	Aug 9, 2011 3:57:53 PM	Aug 9, 2011 3:57:53 PM	Success
Validate Order	S18	Aug 9, 2011 3:58:01 PM	Aug 9, 2011 3:58:01 PM	Success
Is Valid?	S79	Aug 9, 2011 3:58:03 PM	Aug 9, 2011 3:58:03 PM	Success
Write Bad Order	S84	Aug 9, 2011 3:58:05 PM	Aug 9, 2011 3:58:05 PM	Success
Map To OrderTask	S179	Aug 9, 2011 3:58:06 PM	Aug 9, 2011 3:58:06 PM	Success
Review Bad Order	S101	Aug 9, 2011 3:58:43 PM	Aug 9, 2011 3:58:43 PM	Success
Revalidate?	S185	Aug 9, 2011 4:01:42 PM	Aug 9, 2011 4:01:42 PM	Success
Map To OrderRequest	S173	Aug 9, 2011 4:01:43 PM	Aug 9, 2011 4:01:43 PM	Success
[join] Join [#2]	S176	Aug 9, 2011 4:01:44 PM	Aug 9, 2011 4:01:45 PM	Success
Validate Order [#2]	S18	Aug 9, 2011 4:01:46 PM	Aug 9, 2011 4:01:46 PM	Success
Is Valid? [#2]	S79	Aug 9, 2011 4:01:47 PM	Aug 9, 2011 4:01:47 PM	Success
[join] Write Validation Result	S131	Aug 9, 2011 4:01:48 PM	Aug 9, 2011 4:01:48 PM	Success
Map To Canonical	S23	Aug 9, 2011 4:01:49 PM	Aug 9, 2011 4:01:49 PM	Success
Insert ORMS	S30	Aug 9, 2011 4:01:59 PM	Aug 9, 2011 4:01:59 PM	Success
Generate Approval Response	S151	Aug 9, 2011 4:01:52 PM	Aug 9, 2011 4:01:52 PM	Success
Send Approval	S158	Aug 9, 2011 4:01:52 PM	Aug 9, 2011 4:01:52 PM	Success
Terminate Process	S106	Aug 9, 2011 4:01:57 PM	Aug 9, 2011 4:01:57 PM	Success
		Aug 9, 2011 3:57:46 PM	Aug 9, 2011 4:01:59 PM	Process 'HandleNewOrder' is done.

22. Start another debugging session. For the IS input document, load the file *<workshop_dir>*\Exercise12\Resources\Ex12_input2.txt. Step through the process.

23. The process should queue a new User Task instance. Select the User Task instance from the Task List Management page in My webMethods. Open the User Task and inspect its Business Data on the Task UI. Leave the Business Data unchanged and click **Abort** to cancel the User Task.

24. Ensure the task is marked as **Canceled** in My webMethods under **Applications -> Monitoring -> Business -> Tasks -> Task List Management.**

25. Execute the remaining steps in the Process Debugger to complete the process.

Check Your Understanding

1. You implemented the actions as Data Flow. What other language could you have employed?

2. You removed the RevalidateYN field from the view. The process works as expected. Where was the RevalidateYN field set?

3. What caused the last User Task you worked on to be marked as Canceled?

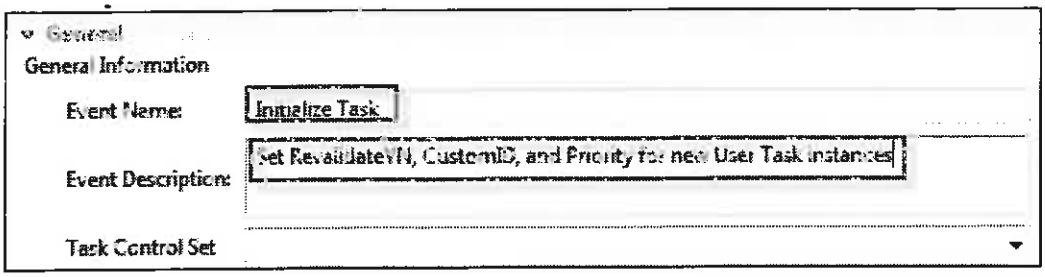
Exercise 13:
User Task Events

Overview

In this exercise, you will enhance your **ReviewBadOrder** User Task by adding a Task Event. The event will be used to automatically initialize some User Task fields by taking values from the Task Business Data whenever the User Task is queued.

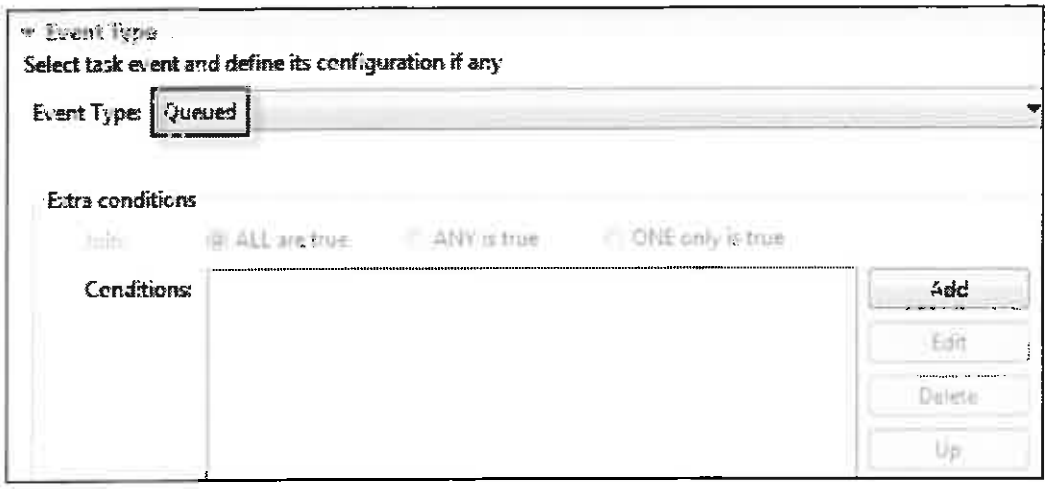
Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Software AG Designer and ensure you are in **UI Development** perspective.
3. In the Solutions view, double click the User Task **ReviewBadOrder** to open the Task Editor. Ensure the Task Configuration Editor is displayed in the Introduction tab of the editor pane.
4. Click the **Events** tab at the bottom of the Task Editor.
5. Add an event to your User Task.
   a) Provide the following **General** Information:

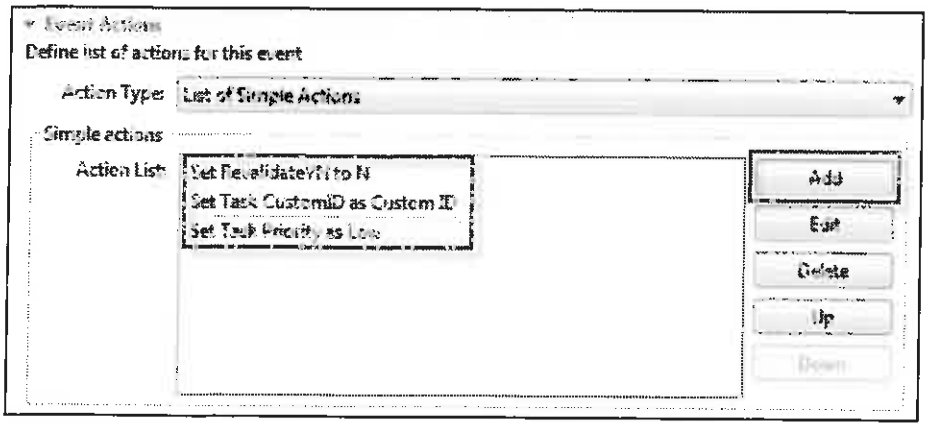


   Click **Save** in the menu bar.

   b) Set the **Event Type** to **Queued**. Leave the **Extra Conditions** empty.



c) Add three **Event Actions** of type **List of Simple Actions** using the lower portion of the Events window as shown by the screen shot below.
Notes: To set the **RevalidateYN** field, select the predefined Task Action **Set Business Data Field** and browse for the Data Field **RevalidateYN** from the Business Data, then set the **Value** text field to **N.**
Custom ID to be assigned to the Task Custom ID is part of the Business Data also.



d) Finally click **Save** in the menu bar.

6. Republish the **SalesDepartment** CAF project to the My webMethods Server.
7. Switch to **Process Debugging** perspective and load your **HandleNewOrder** process model.
8. Start a debugging session. For the IS input document, load an invalid order from the file ***<workshop_dir>*\Exercise13\Resources\Ex13_input.txt**. Step through the process.
9. In Task List Management, ensure the User Task's Custom ID is set to **K55-254-8595** and the User Task's priority is set to **4-low.**
Note: If the column CUSTOM ID isn't displayed, you can add it by clicking the button in the header of the User Task list, selecting Properties and then selecting CUSTOM ID from the list of Available Columns. Make sure you Apply your changes.

10. Open the User Task instance, change the **ProductQuantity** from **-9** to **9**, and click **Revalidate** to complete the User Task.
11. Complete your process in Debugger.

Check Your Understanding

1. When is the User Task priority set?

2. Could the RevalidateYN, Priority and Custom ID be set in three events, or must they be consolidated into a single event?

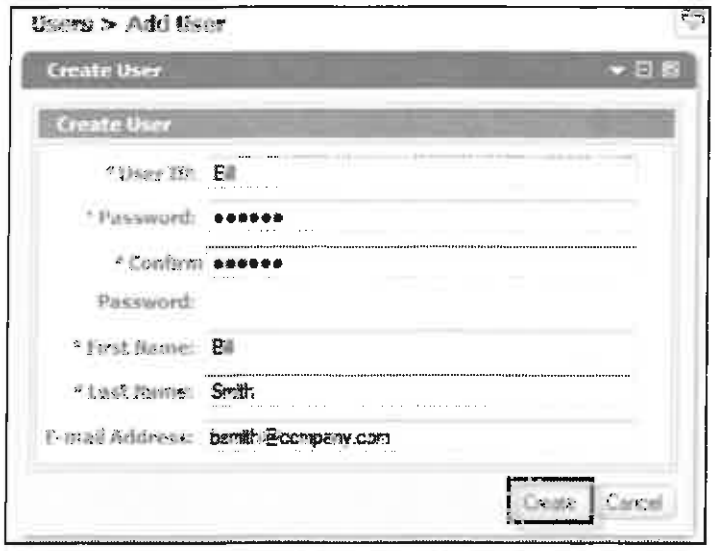
Exercise 14:
User Task Assignment

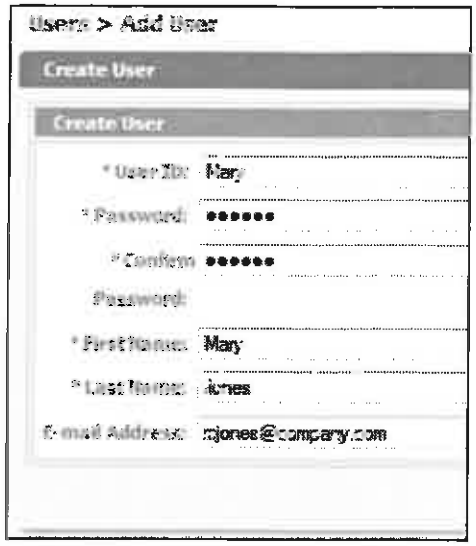
Overview

In this exercise, you will first create users, groups and a role. You will enhance your ReviewBadOrder User Task to use an Assignment to automatically assign new User Task instances to your specified role.

Steps

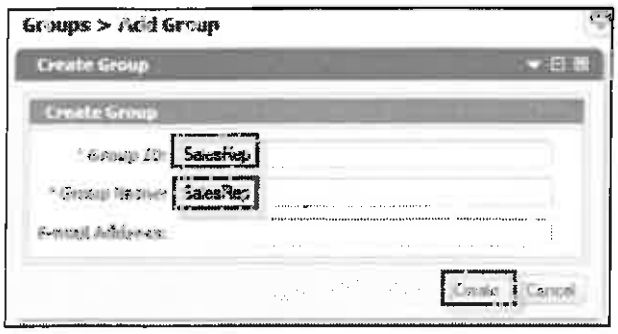
1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Login to My webMethods as **Administrator/manage**. Navigate to the **Applications** -> **Administration** -> **System-Wide** -> **User Management** -> **Users** page.
   Add two users with the following details:

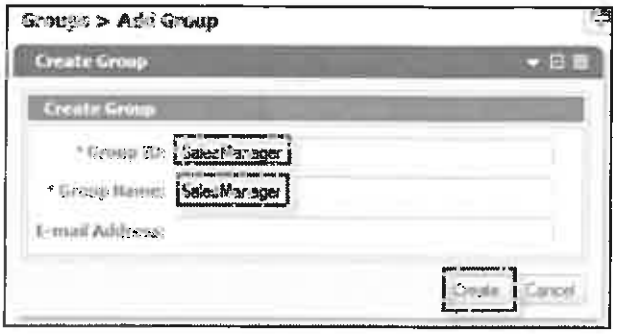




   Password is always **manage**. Confirm with **Create** and **Save**.

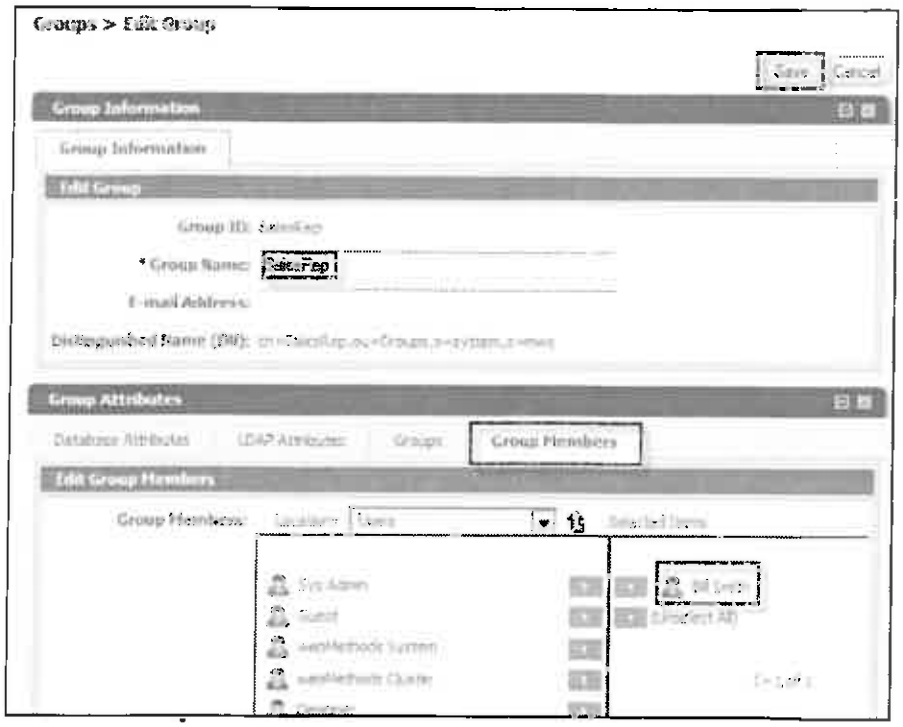
3. Switch to the **Applications** -> **Administration** -> **System-Wide** -> **User Management** -> **Groups** page. Add two groups with the following details:





   Confirm with **Create** and **Save**.

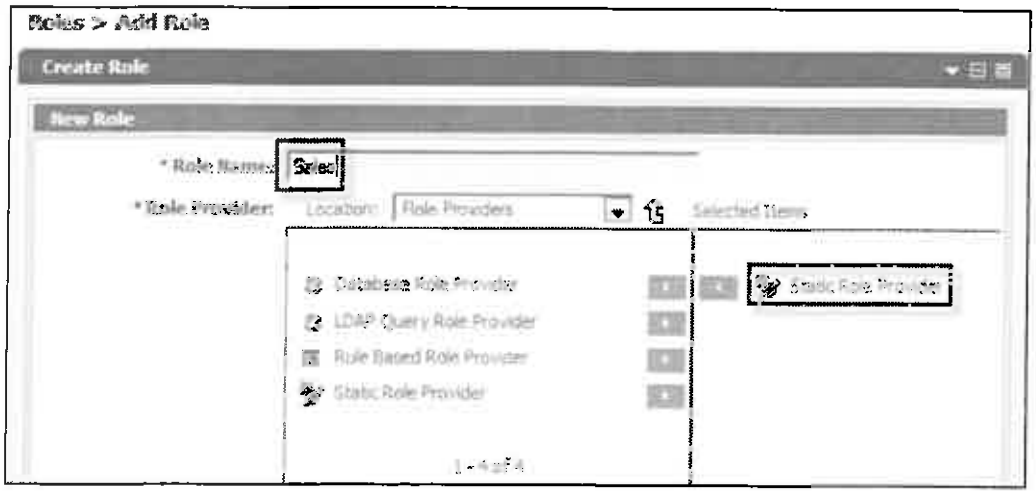
4. On the **Applications -> Administration -> System-Wide -> User Management -> Groups** page, edit the Group Members of the **SalesRep** group. Select and add user **Bill Smith** as a member of the group.



Save your changes.

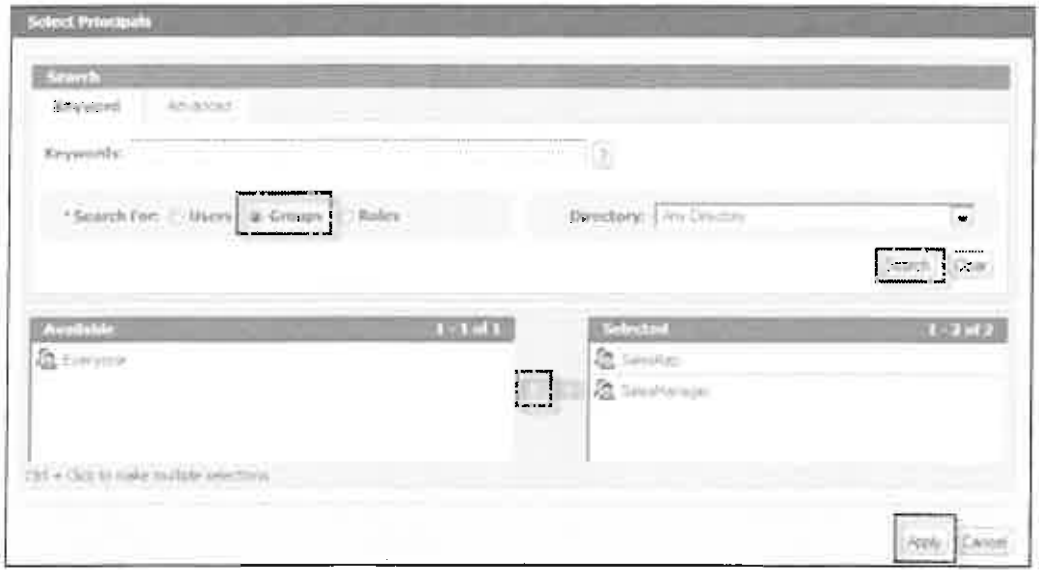
5. On the **Applications -> Administration -> System-Wide -> User Management -> Groups** page, edit the Group Members of the **SalesManagers** group. Select and add the user **Mary Jones** as a member of the group. Save your changes.

6. Switch to the **Applications -> Administration -> System-Wide -> User Management -> Roles** page. Add a role named **Sales** with a **Static Role Provider:**



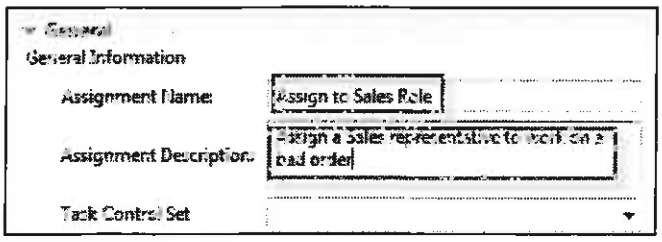
Confirm with **Create Role.**

7. Edit the Role Attributes of the **Sales** role to have the groups **SalesRep** and **SalesManagers** as its members:

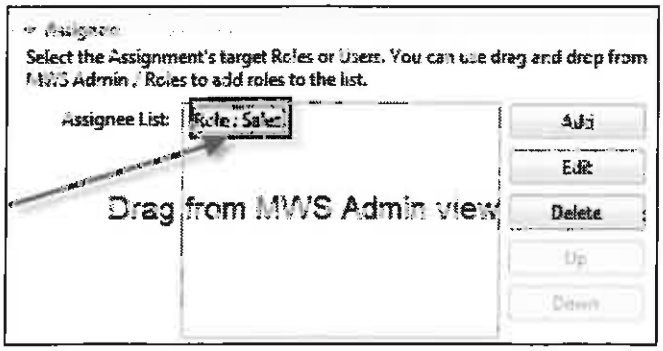


Apply and **Save** your changes.

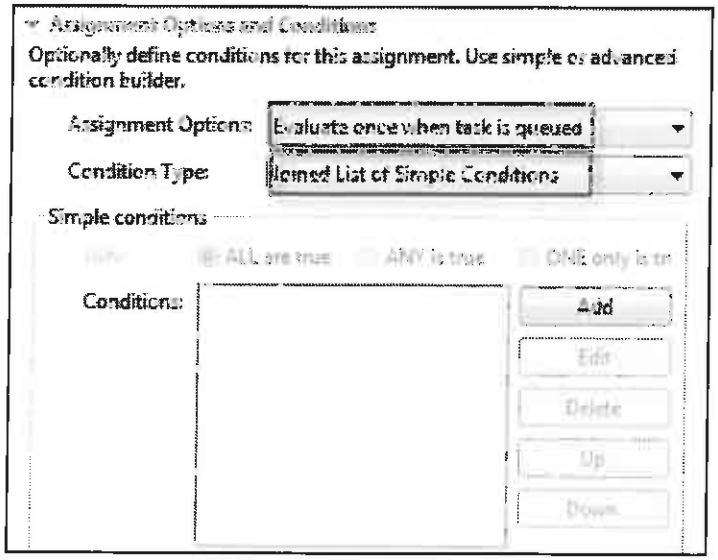
8. Launch Software AG Designer and ensure you are in **UI Development** perspective.
9. Ensure the MWS Admin view (lower left corner of Designer) is available in your perspective. Expand the tree view for MWS within the MWS Admin view. If asked for authentication, provide **SysAdmin/manage**.
10. Switch to the Solutions view. Open the User Task **ReviewBadOrder** in the Task Editor.
11. Click the Assignments tab at the bottom of the **ReviewBadOrder** Task Editor.
12. Add an assignment to your User Task:
    a) Provide the following **General** information:



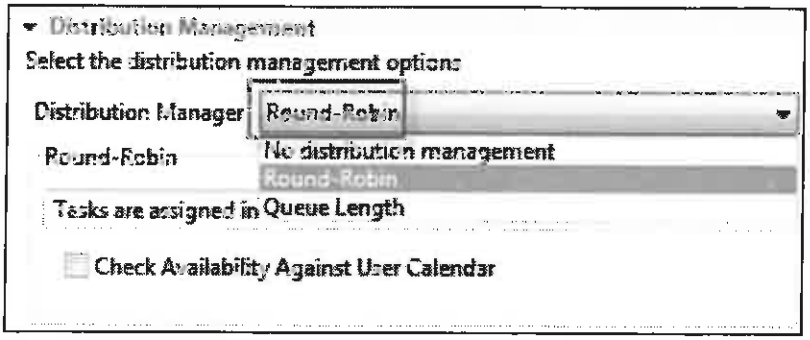
    b) Use drag and drop from the MWS Admin view (or use the **Add** button) to add role **Sales** as an assignee to the Assignee List:



c) Specify the following Assignment Options and Conditions:



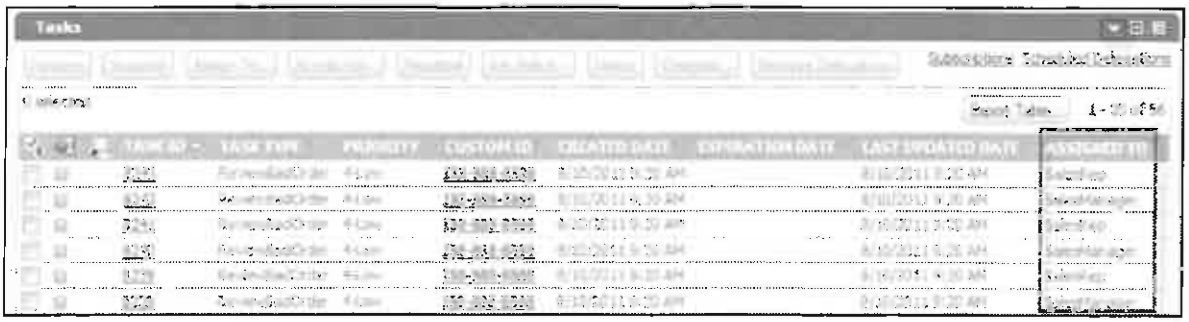
d) In the Distribution Management section select **Round Robin** as Distribution Manager:



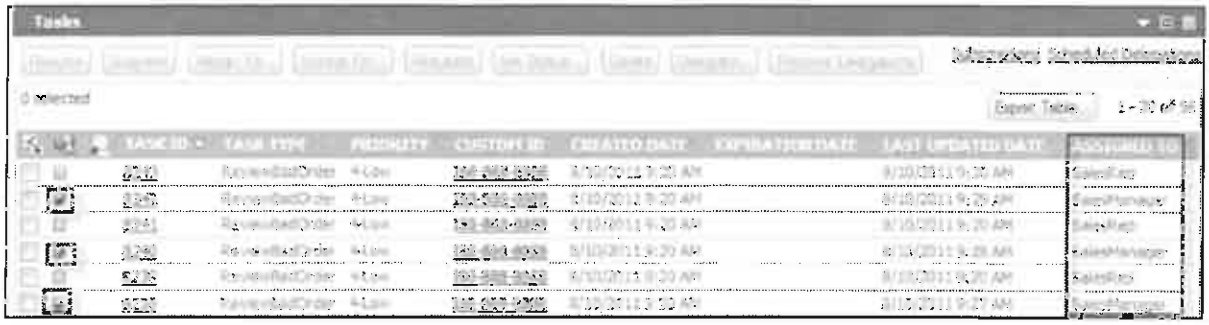
e) Save the User Task assignment.

13. Use the Servers view to republish the **SalesDepartment** project to My webMethods Server.
14. Start a new instance of the HandleNewOrder process by submitting a new Order document using a test web page. To do so, open the following in a web browser: ***<workshop_dir>*\Exercise14\Resources\Ex14_submit.html**.
    a) Click the **Submit** button.
    If prompted for authentication, enter **Administrator/manage** as username/password.
    b) Perform step 14a) at least 5 more times (use the back button of your browser to resubmit).

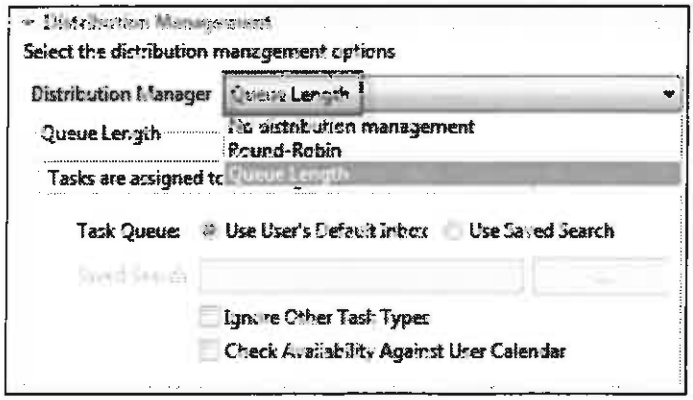
15. Use a browser session to login to My webMethods as **Administrator/manage**.
16. Open the **Applications** -> **Monitoring** -> **Business** -> **Tasks** -> **Task List Management** page. Customize the **Task List Management** table to include the **ASSIGNED TO** column in the tasks list, if not already shown. Ensure the User Tasks are assigned to the SalesRep and SalesManagers. The assignments should be round-robin:



17. Click on Task ID associated with the oldest active User Task assigned to the **SalesManagers**.
18. In the User Task UI, change the ProductQuantity from -9 to **9**. Click Revalidate.
19. Perform steps 17 and 18 for all other active User Tasks assigned to **SalesManagers** (not SalesReps). The Sales Managers should now have zero Active tasks and the SalesRep should have several:

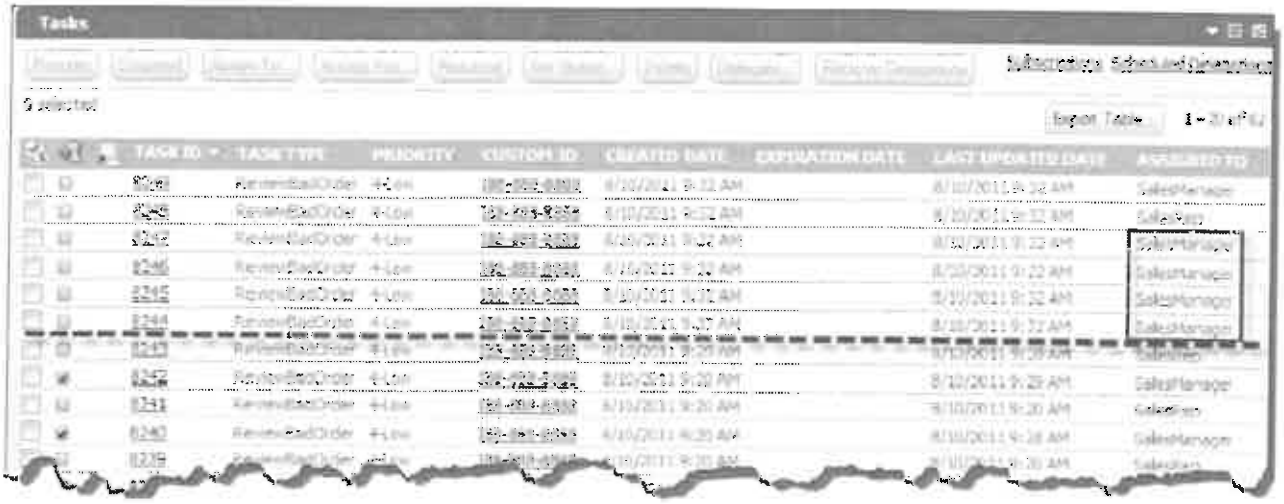


20. In Designer, go to the Assignment configuration of your **ReviewBadOrder** User Task and change the Distribution Management from Round-Robin to **Queue Length**:



21. Save your User Task changes.
22. Use the Servers view to republish the **SalesDepartment** project to My webMethods Server.

23. Start new process instances:
    a) Double-click the file **<*workshop_dir*>\Exercise14\Resources\Ex14_submit.html**.
    b) Click the **Submit** button. If prompted, enter **Administrator/manage** as the username/password combination.
    c) Perform step 23b) at least 5 more times (use the back button of your browser to resubmit).
24. Switch back to My webMethods. Click the **Search** button to refresh the User Task list on the Task List Management page. Inspect the distribution of new User Tasks:



25. Finally, ensure that all active User Tasks are completed by changing **ProductQuantity** to **9** and selecting the **Revalidate** button.
    If more than 20 User Tasks are available, use the **Next>>** link at the bottom of the page to view additional User Tasks.
    Note: The default number of User Task instances that appear in the list can be adjusted by selecting the Task List page **Properties**, then selecting the **Preferences** tab.

Check Your Understanding

1. Does round-robin distribution take the number of User Tasks assigned to an individual, group or role into account when assigning a User Task? No

2. In step 23 you launched multiple process instances. The order in which User Tasks were assigned did not correspond to a round-robin distribution. Why?
   Sales Manager had an empty queue

3. Can a User Task Assignment be based on a Business Calendar?
   Yes

Exercise 15:
User Task Expiration based on Business Calendars

Overview

In this exercise, you will set up a Business Calendar in MWS. The Business Calendar defines workdays and business hours. It will be used for your **ReviewBadOrder** User Task to set up task expiration to cross over working business days.

Steps

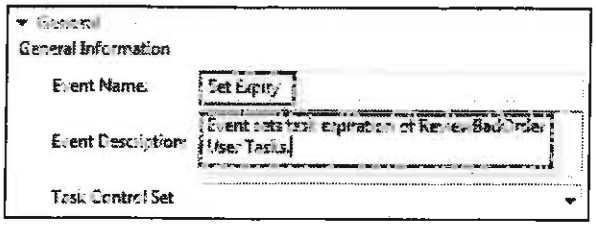
1. Ensure that the time zone and time setting of your VM mirrors your local time zone and time setting. If necessary, double-click the clock in the lower right corner of the Windows task bar to open the Windows Date and Time Properties of your VM, and adjust the settings. Only in the case you have changed the Windows date and time settings of your VM, you should restart your Integration Server, My webMethods Server, and Optimize Analytic Engine.
2. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
3. Login to My webMethods as **Administrator/manage**.
4. Navigate to **Applications** -> **Administration** -> **System-Wide** -> **Calendars Management** -> **Business Calendars**. Click **Create New Calendar** to add a new Business Calendar to MWS with the following details:

Data	Value
Name	**USCalendar**
Lookup Name	**USCalendar**
Time Zone	***<Current>*** **(default)**
Workday 1	***<Today>*** **8:00 am TO *<30 minutes later than current system time>***
Workday 2	***<Tomorrow>*** **8:00 am TO 5:00 pm (8:00 - 17:00)**

   At the end, ensure you click the **Create New** button in My webMethods.
5. Open Software AG Designer and switch to the **UI Development** perspective. Ensure the MWS Admin view (lower left corner of Designer) is available in your perspective. Refresh the tree view for MWS in the MWS Admin view. If asked for authentication, use **Sysadmin/manage**. Drill down to **MWS** -> **Business Calendars**. Your **USCalendar** Business Calendars should appear in the tree.
6. Use the Solutions view to open the User Task **ReviewBadOrder** (**Tasks** -> **SalesDepartment** -> **ReviewBadOrder**). In the Task Editor select the **Events** tab.

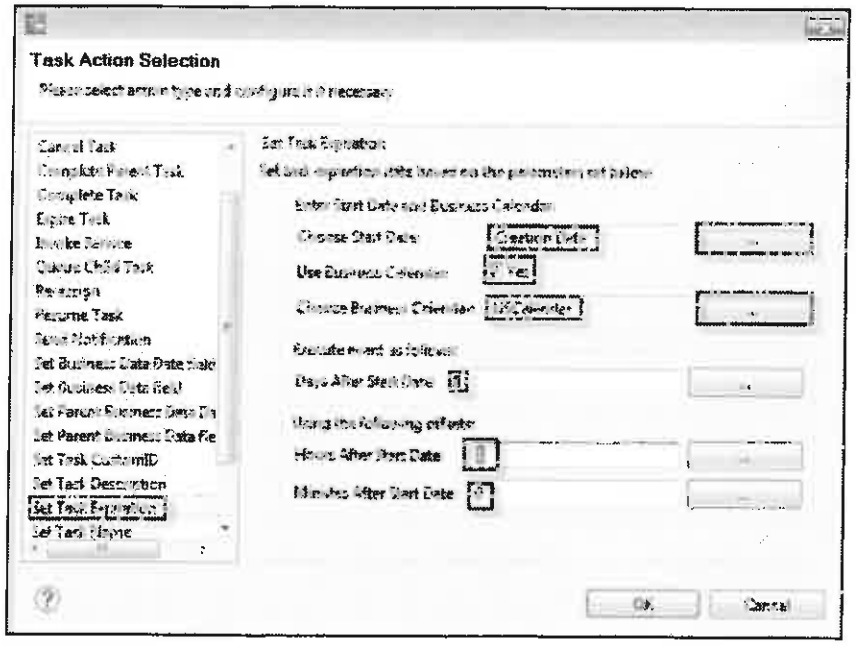
7. Add a second Event to your **ReviewBadOrder** User Task:

   a) Provide the following **General** Information:

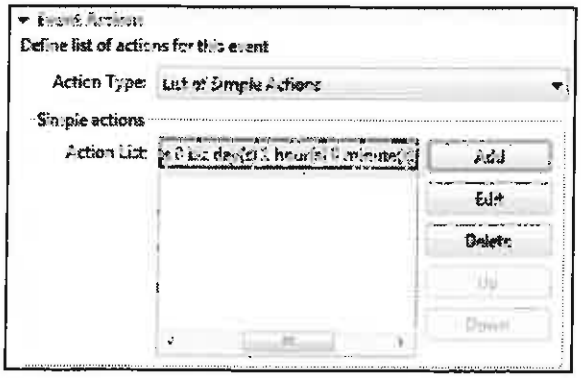


   b) Set the **Event Type** to **Queued**. Leave the **Extra Conditions** empty.

   c) Add an **Event Action** of type **List of Simple Actions** using the lower portion of the Events window. Select the existing simple action **Set Task Expiration** to be executed. Configure the simple action to set the task expiration based on your **USCalendar** Business Calendar and select/provide the additional settings as shown on the screen shot below.
   Note: Field **Creation Date** can be selected from **Task Info**.

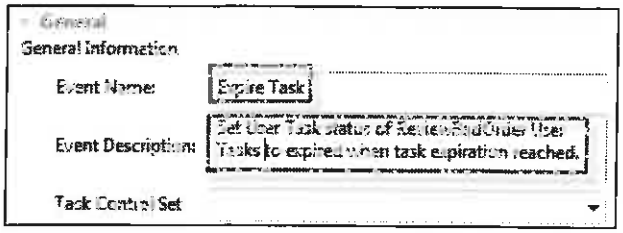


   Click **OK**. Your Event Action should now look like this:

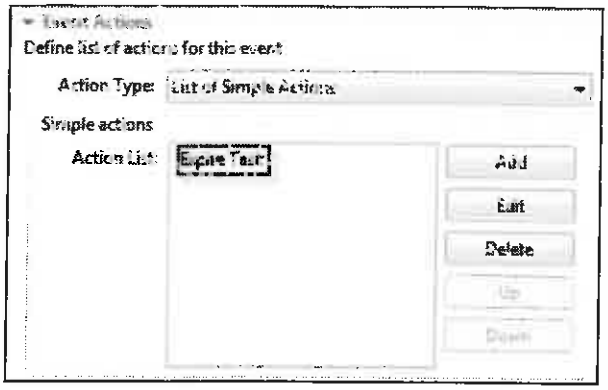


   d) Save your modified User Task.

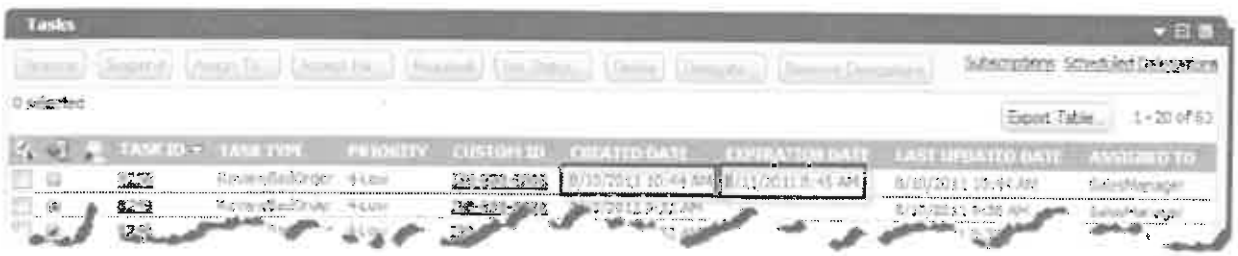
8. Add a third Event to your **ReviewBadOrder** User Task:
   a) Provide the following **General** Information:



   b) Set the **Event Type** to **Expiration Date Reached.** Leave the **Extra Conditions** empty.
   c) Add an **Event Action** of type **List of Simple Actions** using the lower portion of the Events window. Select the existing simple action **Expire Task** to be executed.



   d) Save your modified User Task.
9. Use the Servers view to republish your **SalesDepartment** project to My webMethods Server. If prompted for Authentication use **SysAdmin/manage.**
10. Start a new instance of the HandleNewOrder process by submitting a new Order document using a test web page. To do so, open the following in a web browser: ***<workshop_dir>*\Exercise15\Resources\Ex15_submit.html** and click **Submit.**
11. Login to My webMethods as **Administrator/manage.** Navigate to **Applications -> Monitoring -> Business -> Tasks -> Task List Management.**
12. Note **Created Date** and **Expiration Date** of the newly created User Task instance. The User Task is set to expire one hour from its start time as defined by the Workdays created in the Business Calendar. That is, if there are 30 minutes from the creation of the User Task to the end of today's Business Calendar workday, there should be 30 minutes remaining to complete the User Task from the beginning of tomorrow's Business Calendar workday. If there are 25 minutes for today, there should be 35 minutes remaining for tomorrow, etc.



13. Do NOT complete this task. We want to test the **Expire Task** User Task Event that you created. This event will set the User Task's status to **EXPIRED** when its Expiration Date/Time is reached. Check this User instance after the Expiration Date/Time is reached to make sure the status changes to EXPIRED.

Check Your Understanding

1. Why are Business Calendars important for User Task expiration?
   Ensure operators are available when tasks are up for renew.
2. In step 12, you were asked to verify the User Task expiration. Why did the expiry time roll over to tomorrow's date? It overlapped the end of the business day.
3. Can you complete a User Task outside of a business day window? For example, could you complete the User Task above in, say, 90 clock minutes from the creation time of the User Task? Tasks set to expire the day following the current day, yes.
4. When will the Expire Task Event fire? When the expiration is reached
5. Where could you inspect User Task instance-related error messages? Server Log.

Exercise 16:
User Privileges

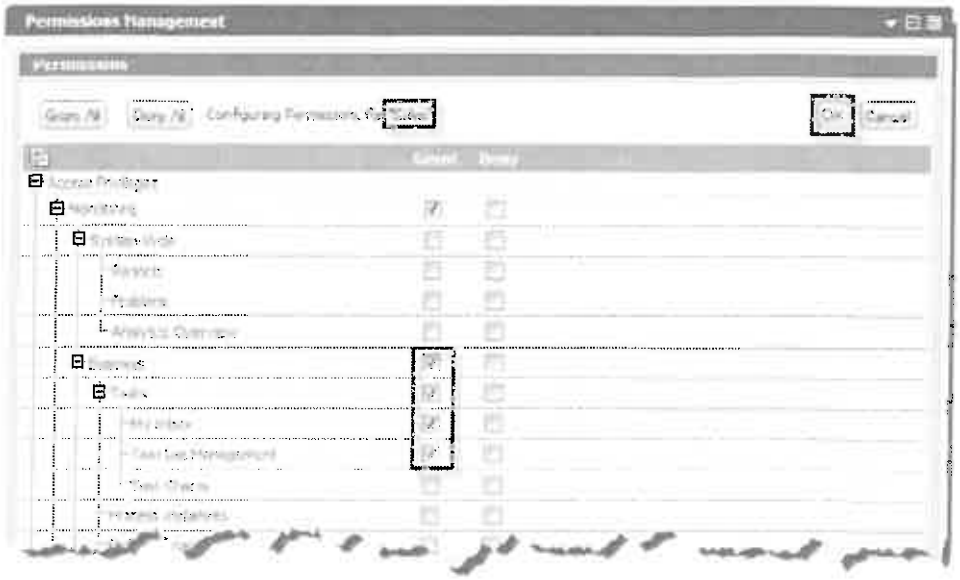
Overview

In this exercise, you will configure User Privileges for the users defined in the previous exercise. This will give the users authorization to login to My webMethods and work on assigned User Tasks.

Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Login to My webMethods as **Bill/manage**.
3. You will notice that Bill has no access privileges in My webMethods. Logout of My webMethods.
4. Login to My webMethods as **Administrator/manage**.
5. Define permissions for My webMethods pages:
   a) Navigate to the **Applications -> Administration -> System-Wide -> Permissions Management page.**
   b) Select Resource Type **My webMethods Application** and click **Next>**. Click **Add Users/Groups/Roles...** and search for **Roles**. Select the **Sales** role and click **Add**.

c) Grant the following Access Privileges to the **Sales** role:
Access Privileges-> Monitoring -> Business -> Tasks -> Task List Management
Access Privileges-> Monitoring -> Business -> Tasks -> My Inbox

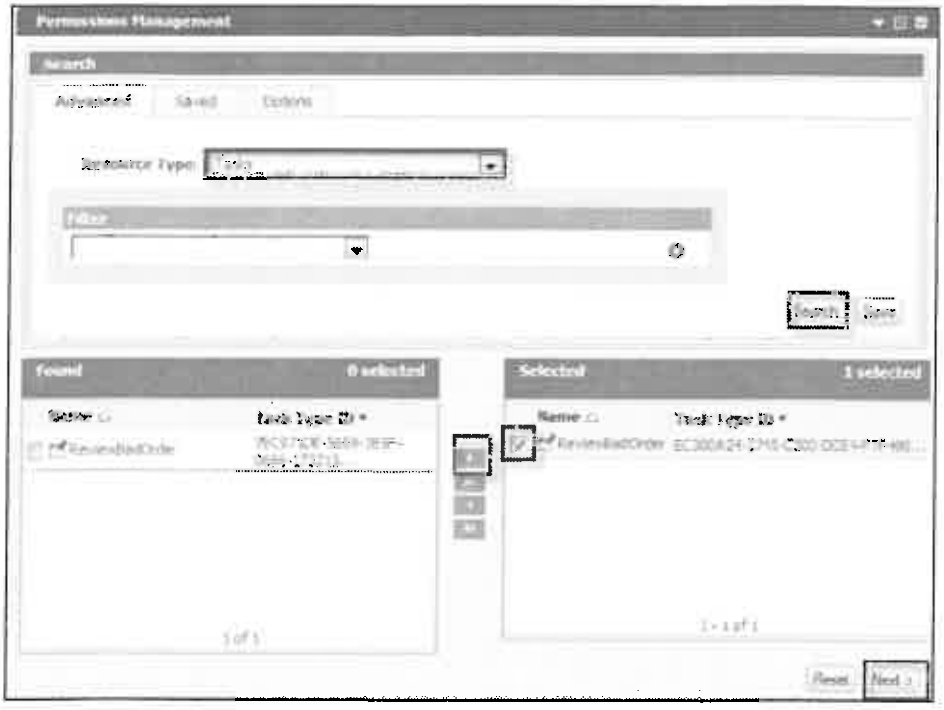


Click **OK** and **Apply** to save your changes.

6. Define permissions for ReviewBadOrder User Tasks:

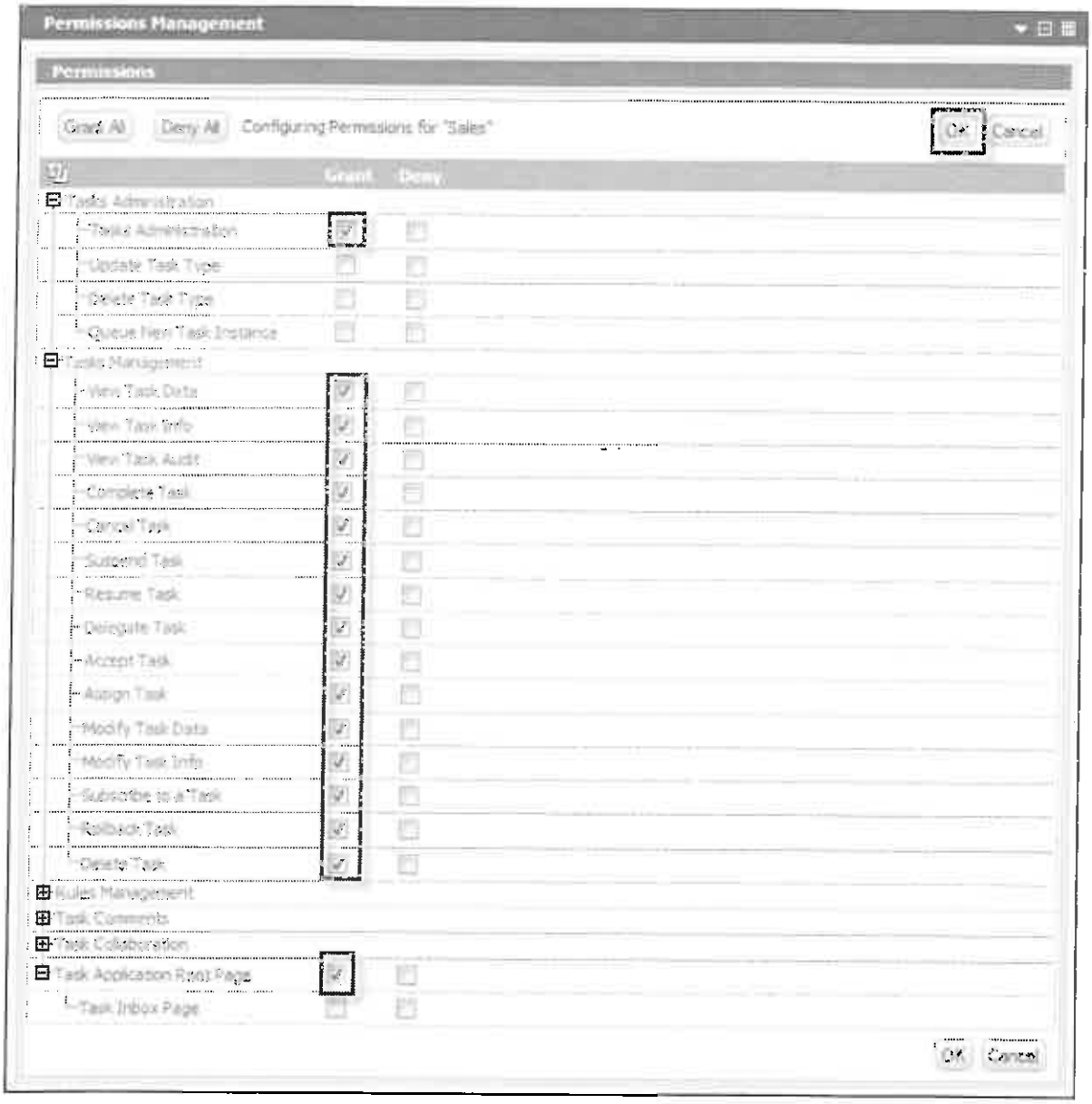
a) Revisit the page **Applications -> Administration -> System-Wide -> Permissions Management.**

b) Select the Resource Type **Tasks** and click **Search.** Select the **ReviewBadOrder** checkbox in the **Found** list and click the arrow right button to move it to the **Selected** list. Click **Next>.**



c) Click **Add Users/Groups/Roles...** and search for **Roles**. Select the **Sales** role and click **Add.**

d) Grant permissions for **Tasks Administration**, all **Tasks Management** activities, and **Task Application Root Page** access:



Click **OK** and **Apply** to save your changes.

7. Double-click the file ***<workshop_dir>*\Exercise16\Resources\Ex16_submit.html.** Use the **Submit** button to start a new process with an invalid Order. If prompted for authentication, provide **Administrator/manage** as the username/password.
8. Logout from My webMethods and login as **Bill/manage** or **Mary/manage.**
9. Navigate to **Applications -> Monitoring -> Business -> Tasks -> Task List Management.**
10. Select the newly created User Task, change the Product Quantity to **9**, and click **Revalidate** to complete the User Task and the corresponding process instance.

Check Your Understanding

1. Why could you not see User Tasks instances assigned to Bill when you originally logged in?

2. What is the difference between privileges for webMethods Applications and privileges for Tasks? One controls access ~~the~~ to webMethods Admin tasks, Two controls access to application tasks.

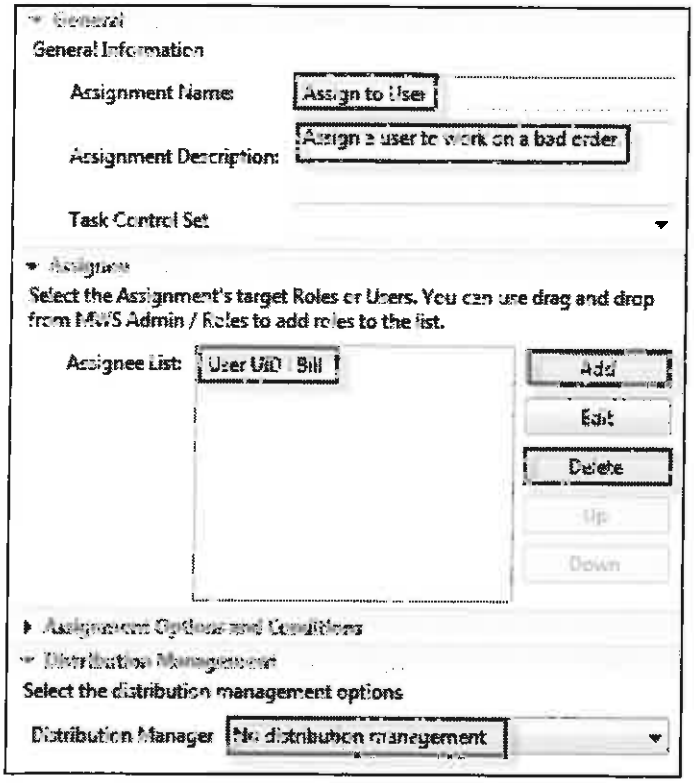
Exercise 17:
User Task Management

Overview

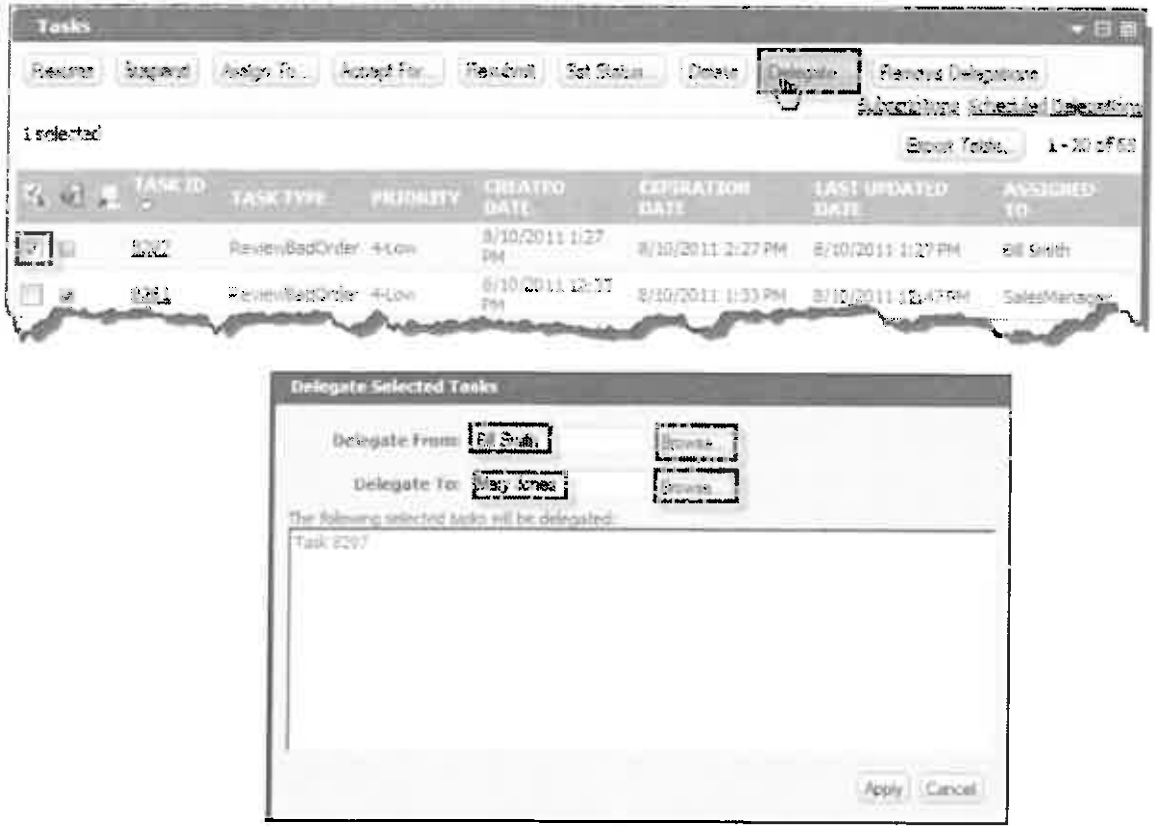
In this exercise, you will perform User Task Management activities on User Task instances, including delegating a User Task instance from one user to another.

Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Software AG Designer and ensure you are in the **UI Development** perspective.
3. Open the User Task **ReviewBadOrder** from the Solutions view.
4. In the Task Editor, open the **Assignments** tab.
5. Select and modify the existing **Assign to Sales Role** assignment.
   a) Set the Distribution Management to **No Distribution management.**
   b) Delete role Sales from the list of assignees. Add user **Bill** to the assignee list.
   Note: Instead of typing, you can drag user Bill from the MWS Admin view into the assignee list.
   c) Change the **Assignment Name** and **Assignment Description** according to the following screen shot:



6. Save your Task changes.
7. Use the Servers view to republish the **SalesDepartment** project to My webMethods Server.
8. Double-click ***<workshop_dir>*\Exercise17\Resources\Ex17_submit.html** and press the **Submit** button to start a new process with an invalid Order. If prompted for authentication use **Administrator/manage** as the username/password.
9. Login to My webMethods as **Bill/manage**.
   a) Open the Task List Management page to ensure a new User Task instance has been queued and is assigned to **Bill Smith**. Delegate the User Task instance from **Bill Smith** to **Mary Jones**. To do so, select the checkbox next to the Task ID, then select the **Delegate...** button in order to begin task delegation definition.



   Click **Apply** to confirm the delegation.

   b) Make sure the **DELEGATIONS** column indicates that the User Task was delegated from **Bill Smith** to Mary Jones:

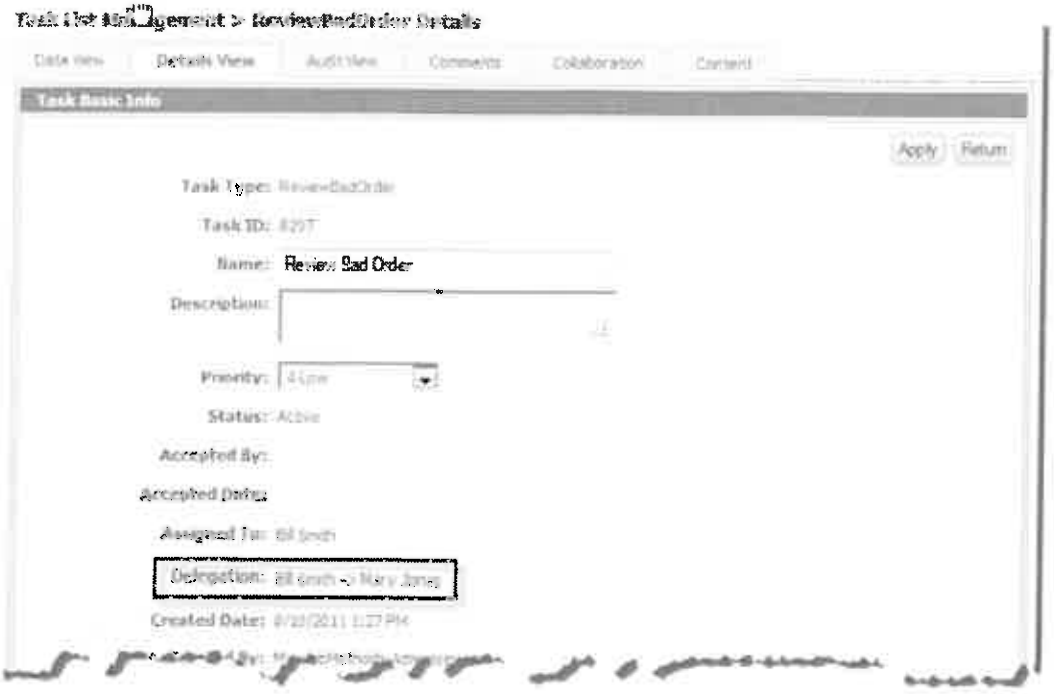
   c) Logout from My webMethods.

10. Login to My webMethods as **Mary/manage.**

a) Navigate to the Task List Management page. If not displayed yet, add the column DELEGATIONS by clicking the button in the header of the User Task list. Select **Properties** and add the column.

b) Inspect at the DELEGATIONS column to ensure the task is delegated to **Mary.**

c) Select the User Task instance and look at the Details View tab also:



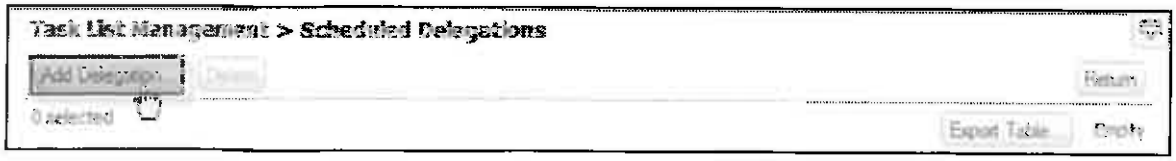
d) Now select the Data View tab, change the **ProductQuantity** from -9 to **9**, and click **Revalidate** to complete the User Task.

e) Logout from My webMethods.

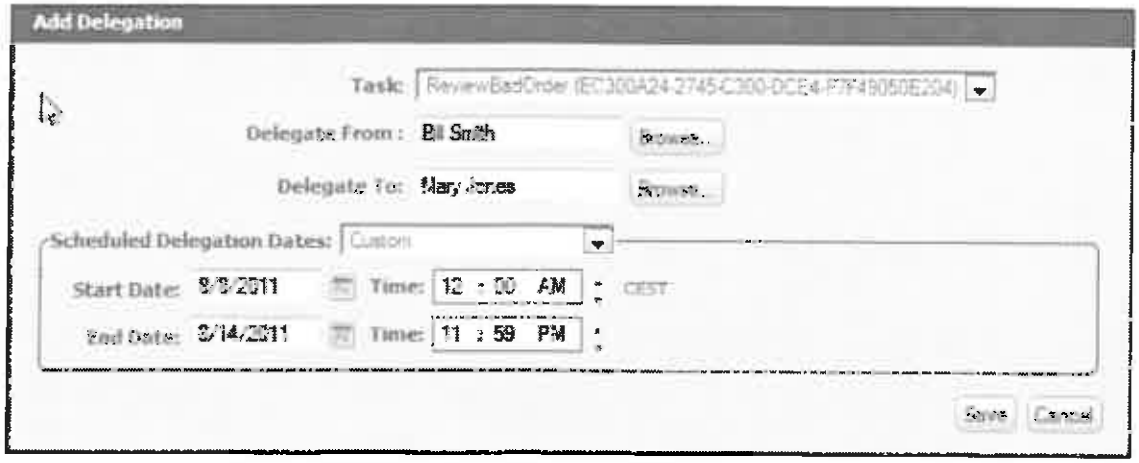
11. Login to My webMethods as **Bill/manage**.

    a) Navigate to **Applications** -> **Monitoring** -> **Business** -> **Tasks** -> **Task List Management** and select **Scheduled Delegations** from the User Task list:

    b) Add a Scheduled Delegation:



    The new Scheduled Delegation is related to the User Task type **ReviewBadOrder** and delegates from **Bill Smith** to **Mary Jones**. It should be valid <***this week***> only:



    c) Save your Scheduled Delegation.

12. Double-click <***workshop_dir***>**\Exercise17\Resources\Ex17_submit.html** and press the **Submit** button to start a new process instance. If prompted for authentication use **Administrator/manage** as the username/password.

13. Login to My webMethods as **Mary/manage**.

    a) Navigate to the Task List Management page and ensure that a new User Task instance is automatically delegated from **Bill** to **Mary**.

    b) Select and open the User Task instance to ensure that **Mary** can access it. To complete the User Task and corresponding process, change the **ProductQuantity** from -9 to **9**, and click **Revalidate**.

Check Your Understanding

1. The User Task above was delegated. Name one difference between delegation and reassignment.

2. What was the difference between the delegation performed in step 9 above, and the delegation in step 11?

   Step 9 delegated a single task.
   Step 11 delegated all tasks this week.

This page intentionally left blank.

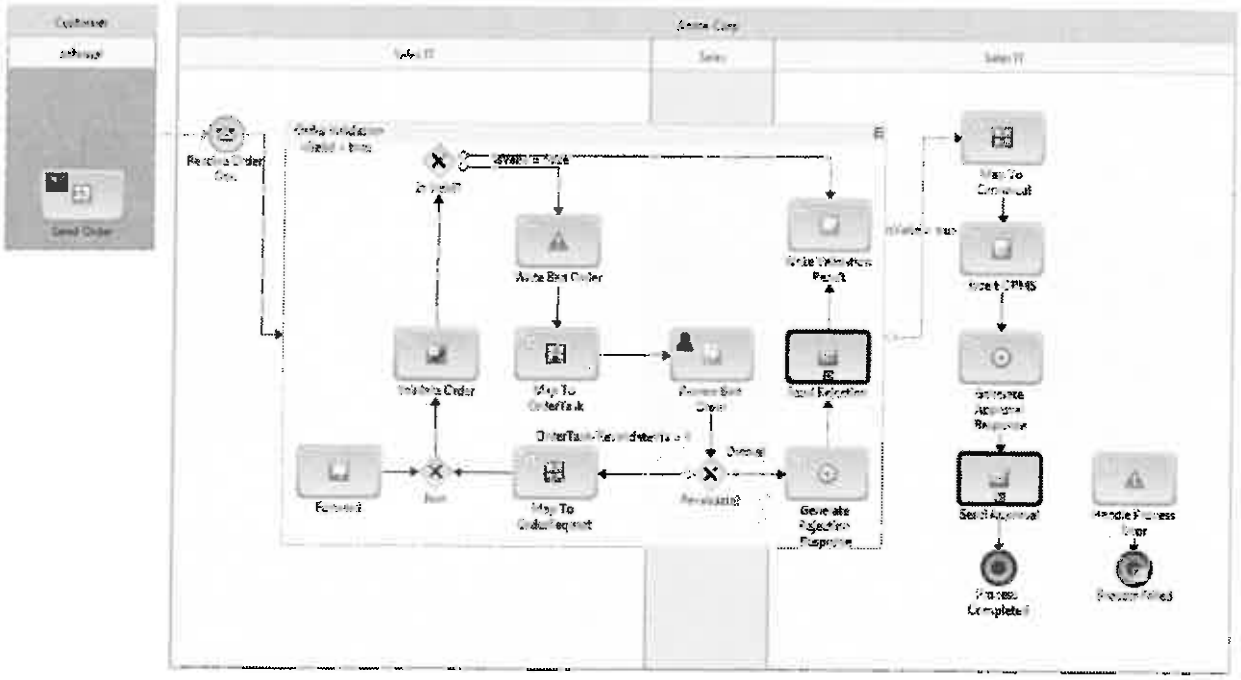
Exercise 18:
Process Error and Timeout Handling

Overview

In this exercise, you will enhance the HandleNewOrder process to capture and handle uncaught process errors and process timeouts at runtime. This will increase the robustness of your business process.

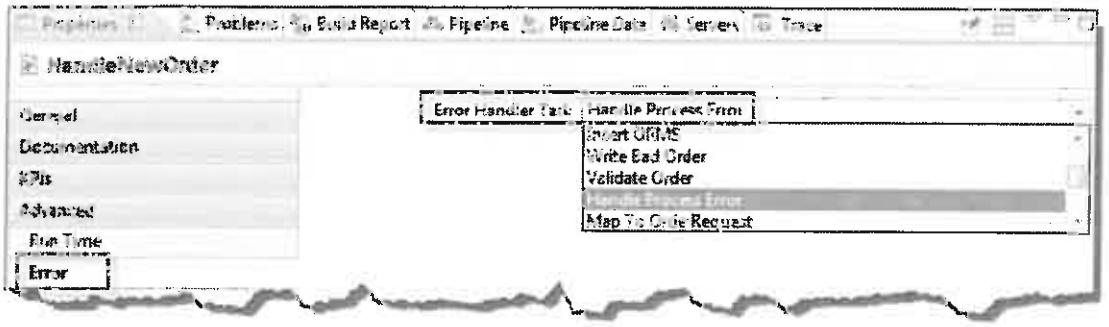
Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Software AG Designer and ensure you are in the **Process Development** perspective.
3. Open the **HandleNewOrder** process.
4. Rename the existing End Terminate Event to **Process Completed**.
5. You will need to resize the Sales IT swimlane on the right side. To do this click on the internal pool Acme Corp, and drag the pool to the right. This will make the swimlane wider.
6. Add a Service Task Activity step named **Handle Process Error,** and a second End Terminate Event named **Process Failed.** Using the Implementation tab, configure Process Failed to set the process status upon termination to **Failed.**
   Change the image of the Service Task Activity and add a transition from **Handle Process Error** to **Process Failed** to correspond to the following image:

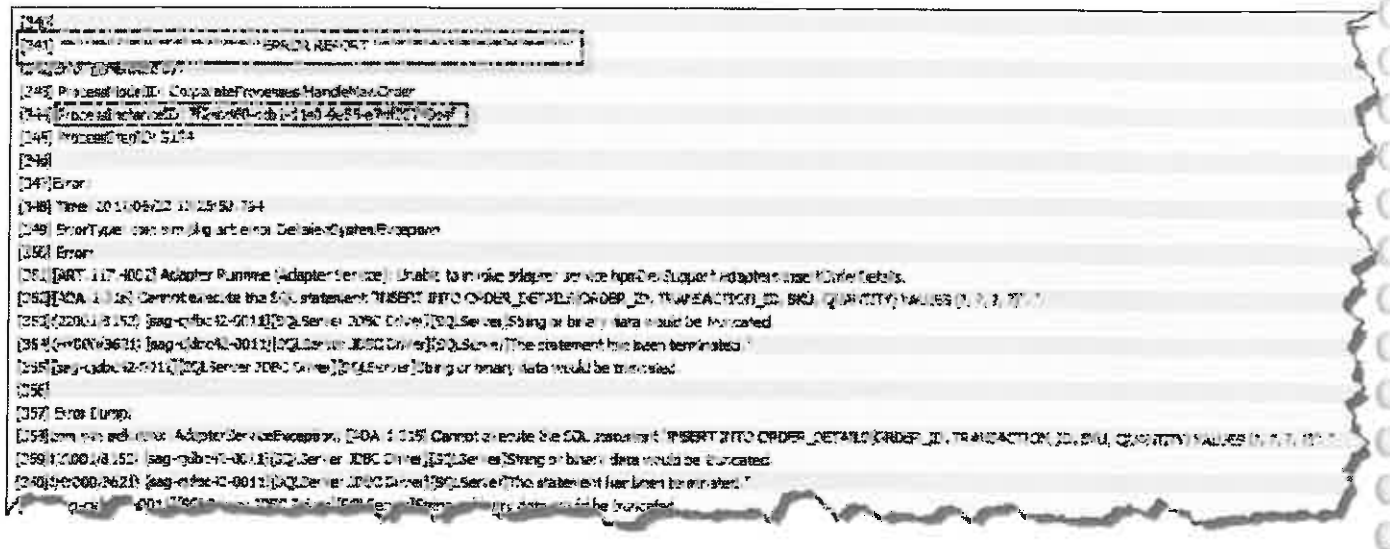


7. Drag the service **bpmDevSupport.utils:handleError** from the Package Navigator view and drop it on the Handle Process Error step. This will automatically configure the step to invoke this service. Check the automatically entered Inputs/Outputs of the **Handle Process Error** step.

8. Click in the whitespace in the process editor to view the process properties. On the **Error** tab of the **HandleNewOrder** process properties, select **Handle Process Error** as Error Handler Task.

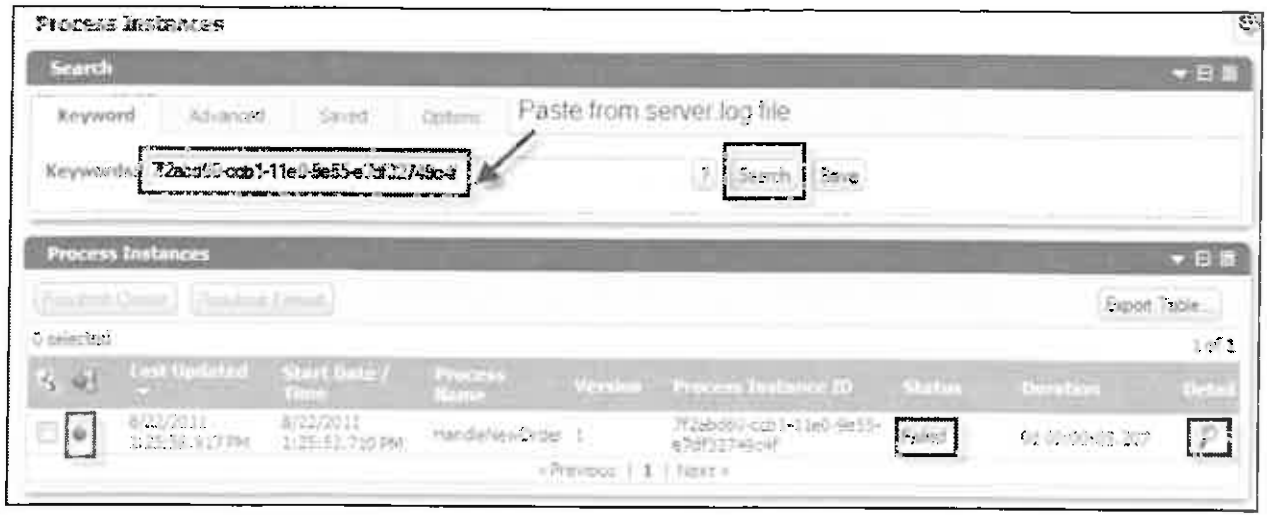


9. Save, build and upload the **HandleNewOrder** process.
10. To start the **HandleNewOrder** process using a browser, double-click ***<workshop_dir>*\Exercise18\Resources\Ex18_submit1.html** and click the **Submit** button. If asked for authentication use **Administrator/manage**. The process will fail due to an order quantity string that is too long.
11. Open the IS **server.log** file by using an editor or the IS Administration console. Search for "*** ERROR REPORT ***" and then copy the value of the **ProcessInstanceID** to the Windows clipboard.

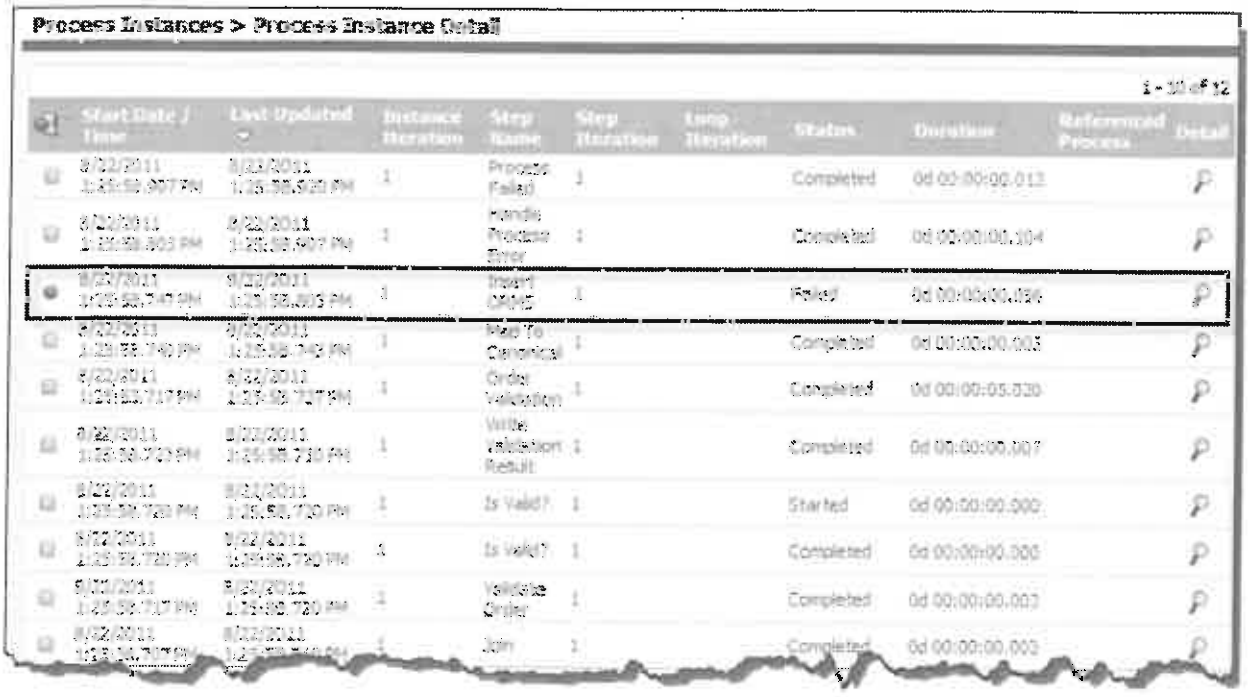


12. Open a browser session and login to My webMethods as **Administrator/manage.**

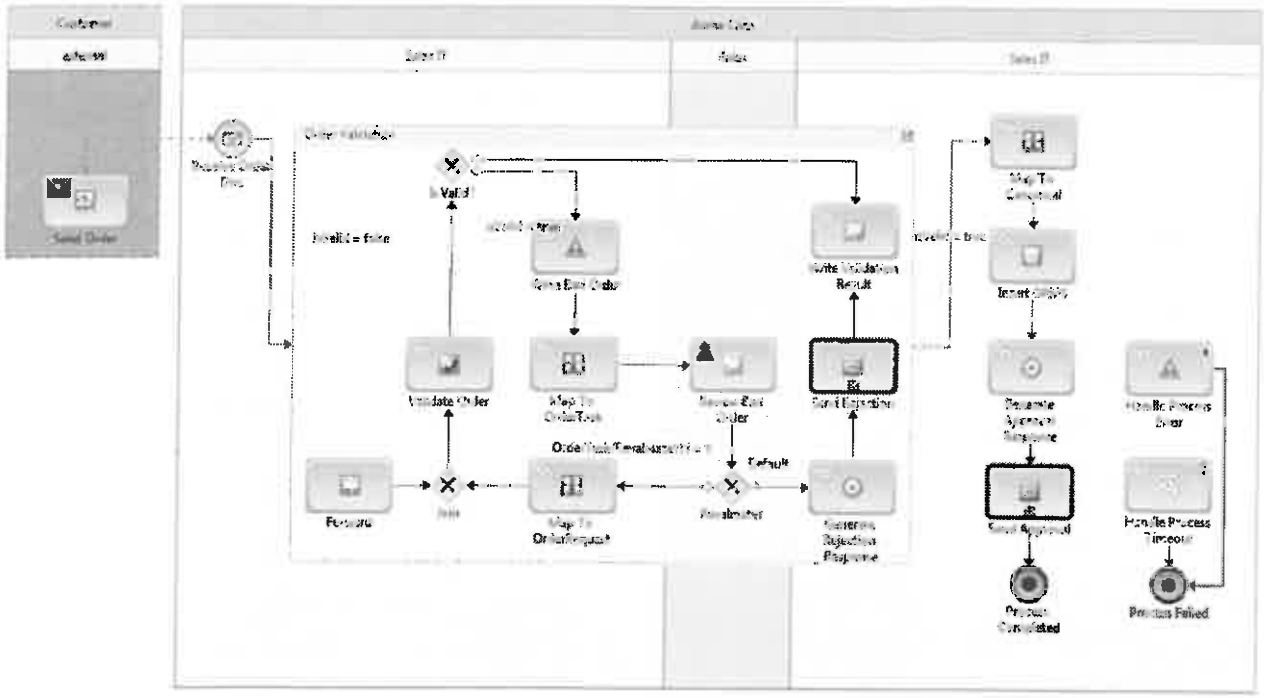
13. Navigate to **Applications** -> **Monitoring** -> **Business** -> **Process Instances**. Look for, or paste and search for the process instance ID of the failed process mentioned in the IS server.log file. Ensure that the process has a **Failed** status.



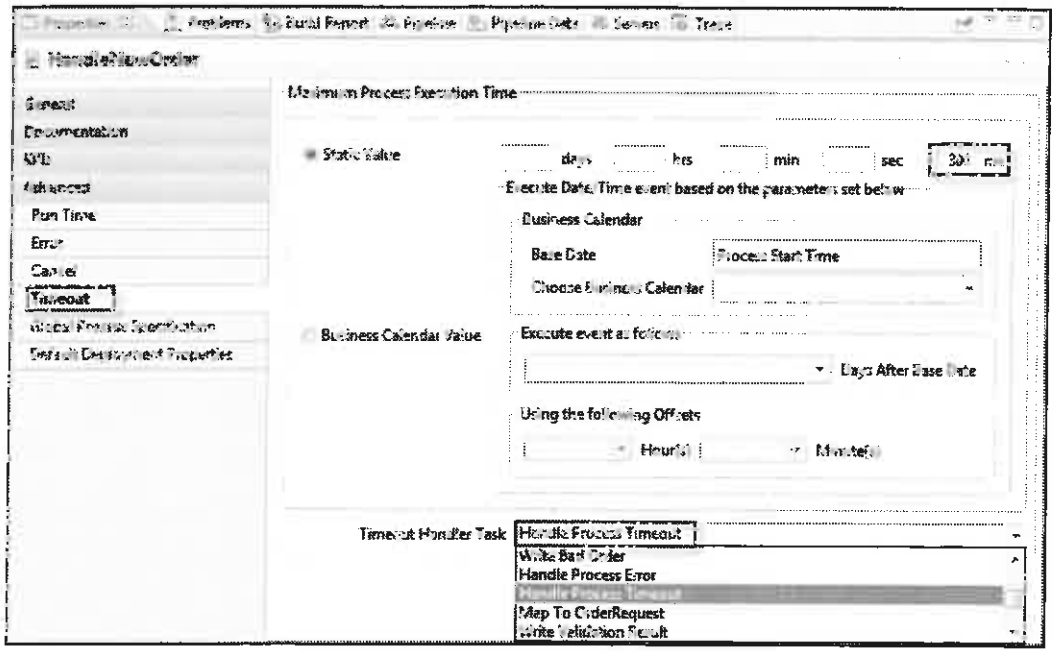
View the details of the failed process instance and determine the step that failed from the Process Diagram section or the Step Summary section:



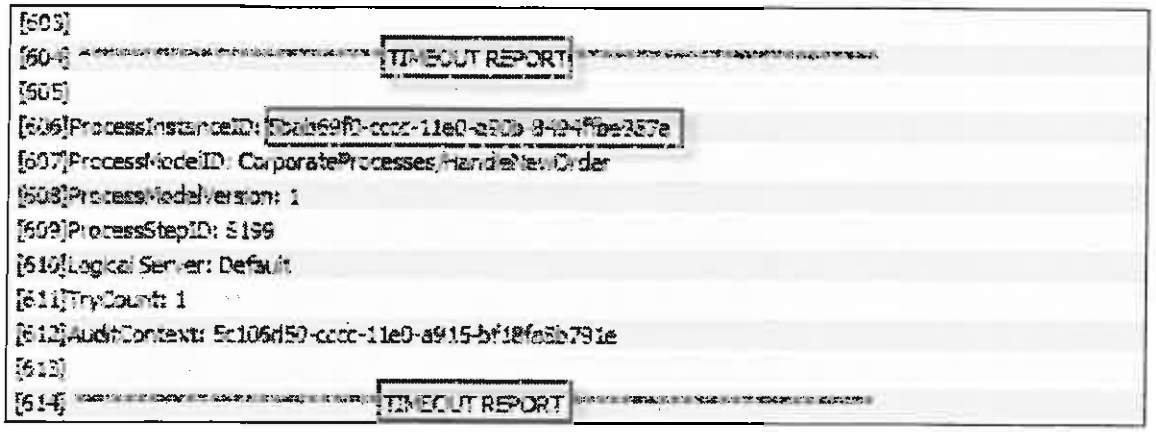
14. In the **HandleNewOrder** process, add another Service Task Activity named **Handle Process Timeout**. Add the step image and add a transition from **Handle Process Timeout** to **Process Failed**. Use the following image as a reference:



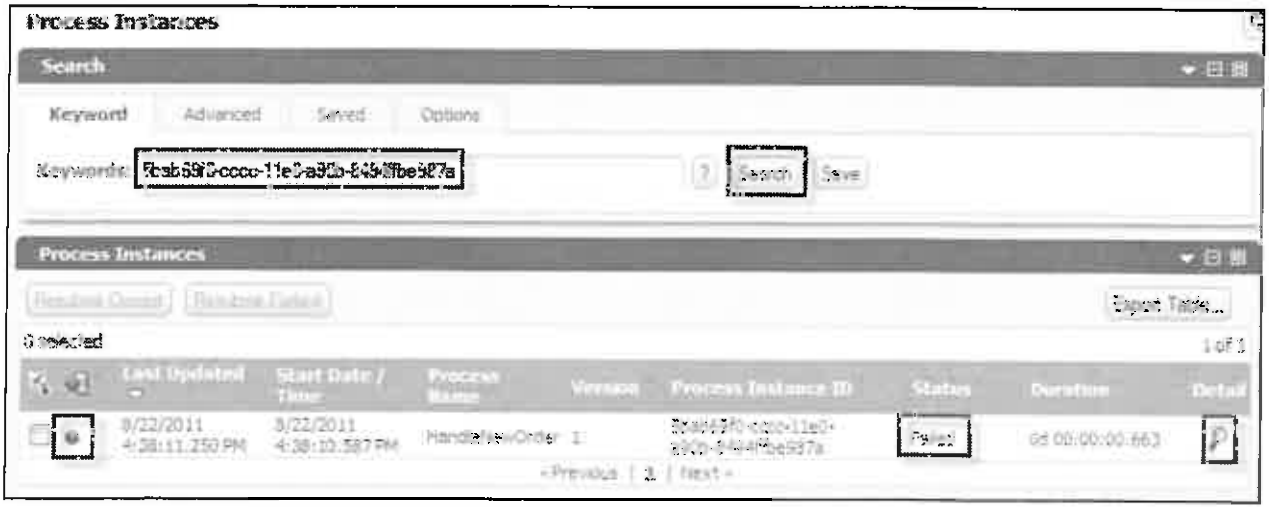
15. Set the **Join Type** of the **Process Failed** event to **Unsynchronized Or**.
16. Drag the service **bpmDevSupport.utils:handleProcessTimeout** from the Package Navigator view and drop it on the **Handle Process Timeout** Activity. This will automatically configure the Activity to invoke this service. Check the automatically entered Inputs/Outputs of the **Handle Timeout** step.
17. In the HandleNewOrder process properties, open the Timeout tab. Set the **Maximum Process Execution Time** to a **Static Value** of **300 milliseconds** and set the **Timeout Handler Task** to **Handle Process Timeout:**



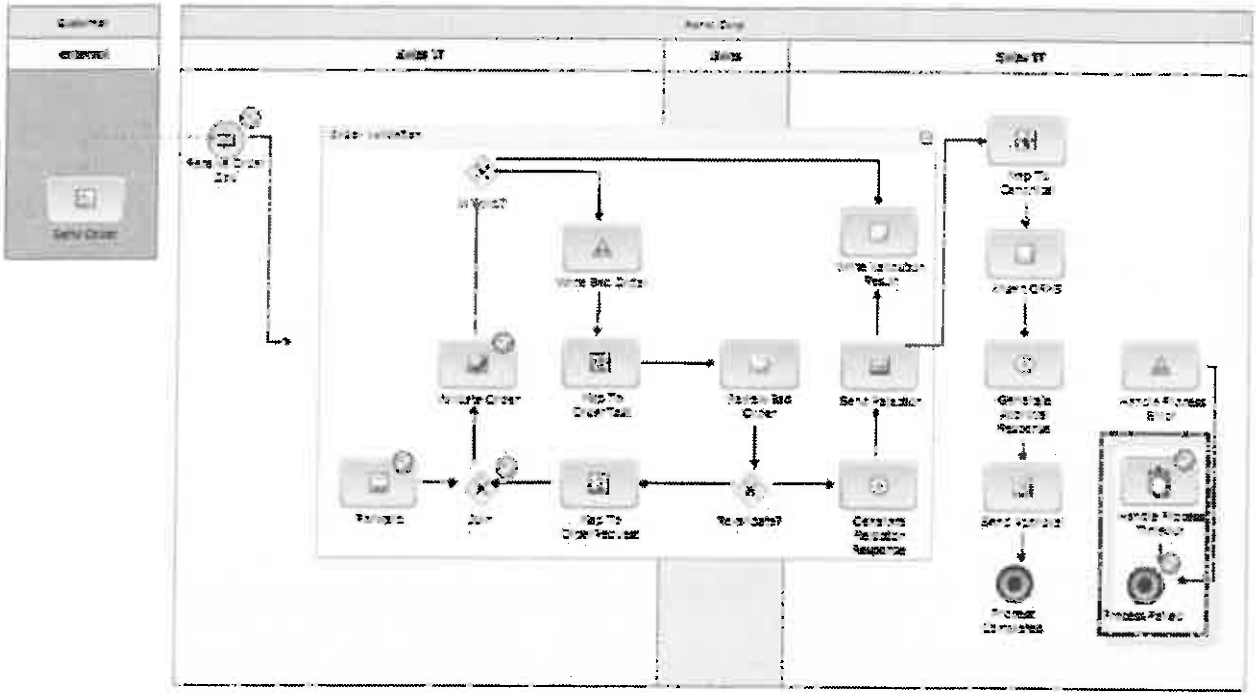
18. Save, build and upload the **HandleNewOrder** process.
19. To start the **HandleNewOrder** process using a browser, double-click ***<workshop_dir>*\Exercise18\Resources\Ex18_submit2.html.**
    Click the **Submit** button. If asked for authentication use **Administrator/manage**.
20. Open the IS **server.log** file by using an editor or the IS Administration console. Search for "*** TIMEOUT REPORT ***" and then copy the value of the **ProcessInstanceID** to the Windows clipboard.



21. Using a browser, login to My webMethods as **Administrator/manage.**
22. Navigate to the **Applications -> Monitoring -> Business -> Process Instances** page. Look for, or paste and search for the process instance ID of the failed process displayed in the IS server.log file. Ensure that the process has a **Failed** status:



23. View the details of the failed process and ensure that the **Handle Process Timeout** step and the **Process Failed** event have been reached:



24. *Housekeeping*: To avoid premature process timeouts in the next exercises, adjust the **HandleNewOrder** process to a proper state:

   a) Switch back to Designer. Open the Properties of your **HandleNewOrder** process. Use the Timeout tab to adjust the **Maximum Process Execution Time** to a more realistic **Static Value** of **3 days**.

   b) Save, build and upload the **HandleNewOrder** process.

Check Your Understanding

1. The Handle Process Error step does not have any incoming transitions. When will it be invoked? When called by the process.

2. Which two documents are automatically sent to the Handle Process Error step?
   perror & ProcessData.

3. Which step caused the process instance to be marked as "Failed"?
   End terminate event "Process Failed".

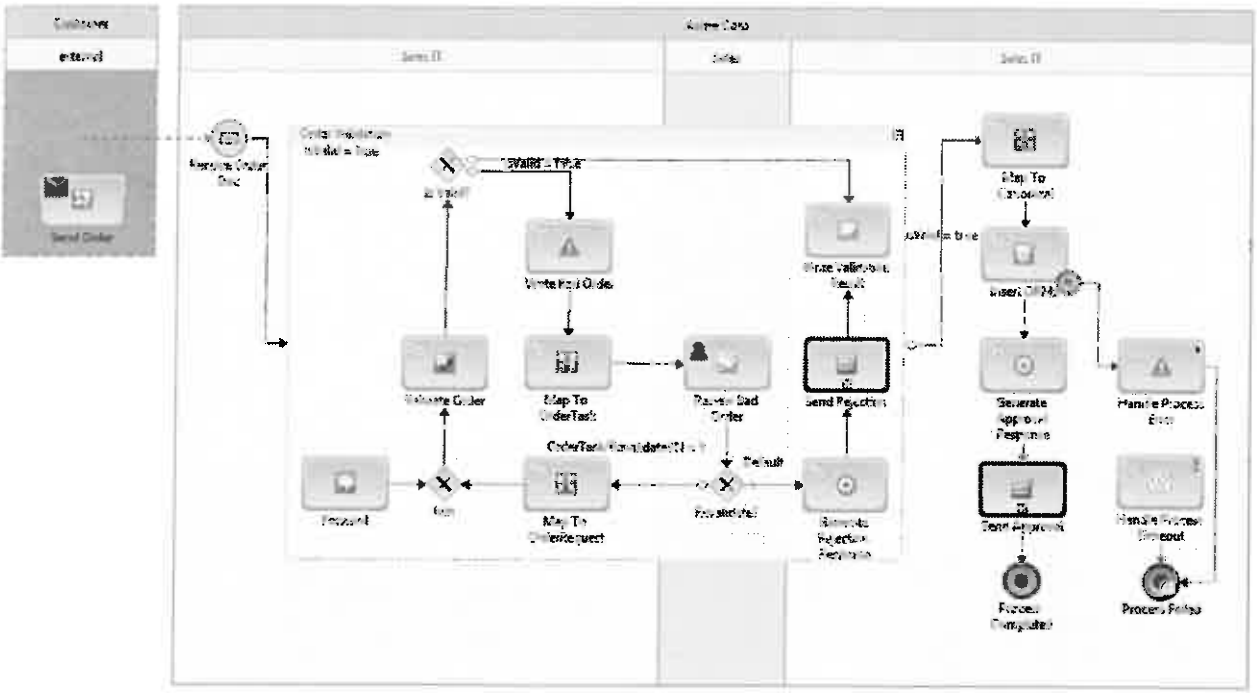
Exercise 19:
Step Error and Timeout Handling

Overview

In this exercise, you will enhance the robustness of the **HandleNewOrder** process by inserting event-driven error and timeout handling on the step level.

Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Software AG Designer and ensure you are in **Process Development** perspective.
3. Open the **HandleNewOrder** process.
4. Right-click step **Insert ORMS** to add a Boundary Error Event to this Service Task Activity. Select the Boundary Event and set its label name to **Catch Database Error**.
5. Add a transition from the Boundary Error Event **Catch Database Error** to the step **Handle Process Error.** Your process should now correspond to the following image:

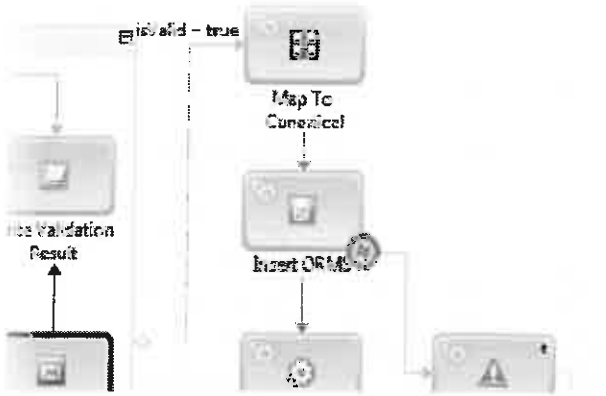


6. Save, build and upload the **HandleNewOrder** process.

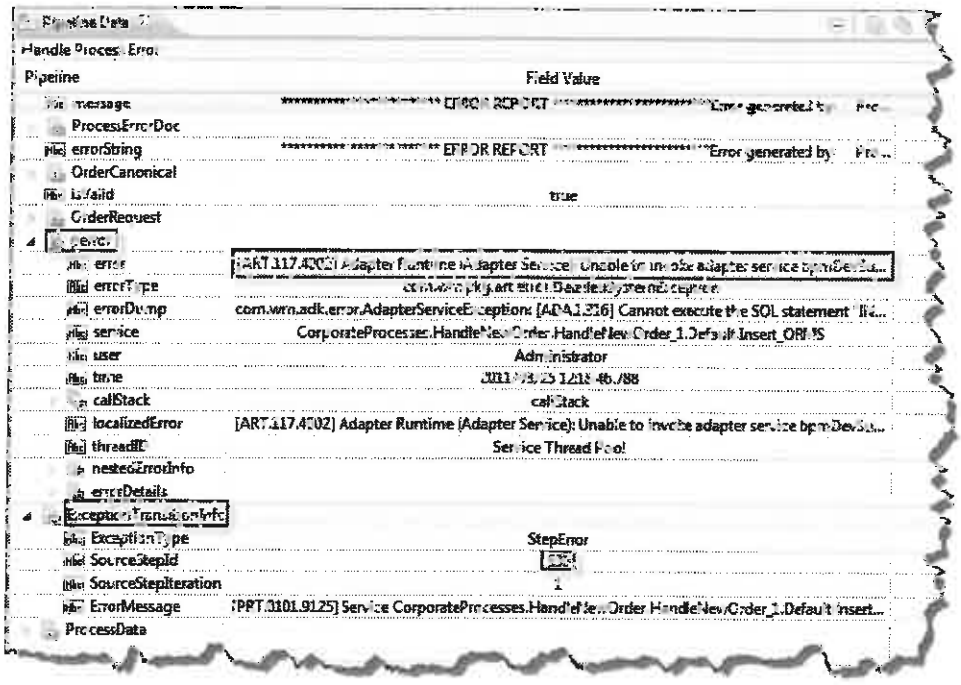
7. Debug your HandleNewOrder process:

   a) Switch to the **Process Debugging** perspective. Click the **Debug Selected Process** icon to start a debugging session for the **HandleNewOrder process.** For the IS input document, load the file ***<workshop_dir>*\Exercise19\Resources\Ex19_input1.txt** as input data.

   b) Step thru the process until you reached the **Insert ORMS** step. The process should fail at the Insert ORMS step due to an order quantity string that is too long. Proceed to the next step by clicking the icon or pressing F6. The Error Boundary Event should get triggered and its outgoing transition to the Handle Process Error should be used:



   c) At the Handle Process Error step, open the Pipeline Data View and inspect the pipeline variables **perror** and **ExceptionTransitionInfo:**

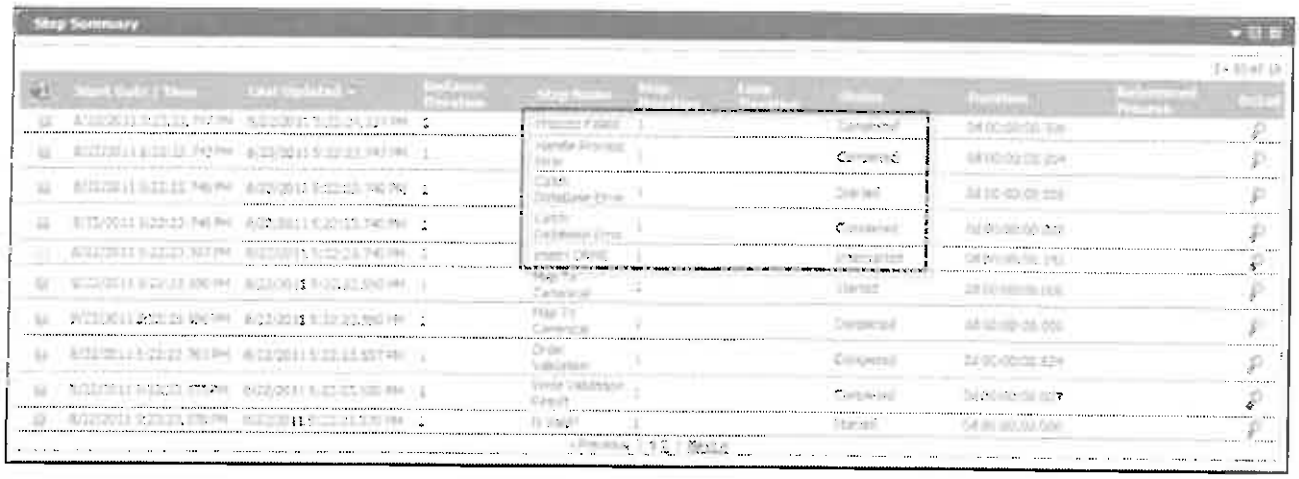


   d) Terminate your debug session by clicking the Run/Resume button or pressing F8.

8. Open the IS **server.log** file by using an editor or the IS Administration console. Search for "*** ERROR REPORT ***" and then copy the value of the **ProcessInstanceID** to the Windows clipboard.

9. Using a browser, login to My webMethods as **Administrator/manage**. Navigate to the **Applications** -> **Monitoring** -> **Business** -> **Process Instances** page. Look for, or paste and search for the process instance ID of the failed process displayed in the IS server.log file. Ensure that the debugged process has a **Failed** status.

10. View the process details to ensure that the step **Insert ORMS** has been interrupted, the Boundary Error Event **Catch Database Error** fired, and that the outgoing transition of the Boundary Error Event has been used:

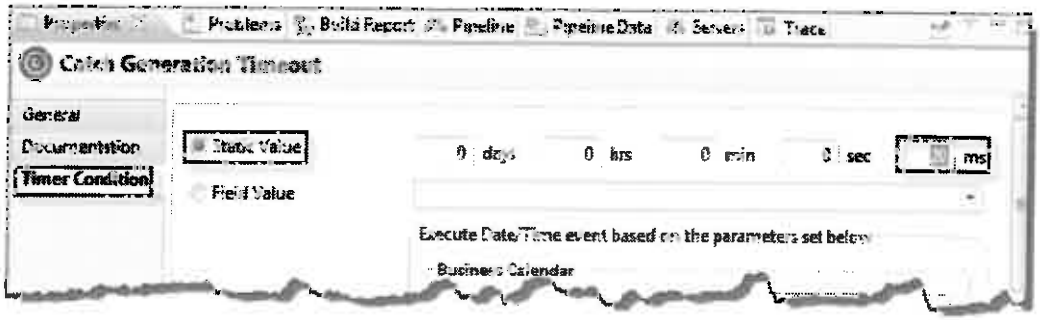


11. Switch back to Designer. In your **HandleNewOrder** process, open the Properties view of the Service Task Activity **Generate Approval Response**. On the Implementation tab, replace the invoked IS Service by the IS Service **bpmDevSupport.utils:generateResponseSleep**.
Note: This service enforces a delay of approximately 180 seconds.

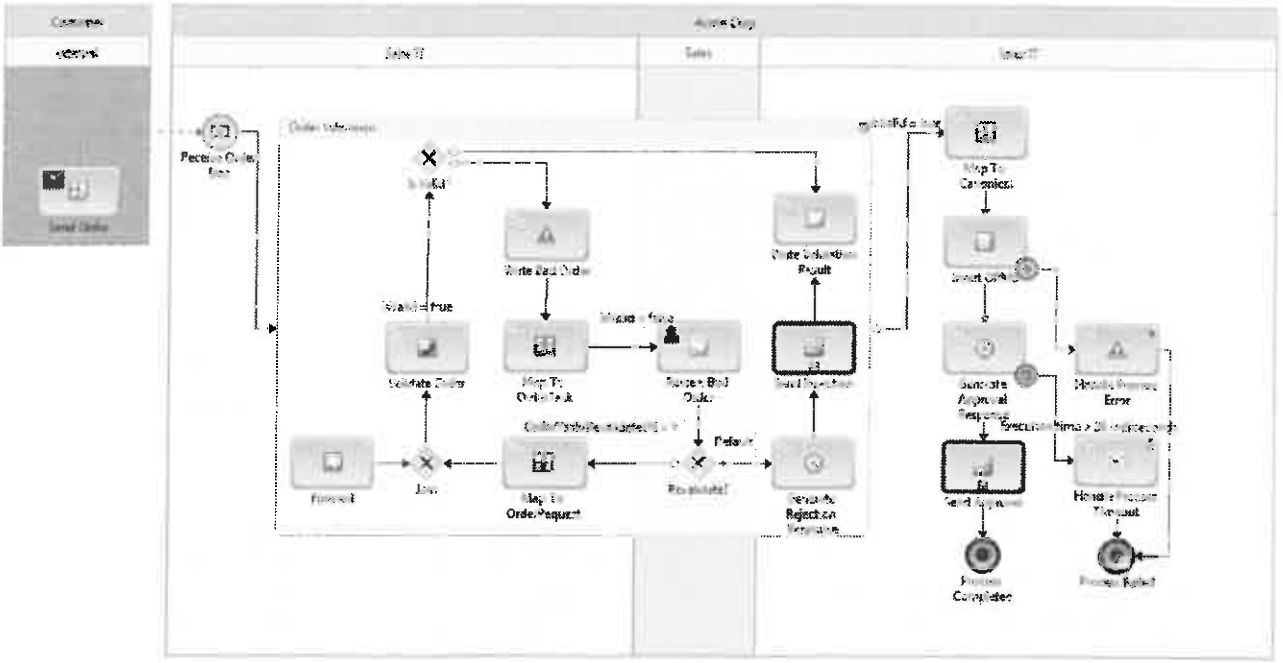
12. Right-click step **Generate Approval Response** to add a Boundary Timer Event to this Service Task Activity. Select the Boundary Event to open its Properties view.

    a) On the General tab, set the label to **Catch Generation Timeout** and make sure the option **Allow Boundary Timer Event to interrupt step** is checked.

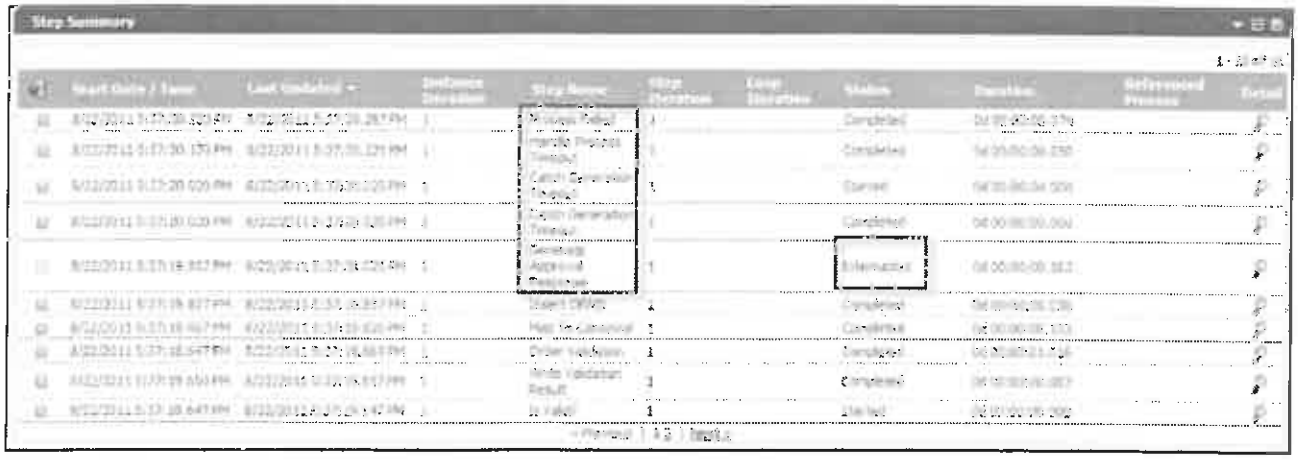
    b) On the Timer Condition tab specify a timeout as a **Static Value** of **20 milliseconds.**



13. Add a transition from the Boundary Timer Event **Catch Generation Timeout** to the step **Handle Process Timeout**. Your process should now correspond to the following image:



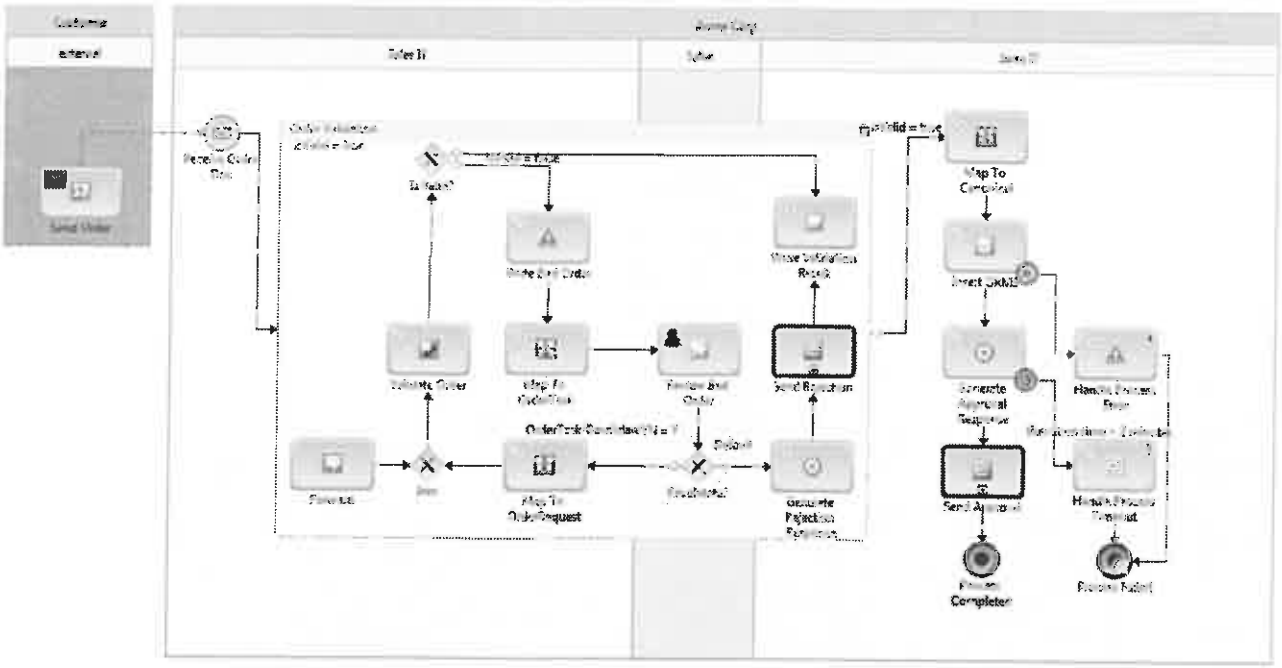
14. Save, build and upload the **HandleNewOrder** process.
15. To start the **HandleNewOrder** process using a browser, double-click ***<workshop_dir>*\Exercise19\Resources\Ex19_submit2.html**.
    Click the **Submit** button. If asked for authentication use **Administrator/manage**.
16. Open the IS **server.log** file by using an editor or the IS Administration console.
    Search for "*** TIMEOUT REPORT ***" and then copy the value of the **ProcessInstanceID** to the Windows clipboard.
17. Use a browser to login to My webMethods as **Administrator/manage**. Navigate to the **Applications -> Monitoring -> Business -> Process Instances** page. Look for, or paste and search for the process instance ID of the failed process displayed in the IS **server.log** file. Ensure that the process has a **Failed** status.
18. View the process details to ensure that the **Generate Approval Response** step has an **Interrupted** status and the **Catch Generation Timeout** Boundary Event fired:



19. *Houskeeping*: To avoid premature timeouts in the next exercises, adjust the **HandleNewOrder** process to a proper state:

    a) Reset the **Generate Approval Response** Service Task Activity to invoke the original IS Service **bpmDevSupport.utils:generateResponse.** Refresh Inputs/outputs.

    b) Change the Timeout value of the **Catch Generation Timeout** Boundary Timer Event to a value of **2 minutes.** The **HandleNewOrder** process should now look like this:



    c) Save, build and upload the **HandleNewOrder** process.

Check Your Understanding

1. Can the Handle Process Error step catch errors thrown from steps in the subprocess?

2. Can you change the Boundary Error Event Catch Database Error at the Service Task Activity Insert ORMS to be non-interrupting? No

3. Briefly describe the impact if you change the Boundary Timer Event to be non-interrupting.

This page intentionally left blank.

Exercise 20a:
webMethods Business Rules

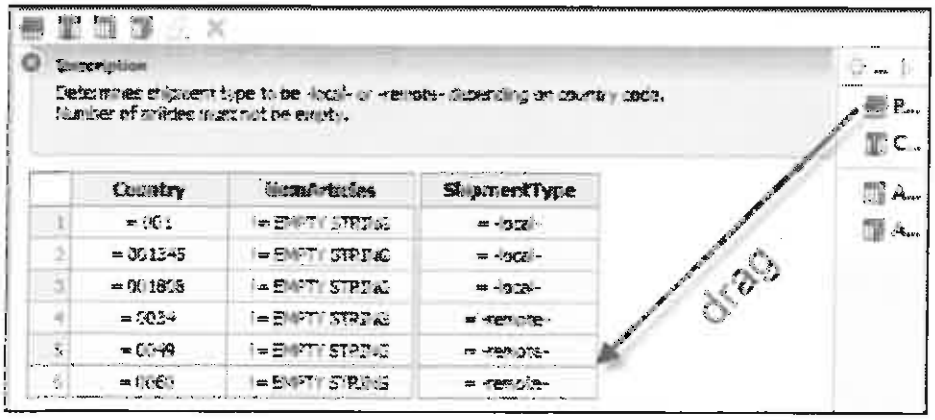
Overview

In this exercise, you will first modify a provided webMethods Business Rules project using Designer. The business rule in the project will return a value of "-local-" or "-remote-", depending on the global country code that is passed in as input. You will deploy the rules to the Integration Server from Designer.
To use the Decision Table, you will enhance the existing process NotifyCustomer. The process will invoke the Decision Table and display the returned value in the notification message. Finally, you will deploy a Rule Management Console (RMC) to your MWS to allow business users to perform instant rule modifications.

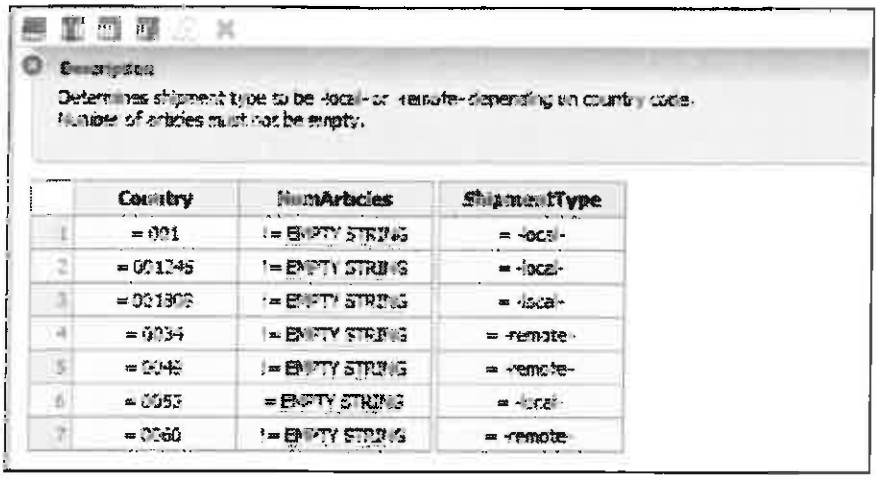
Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Designer and switch to the Rules Development perspective and open the Solutions view.
3. Import a webMethods Business Rules project containing a Decision Table:
   a) Select **File -> Import**
   b) Choose **General -> Existing Projects into Workspace**. Then click **Next**.
   c) Click **Select archive file** and browse for the provided zip file:
   <*workshop_dir*>\Exercise20a\Resources\CorporateProcessesRules.zip
   d) Select the zip file and click **Open**.
   e) Click **Finish**.
   As a result you should see a project named **CorporateProcessesRules** contained in the Rules folder in the Solutions view.
4. Switch to the Rules Explorer view and double-click to open the existing Decision Table **DetermineShipmentDestination** in the Rule Editor.
   Note:
   The Decision Table has an In/Out parameter named **ShipmentDoc** of Data Model type **ShipmentDoc**. This Data Model refers to the existing IS document **bpmDevSupport.docs.rules.ShipmentDoc**.
   It is the task of the rules contained in the Decision Table instance to investigate the value of the incoming **Country** field - as contained in the input parameter **ShipmentDoc** - and to assign a string value in a field called **ShipmentType** which is also part of the ShipmentDoc. As Acme is located in the United States, country codes of 001 (US), 001345 (Cayman Islands), and 001808 (Hawaii) should be handled as **-local-** shipments, 0034 (Spain), 0049 (Germany), and 0060 (Malaysia) should be **-remote-**. Additionally each rule ensures, that articles are ordered by verifying that **NumArticles** does not contain an empty string.

5. In the Rule Editor, insert a new rule after row five by dragging a Rule from the Palette and dropping it in the Decision Table.

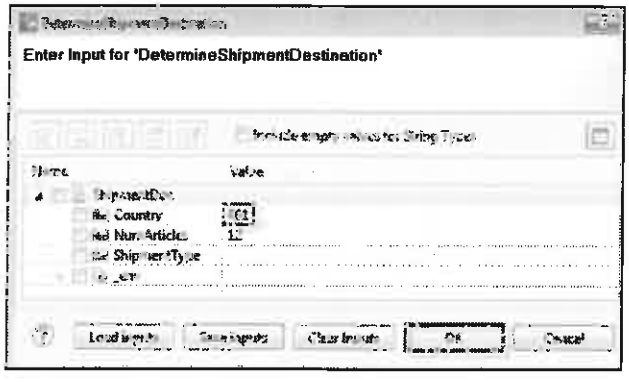


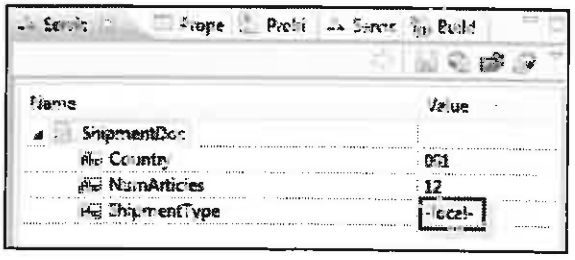
6. In the inserted rule, add a condition = **0052** (Mexico) for Country, != **Empty String** for NumArticles, and the assignment = **-local-** for ShipmentType.
Note: To customize the condition for NumArticles, left-click the cell and click the pencil icon to open the extended cell editor. Empty String is available at the Constants tab.



7. Save your definitions.

8. To test your Decision Table **DetermineShipmentDestination** in Designer, right-click your Decision Table in the Rules Explorer view and select Run As -> Run Decision Table. You can load the file ***<workshop_dir>*\Exercise20a\Resources\ Ex20a_ruleInput1.txt** as input.





Repeat the test with the input file:
***<workshop_dir>*\Exercise20a\Resources\ Ex20a_ruleInput2.txt**

9. To invoke your Decision Table from a BPM process, the webMethods Business Rule project has to be deployed to an Integration Server hosting the Rule Engine.
   To do so:
   a) In Designer, select **File -> Export**
   b) Choose **Software AG -> Rule Project to Integration Server runtime**. Then click **Next**.
   c) Select Rule project **CorporateProcessesRules** and Integration Server **Default**. Then click **Finish**.

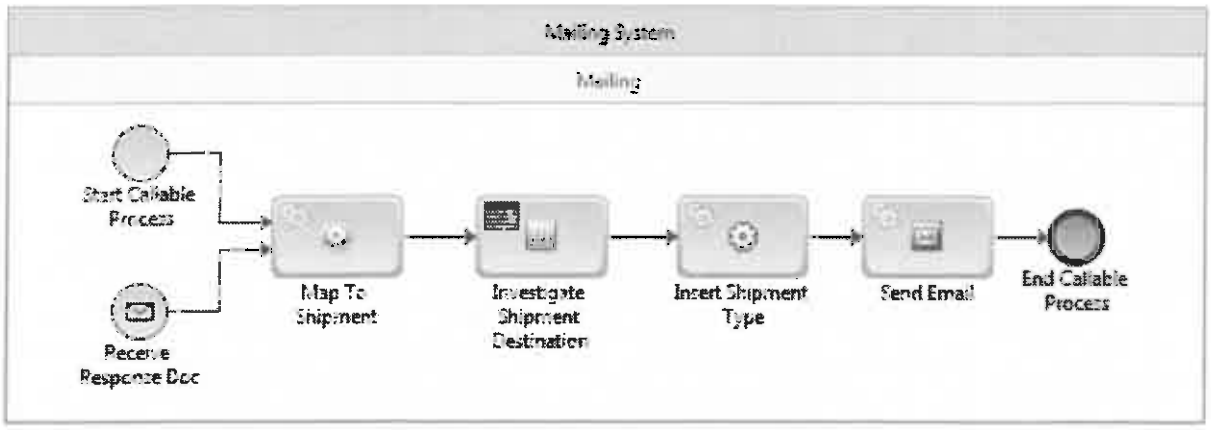
   Note: When the webMethods Rule project gets deployed to the Integration Server, it will be stored as a jar file in the folder:
   <install root>\IntegrationServer\packages\WmBusinessRules\projects
   Therefore, you should now see the file **CorporateProcessesRules.jar** in this folder.

10. Switch to the Process Development Perspective.

11. Open the existing process model **NotifyCustomer** as contained in your **CorporateProcesses** process project and perform the following modifications:

    a) You will need to resize the Mailing swimlane on the right side. To do this click on the internal pool Mailing System, and drag the pool to the right. This will make the swimlane wider.

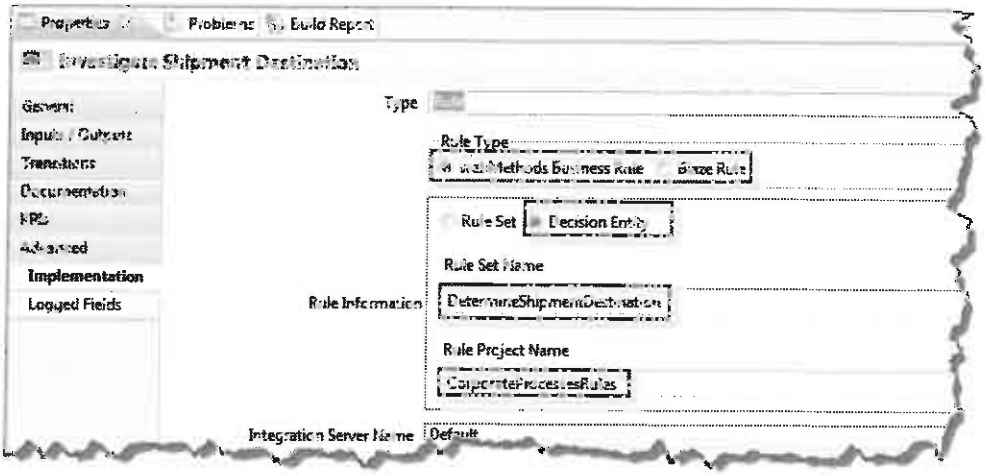
    b) Add two Service Task Activities of type **IS Service** named **Map To Shipment** and **Insert Shipment Type,** and a Rule Task Activity named **Investigate Shipment Destination.** Adjust step images and add transitions to correspond with the following image:



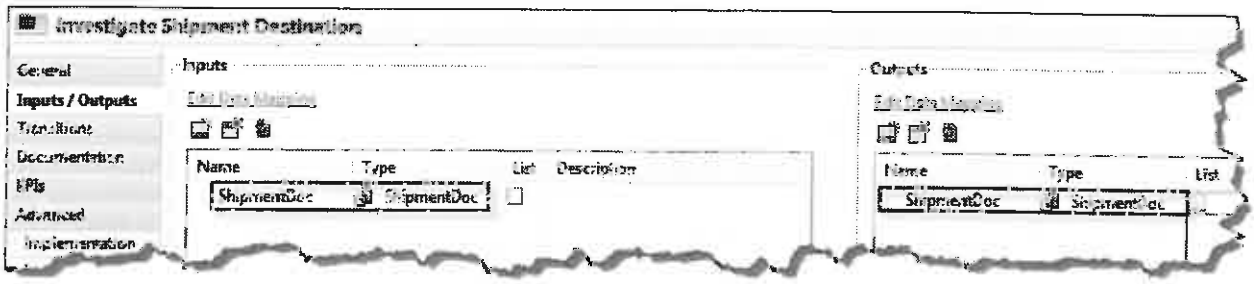
    c) Use drag and drop from the Package Navigator view to assign **bpmDevSupport.maps:OrderResponseToShipment** as the implementing IS service of step **Map To Shipment**. Ensure that the Join Type at the Map To Shipment step is of type **Unsynchronized Or**. Also check that the step's Inputs/Outputs have been set accordingly.

d) Configure the properties of the Rule Task Activity **Investigate Shipment Destination** like this:

- On the Implementation tab, set Rule Type as **webMethods Business Rules.** Select subtype **Decision Entity** to invoke a Decision Table.
  Browse for the Decision Table **DetermineShipmentDestination** contained in your Rule project **CorporateProcessesRules.**
  The Implementation properties should now look like this:

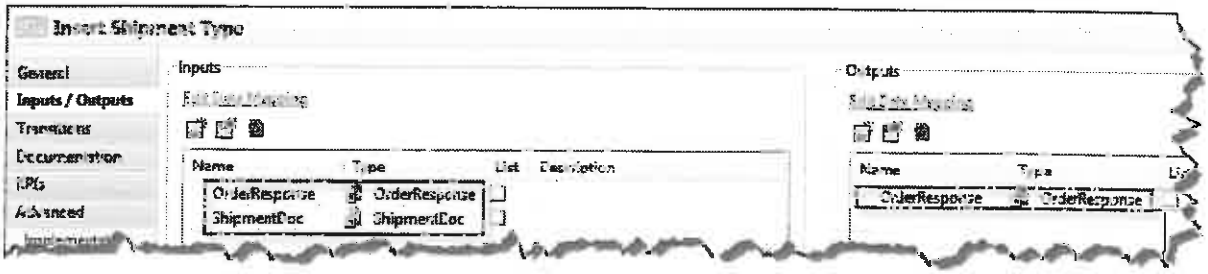


- Ensure the inputs and outputs of your Rule Task Activity contain **ShipmentDoc:**



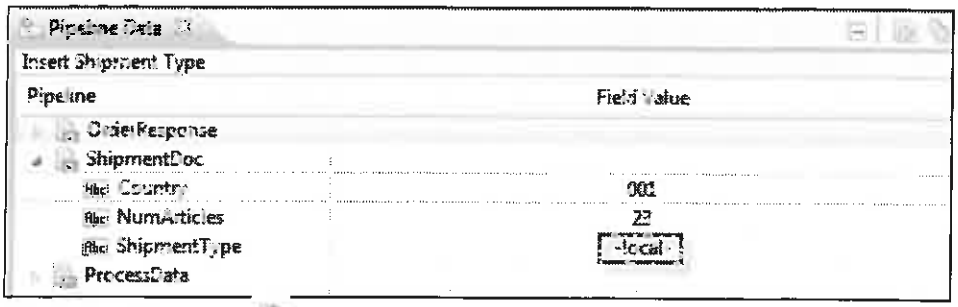
- Rename your Rule Task Activity to **Investigate Shipment Destination.**

e) Assign the IS service **bpmDevSupport.utils:insertShipmentType** to the step **Insert Shipment Type.** Ensure the input of the step contains **OrderResponse** and **ShipmentDoc,** and the output contains **OrderResponse:**



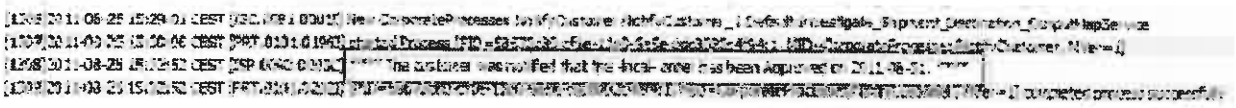
12. Save, build and upload your **NotifyCustomer** process.

13. Switch to the **Process Debugging** perspective. Click the Debug Selected Process icon to start a debugging session. For the IS input document, load the file ***<workshop_dir>*\Exercise20a\Resources\Ex20a_input1.txt.**
Note the provided Global Country Code is **001**. Step thru the process. Use the Pipeline Data view to ensure the step data of **Insert Shipment Type** contains the expected rule result:



Click Run/Resume to complete your process in Debugger.

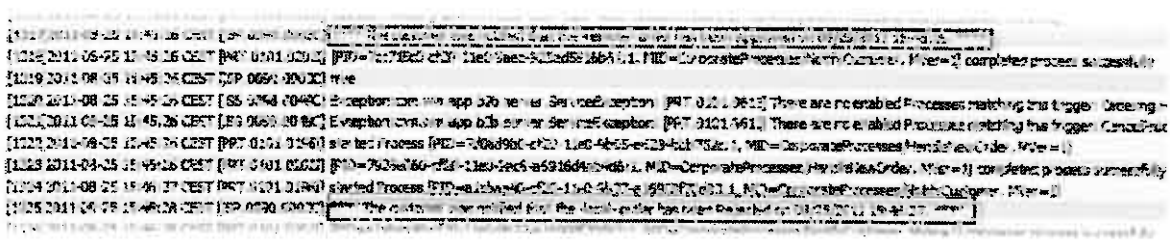
14. Open the Integration Server's server.log file by using an editor or the IS Administration console. If using an editor the server.log file can be found in the folder C:\SoftwareAG\IntegrationServer\logs. You should see an enhanced message like "***** The customer was notified that the -**local**- order has..." written by the **Send Email** step:



15. Test your entire **HandleNewOrder** process.
Note: Remember that **HandleNewOrder** invokes **NotifyCustomer** as twice, once as a BPMN Callable Process, second as a webMethods Referenced Process.
Test both cases by double-clicking the provided HTML forms:
- ***<workshop_dir>*\Exercise20a\Resources\Ex20a_submit_DE_validOrder.html**
- ***<workshop_dir>*\Exercise20a\Resources\Ex20a_submit_US_invalidOrder.html.**
In both cases click the **Submit** button to start the process. If asked for authentication use **Administrator/manage**.

16. Login to My webMethods as **Administrator/manage**. Navigate to the **Applications** -> **Monitoring** -> **Business** -> **Process Instances** page. Ensure that the first process instance completed successfully.
Because of invalid data, the second process instance should wait for a User Task first. Navigate to the **Applications** -> **Monitoring** -> **Business** -> **Tasks** -> **Task List Management** page. Select and open the User Task instance and **Abort** the User Task instance. This should complete the process also.

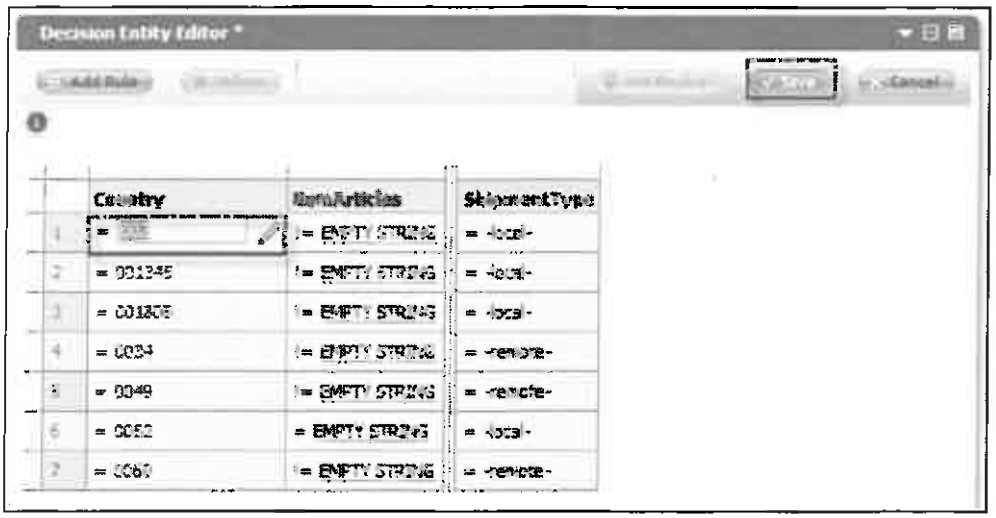
17. Inspect the IS Server log to check that the first order is flagged as an accepted -**remote**- order. The second order should be flagged as a rejected -**local**- order in the server.log file.



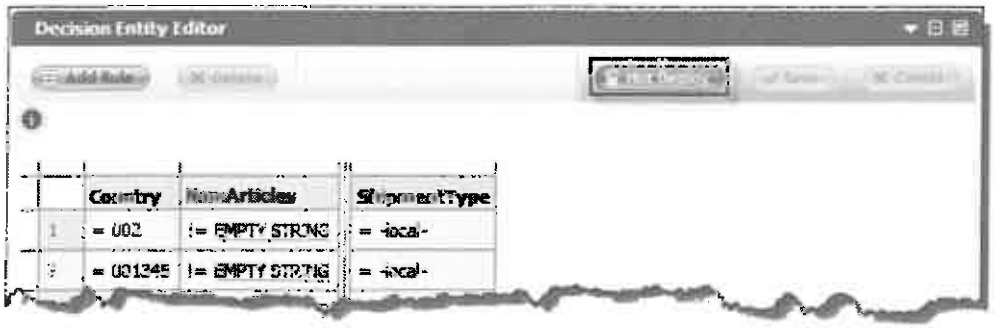
18. Deploy a Rule Management Console that corresponds to your webMethods Business Rules project to MWS. To do so:
    a) In Designer, select **File -> Export**
    b) Choose **Software AG -> Rule Project to My webMethods Server repository.** Click **Next.**
    c) Select Rule project **CorporateProcessesRules** and My webMethods Server **MWS Content Repository.** Then click **Finish.**

19. After successful deployment, use a browser to login to My webMethods as **Administrator/manage.** Navigate to **Applications -> Administration -> Business -> webMethods Business Rules -> Welcome.**
    On the Welcome page, click **Update Entries in Navigation Pane** to add your deployed RMC to the Navigation bar. In the Navigation bar, click the Refresh Navigation Tree icon.

20. Navigate to **Applications -> Administration -> Business -> webMethods Business Rules -> CorporateProcessesRules -> Decision Tables.** On the appearing page, click Decision Table **DetermineShipmentDestination** to open it in the Decision Entity Editor. Replace the Country value 001 by **002** and click **Save** to commit your modification:



21. Using the RMC, perform a Hot Deploy of your modified Business Rule project to your Integration Server:

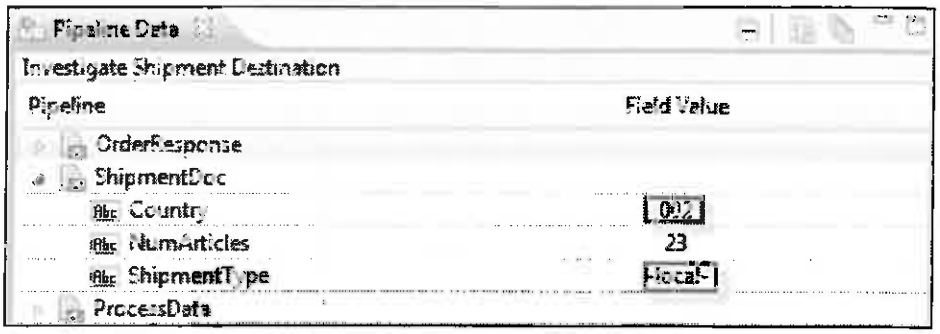


On the appearing panel, confirm that you wish to deploy.

22. Double-check the modification done in the RMC by debugging the **NotifyCustomer** process again:

    a) Go back to Designer, if necessary switch to the **Process Debugging** perspective.

    b) Ensure that your process **NotifyCustomer** is opened in Designer and click the Debug Selected Process icon to start a debugging session. For the IS input document, load the file ***<workshop_dir>*\Exercise20a\Resources\Ex20a_input2.txt**.
    Note the provided Global Country Code is **002**. Step thru the process. Use the Pipeline Data view to ensure the step data of **Insert Shipment Type** contains the expected rule result:



    c) Click Run/Resume to complete your process in Debugger.

Check Your Understanding

1. What happens when you export a webMethods Business Rules project to an Integration Server?

2. Is it always necessary to deploy a Rules Management Console to MWS? Only if you want business users to modify rules.

This page intentionally left blank.

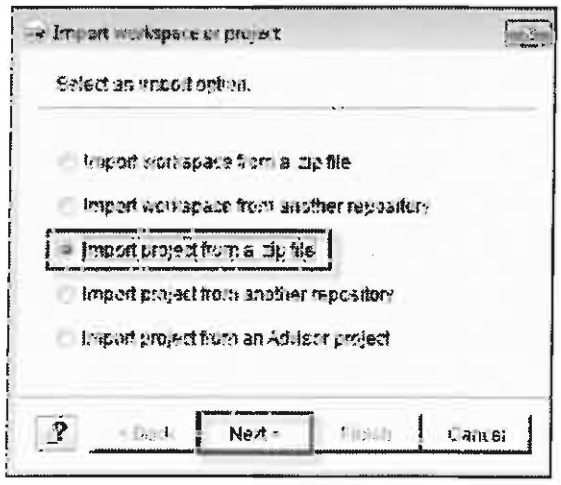
Exercise 20b:
Blaze Business Rules

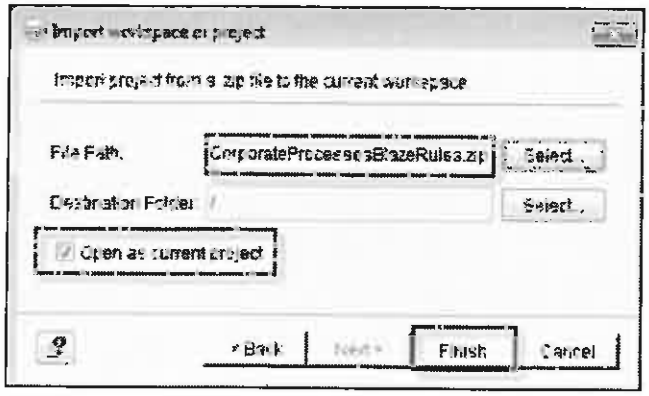
Overview

In this exercise, you will first modify a provided Blaze Business Rule project using Blaze Advisor. The contained business rules will determine and return a value of "-local-" or "-remote-", depending on the incoming global country code. Using Blaze Advisor, the rules will be deployed to your Integration Server and stored as a Blaze Business Rule service.
To use the Blaze Business Rule service, you will enhance the existing process NotifyCustomer. The process will invoke the Blaze Business Rule service and display the returned value in the notification message.
Finally, you will deploy a Blaze Rule Maintenance Application (RMA) to your MWS to allow business users to perform instant rule modifications.

Steps

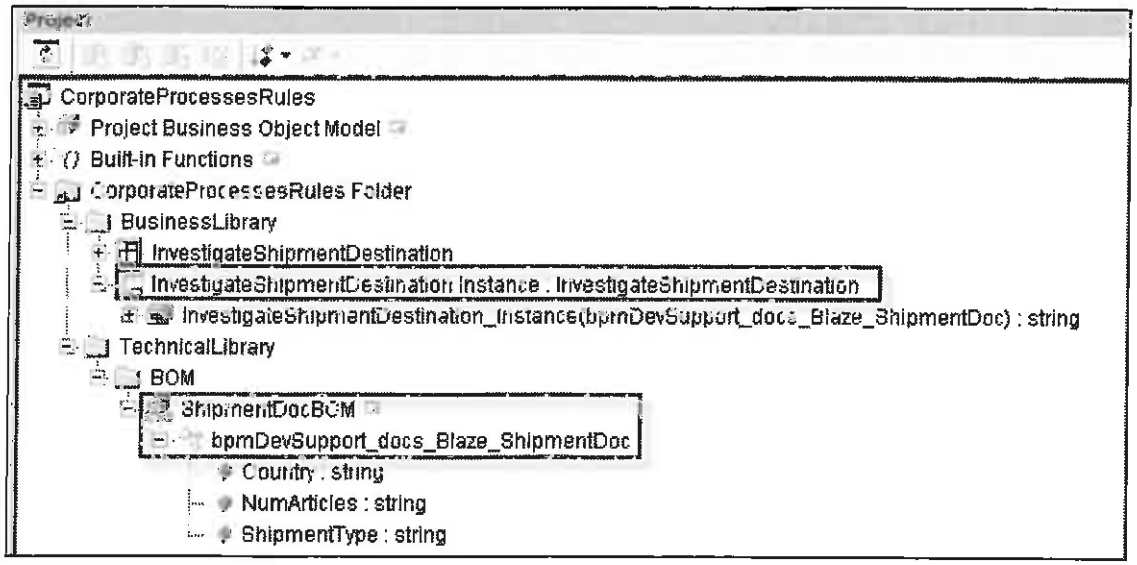
1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Start Blaze Advisor from **Start -> All Programs -> Software AG -> Tools -> Blaze Advisor.**
3. In Blaze Advisor, first choose **File -> Connect to Repository...** and select the existing **BlazeRepository**. If asked, connect as user **Administrator** with password **manage.**
4. Choose **File -> Import** and select the import option **Import project from a .zip file** to import the Blaze project from:
   ***<workshop_dir>*\Exercise20b\Resources\CorporateProcessesBlazeRules.zip**
   Check to open the project as current project.



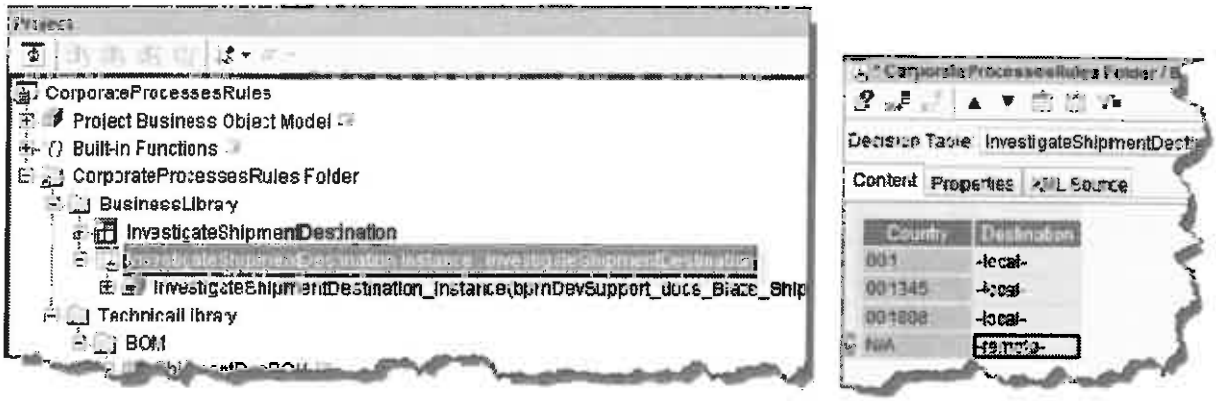


If you are asked for a username/password to connect to the Integration Server, enter **Administrator/manage.**

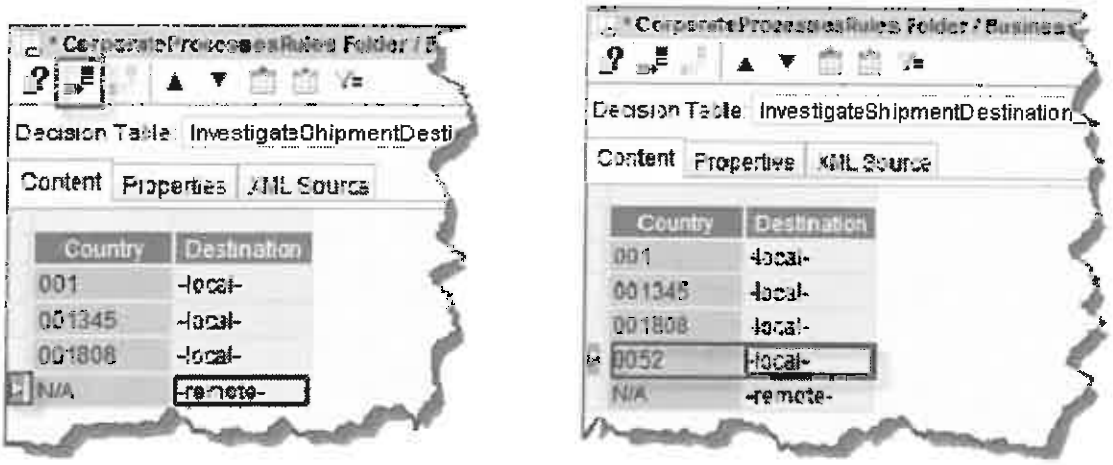
5. As a result, a Blaze Business Rule project named **CorporateProcessesRules** should be opened in the Project tab of Blaze Advisor. Note that the project already contains a document named **ShipmentDocBOM** in the **CorporateProcessesRules Folder -> TechnicalLibrary -> BOM** folder. This was imported from the IS document: **bpmDevSupport.docs.Blaze:ShipmentDoc**
Also it contains a rule instance named **InvestigateShipmentDestination Instance** located in the **CorporateProcessesRules Folder -> BusinessLibrary** folder:



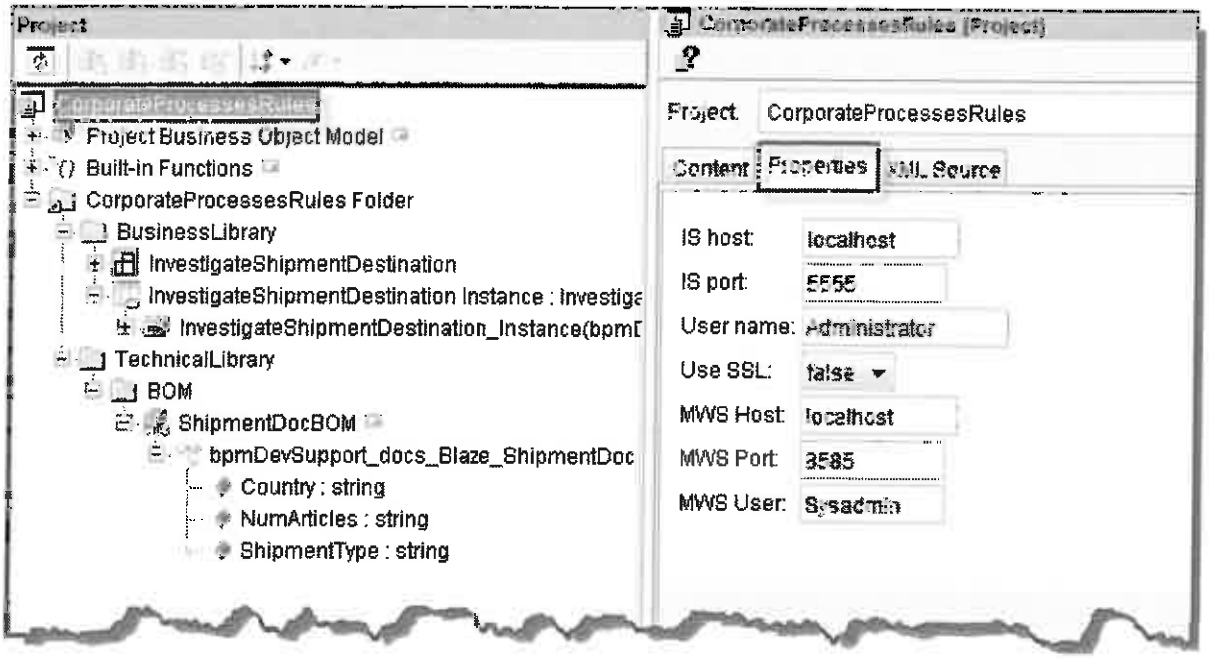
6. Double-click the Decision Table instance **InvestigateShipmentDestination Instance** to open it. It is the task of the rules contained in the Decision Table instance to investigate the value of the incoming **Country** field - as contained in the input document **ShipmentDocBOM** - and to return a string value in a field called **Destination**. As Acme is located in the United States, country codes of 001 (US), 001345 (Cayman Islands), and 001808 (Hawaii) should be handled as **-local-** shipments, others should be **-remote-**.



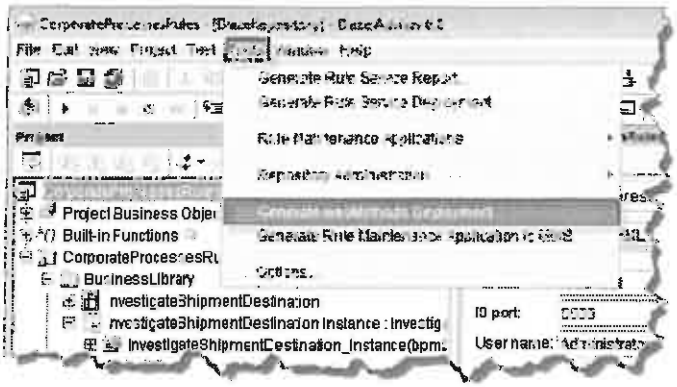
Mark the last row of the Decision Table and insert a new row above. Add country value **0052** (Mexico) as another **-local-** country code:

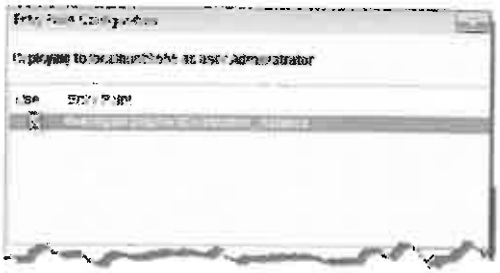


7. Save your definitions.
8. Click to compile your rule in Blaze Advisor.
9. From the Project tab, double-click the **CorporateProcessesRules** project. Ensure that the project connectivity parameter on the Properties tab fit to your webMethods environment:



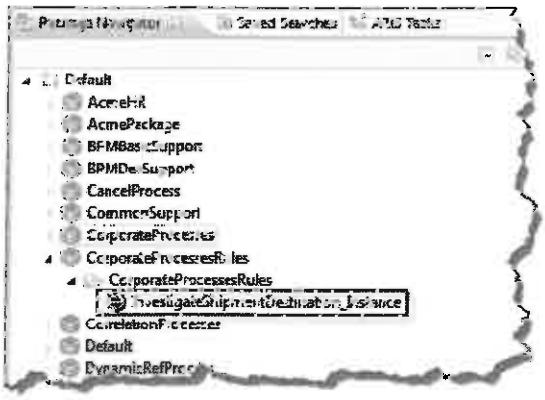
10. Select **Tools** -> **Generate webMethods Deployment** to deploy the **InvestigateShipmentDestination** rule instance to your Integration Server. If asked, specify **manage** as password.



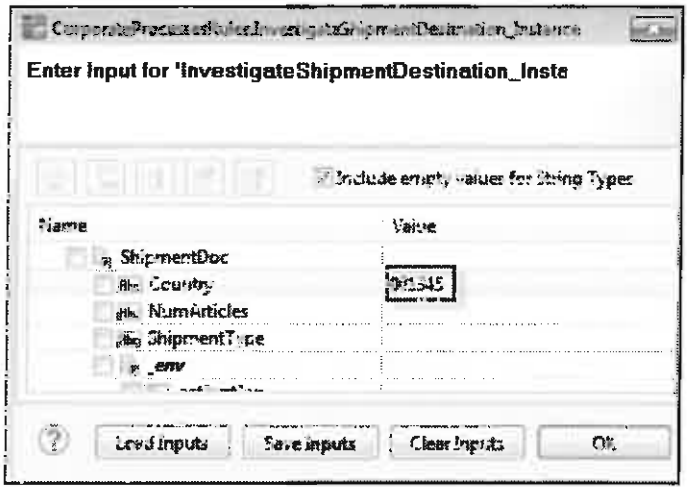


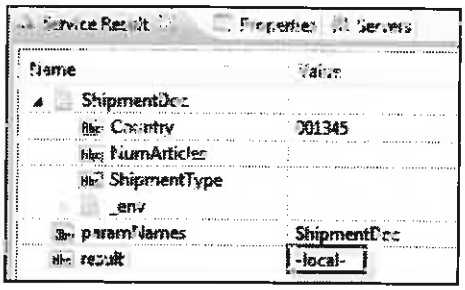
11. If not already started, launch Software AG Designer.

12. Switch to the **Process Development** perspective and refresh the content of the Package Navigator view. After successful deployment (step 10), there should be a new IS package named **CorporateProcessesRules** containing a rule service called **InvestigateShipmentDestination_Instance**:



13. To test the rule service, right-click the service in the Package Navigator view and select **Run As** -> **Run Service** from the context menu. Specify different country codes and check the rule service results in the Service Result view.

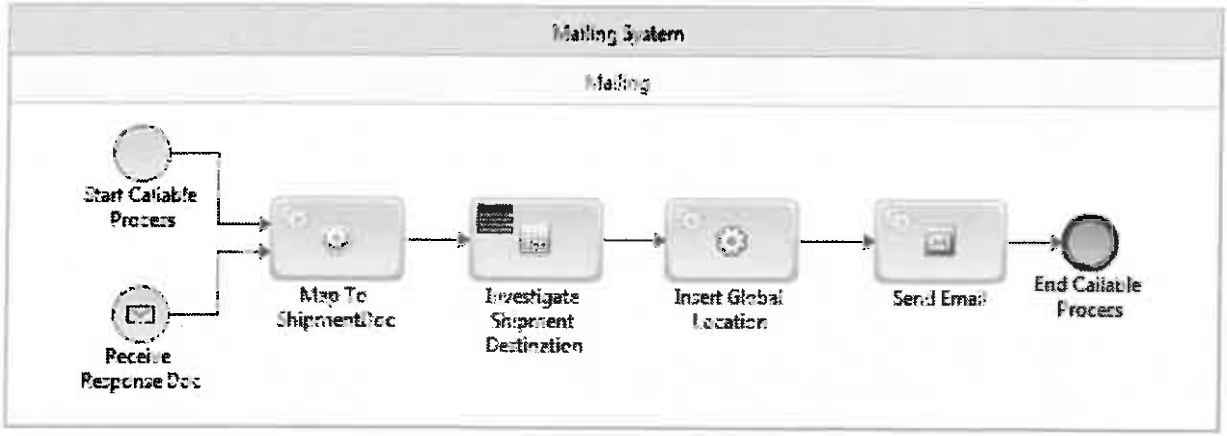




14. Open the existing process model **NotifyCustomer** as contained in your **CorporateProcesses** process project and perform the following modifications:

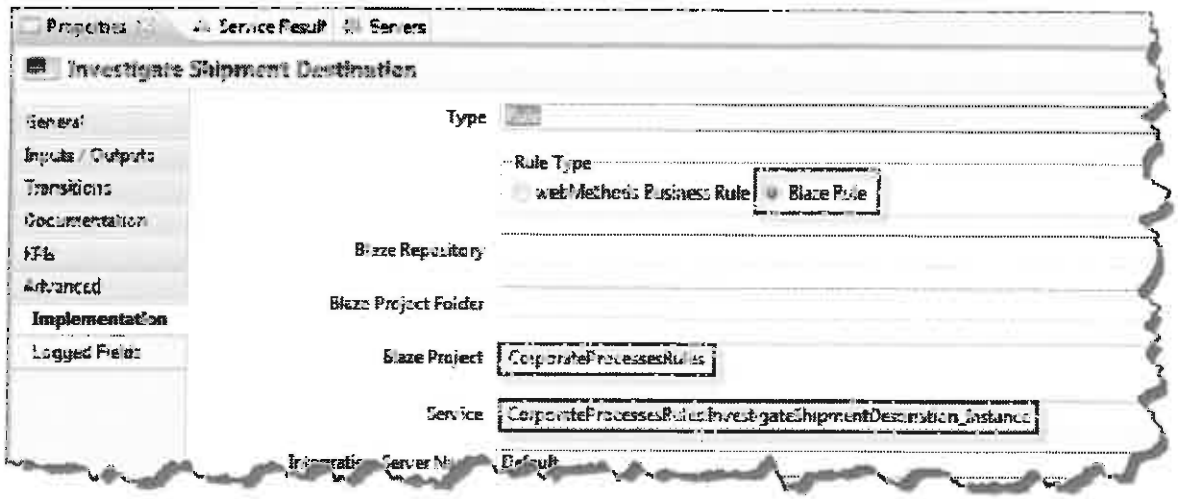
    a) You will need to resize the Mailing swimlane on the right side. To do this click on the internal pool Mailing System, and drag the pool to the right. This will make the swimlane wider.

    b) Add two Service Task Activities of type **IS Service** named **Map To ShipmentDoc** and **Insert Global Location,** and a Rule Task Activity named **Investigate Shipment Destination.** Adjust step images and add transitions to correspond with the following image:

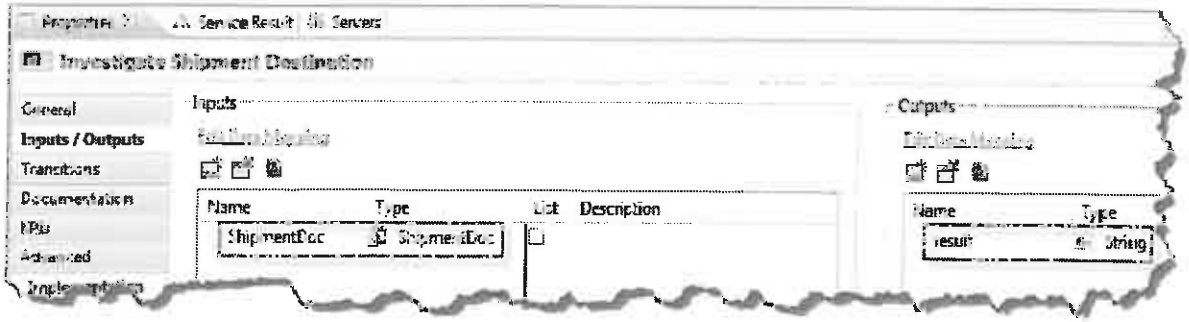


    c) Drag the service **bpmDevSupport.maps:OrderResponseToShipmentDoc** from the Package Navigator view and drop it on the **Map to ShipmentDoc** step. This will automatically configure the step to invoke this service.
    Ensure that the Join Type at the **Map To ShipmentDoc** step is of type **Unsynchronized Or.** Also check that the step's Inputs/Outputs have been set accordingly.

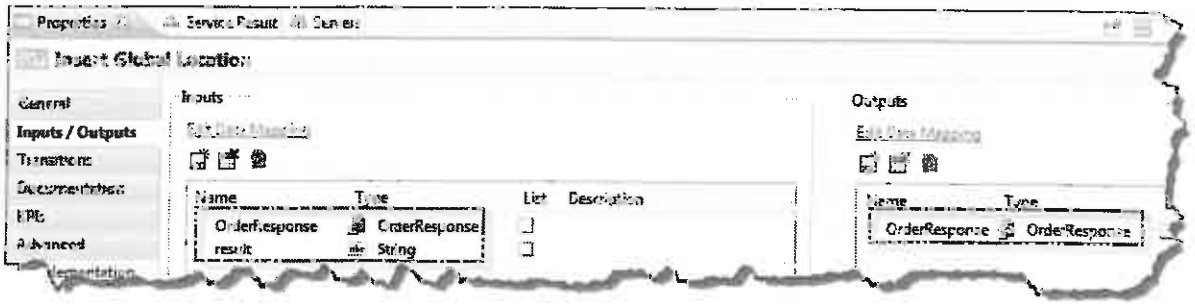
    d) To configure the Rule Task Activity, drag and drop the Blaze Business Rule Service **CorporateProcessesRules:InvestigateShipmentDestination_Instance** onto the step **Investigate Shipment Destination.** Its Implementation properties should now look like this:



Ensure the input of Rule Task Activity contains **ShipmentDoc** and the output contains **result:**

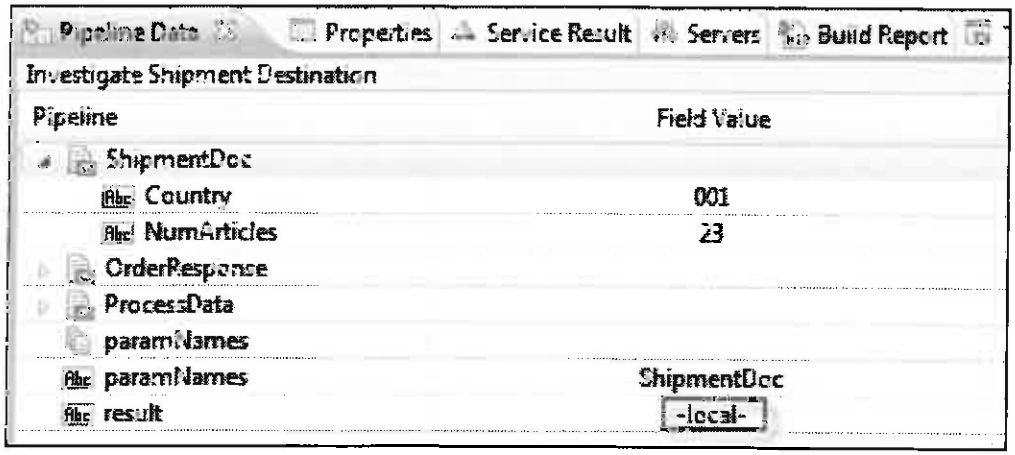


e) Assign the IS service **bpmDevSupport.utils:insertGlobalLocation** to the step **Insert Global Location.** Ensure the input of step contains **OrderResponse** and **result** and the output contains **OrderResponse:**



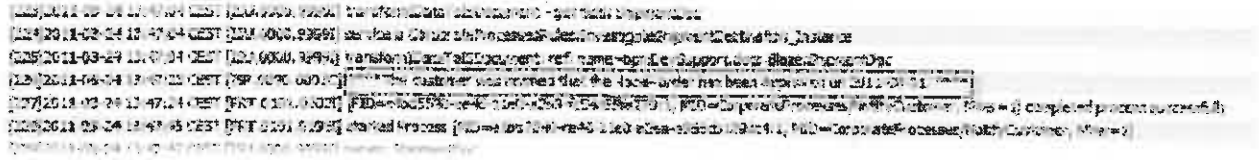
15. Save, build and upload your **NotifyCustomer** process.
16. Switch to the **Process Debugging** perspective.

    Click the Debug Selected Process icon to start a debugging session. For the IS input document, load the file ***<workshop_dir>*\Exercise20b\Resources\Ex20b_input.txt.** Note the provided Global Country Code is **001.** Step thru the process. Use the Pipeline Data view to ensure the step data of the step **Insert Global Location** contains the expected rule result:

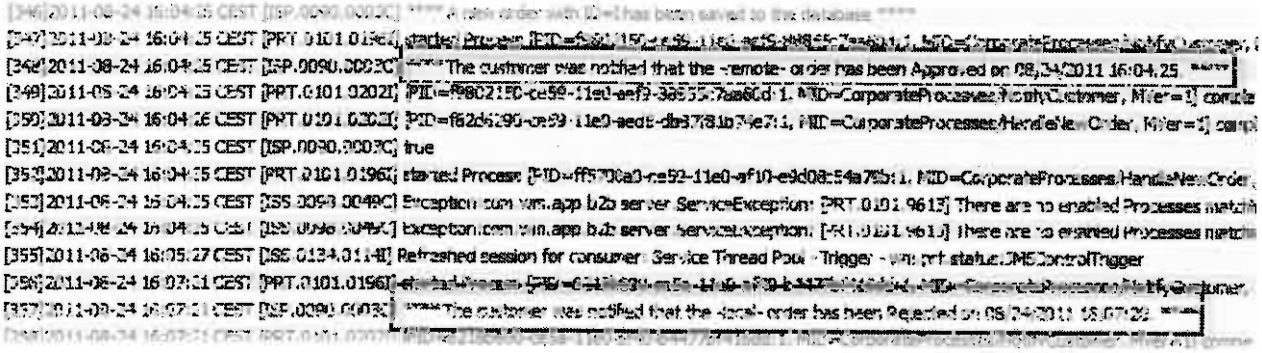


Click Run/Resume to complete your process in Debugger.

17. Open the Integration Server's server.log file by using an editor or the IS Administration console. If using an editor the server.log file can be found in the folder C:\SoftwareAG\IntegrationServer\logs. You should see an enhanced message like "**** The customer was notified that the -**local**- order has..." written by the **Send Email** step:



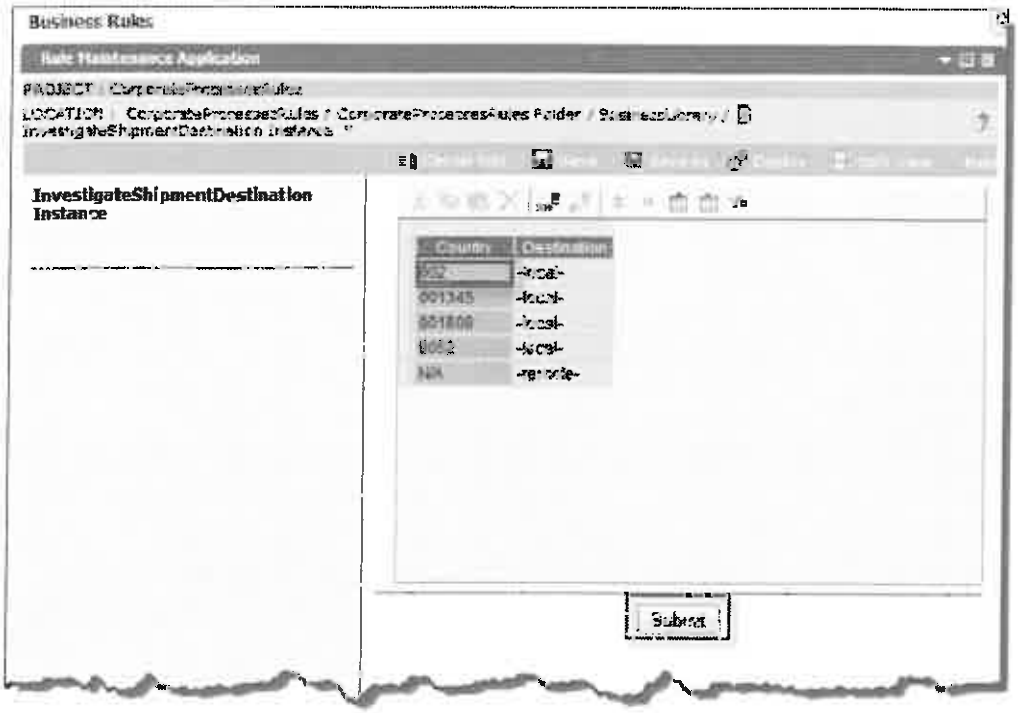
18. Test your entire **HandleNewOrder** process.
    Note: Remember that **HandleNewOrder** invokes **NotifyCustomer** as twice, once as a BPMN Callable Process, second as a webMethods Referenced Process.
    Test both cases by double-clicking the provided HTML forms:
    - ***<workshop_dir>*\Exercise20b\Resources\Ex20b_submit_DE_validOrder.html**
    - ***<workshop_dir>*\Exercise20b\Resources\Ex20b_submit_US_invalidOrder.html.**
    In both cases click the **Submit** button to start the process. If asked for authentication use **Administrator/manage**.
19. Login to My webMethods as **Administrator/manage**. Navigate to the **Applications** -> **Monitoring** -> **Business** -> **Process Instances** page. Ensure that the first process instance completed successfully.
    Because of invalid data, the second process instance should wait for a User Task first. Navigate to the **Applications** -> **Monitoring** -> **Business** -> **Tasks** -> **Task List Management** page. Select and open the User Task instance and **Abort** the User Task instance. This should complete the process also.
20. Inspect the IS Server log to check that the first order is flagged as an accepted -**remote**- order. The second order should be flagged as a rejected -**local**- order in the server.log file.



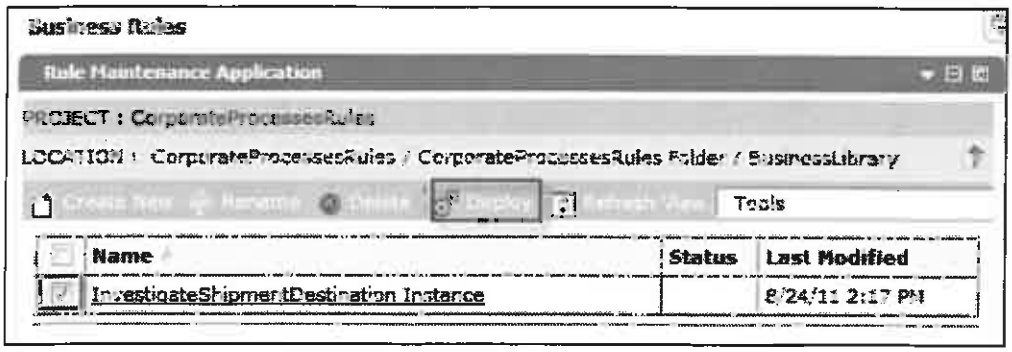
21. Switch back to the Blaze Advisor tool. Select **Tools** -> **Generate Rules Maintenance Application to MWS**. If asked for a **Sysadmin** password, specify **manage**.

22. After successful deployment, use a browser to login to My webMethods as **Administrator/manage.**
Navigate to **Applications** -> **Administration** -> **Business** -> **Business Rules.** Select the project name **CorporateProcessesRules** (it may take awhile the first time you do this). Drill into: **CorporateProcessesRules Folder** -> **BusinessLibrary** -> **InvestigateShipmentDestination Instance**

Note: If prompted to trust a digital certificate, check **Always trust...** and select **Run.**
In the tool bar of the Rule Maintenance Application portlet select **Edit** to allow modifications of your Decision Table instance. Replace the Country value 001 by **002** and click **Submit** to commit your modification:

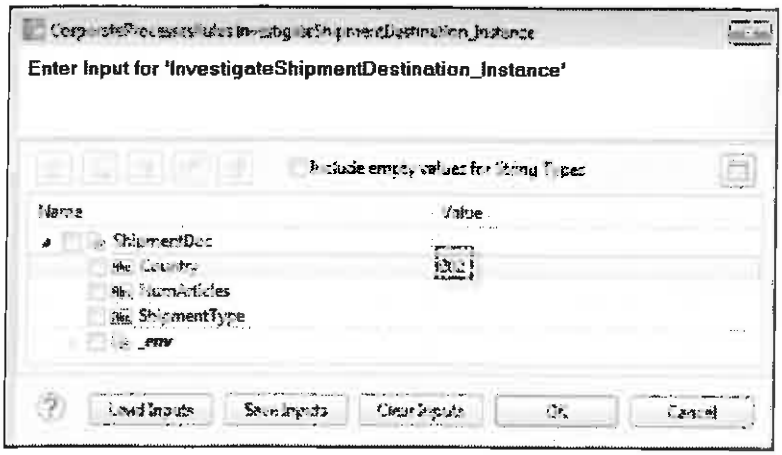


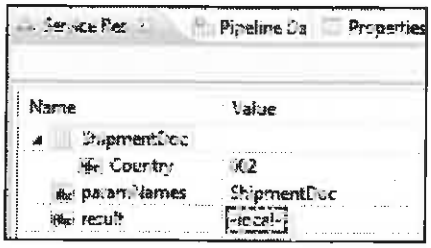
23. Click  to step back to the previous view. If prompted, confirm to save the submitted changes. In the appearing view, check the modified Decision Table instance and **Deploy** it to your IS:



Confirm the deployment with **Submit.**
Important! Before you press the **Submit** button, click the plus sign (+) beside **Advanced Settings**, and UNCHECK the box beside **Distribute Rules to a cluster.**

24. Double-check the modification done in the RMA by re-running the Blaze Business Rule Service from Designer. To do so, refresh the Package Navigator view. Right-click the rule service **CorporateProcessesRules:InvestigateShipmentDestination_Instance** in the Package Navigator view and select **Run As -> Run Service** from the context menu. Specify **002** as Country value and check the Blaze Business Rule Service result in the Service Result view.





Check Your Understanding

1. What happens when you perform a webMethods Deployment from Blaze Advisor?

2. Is it always necessary to deploy a Rules Maintenance Application to MWS?

This page intentionally left blank.

Exercise 21:
Starting a Process from a CAF Portlet

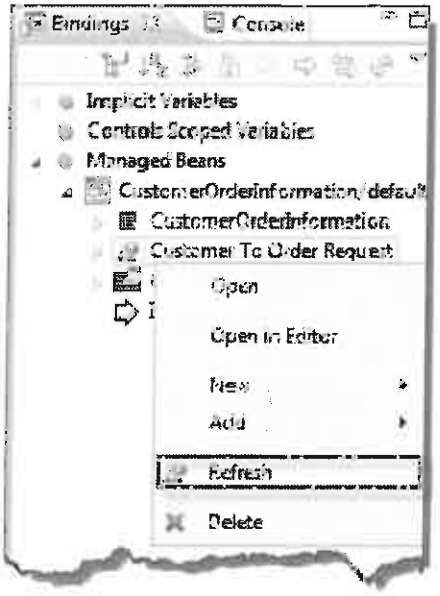
Overview

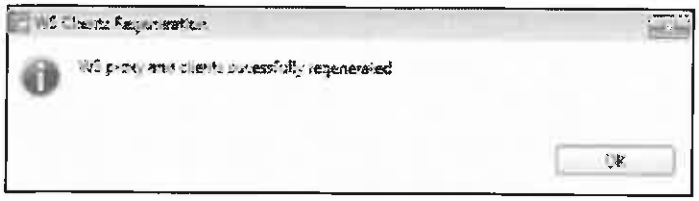
In this exercise, you will use a provided CAF portlet user interface that will allow a customer to create an order and send it to the **HandleNewOrder** process.
The portlet UI contains a form that invokes an existing IS service as a Web service. The invoked service converts the input data into an IS document and publishes it to start a new process instance. Moreover, input data on the UI are checked by several Validators.
You will enhance the portlet UI to invoke another Web service to initialize the contents of a dropdown control. All necessary Web services are already provided in your IS environment.

Steps

1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Launch Software AG Designer and ensure you are in **UI Development** perspective.
3. Switch to the Solutions view. Import a provided CAF project containing the portlet:
   a) Select **File -> Import**
   b) Choose **Software AG > Existing CAF Projects into Workspace**. Click **Next**.
   c) Click **Select archive file** and browse for the provided zip file:
   <*workshop_dir*>\Exercise21\Resources\CustomerUIBasic.zip
   d) Select the zip file and click **Open**.
   e) Click **Finish**.
   As a result you should get a new project named **CustomerUI** containing a portlet called **CustomerOrderInformation**.
4. Open the Package Navigator view. If asked for authentication, provide **Administrator/ manage**. Open the folder **bpmDevSupport\webservices** of the **BPMDevSupport** package.
5. Use the Solutions view and navigate to **User Interfaces -> CustomerUI -> CustomerOrderInformation -> Default**. Open the **Default** view of the imported CAF portlet.
   Note:
   The view was previously created by dragging and dropping the Web service descriptor **bpmDevSupport.webservices:CustomerToOrderRequestWS** onto the design canvas. By selecting the Web service operation **CustomerToOrderRequest**, all fields related to the Web service operation signature were provided on the form using default controls.
   In addition, a Web service connector named **Customer To Order Request** was automatically added to the Bindings view of the portlet.
   Note that some of the default controls have been doctored, e.g. **Delivery Methods** has been changed into a Radio Button group. Moreover, the provided CAF portlet UI already contains Validators for the input fields **ZIP** (Regular Expression Validator), **Email** (Email Validator) and **Credit Card Number** (Credit Card Validator).

6. In the Bindings view, right-click the existing Web service connector named **Customer To Order**. Select **Refresh** from the context menu to regenerate the Web service proxy and client for your environment.

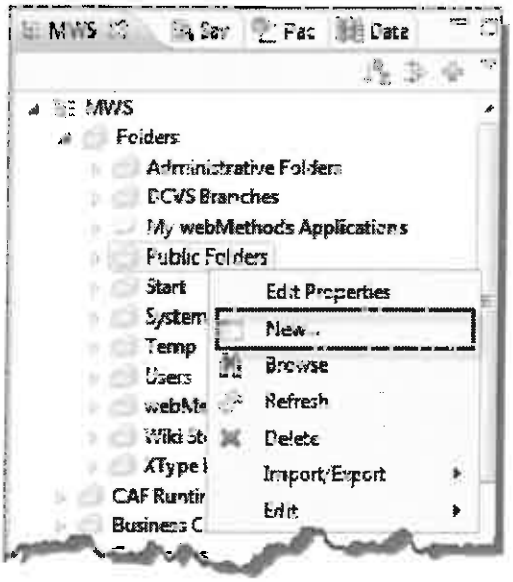


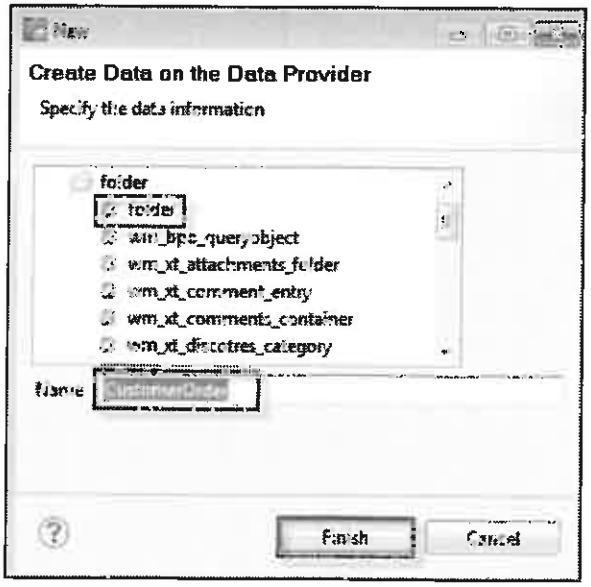


7. Use the Servers view to add the **CustomerUI** project to the Remote My webMethods Server. Ensure that it is synchronized with the server. If asked for authentication, provide **Sysadmin/manage**.

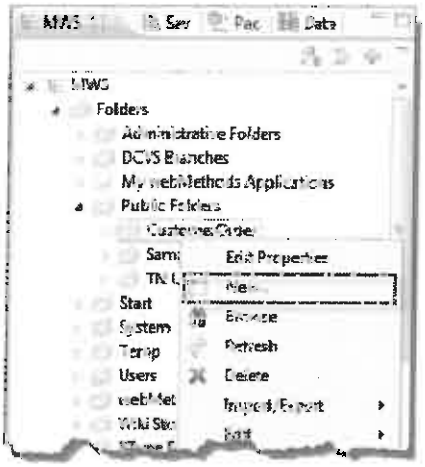
8. To enable the portlet to run as an MWS portlet application, you have to add it to a MWS Public Folder:

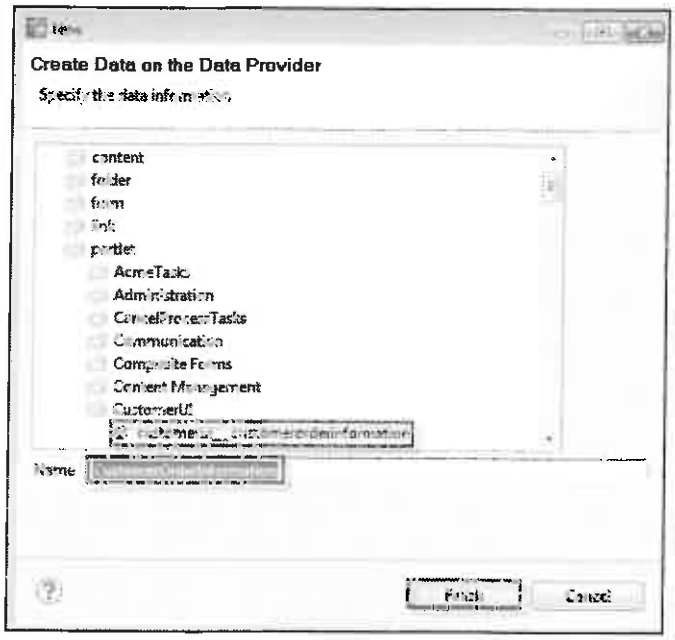
   a) Use the MWS Admin view and create a new folder below **Public Folders** named **CustomerOrder**. If asked for authentication, provide **Sysadmin/manage**.





b) In the MWS Admin view, add a new portlet named **CustomerOrderInformation** to your **CustomerOrder** public folder. The portlet is associated with the existing **customerui_customerorderinformation** portlet.





9. To test your CAF portlet:

   a) Launch a browser and navigate to **http://localhost:8585**. Login as **Sysadmin/manage**.

   b) Navigate to **Public Folders -> CustomerOrder**.

   c) Click the **CustomerOrderInformation** link.

   d) In the portlet UI, enter data in all of the fields. Enter at least one product item by clicking the Append Row button.
   Because of the existing Credit Card Validator, you have to provide a valid credit card number. For testing you can use one of the following valid sample credit card numbers, or just use one of yours ☺:

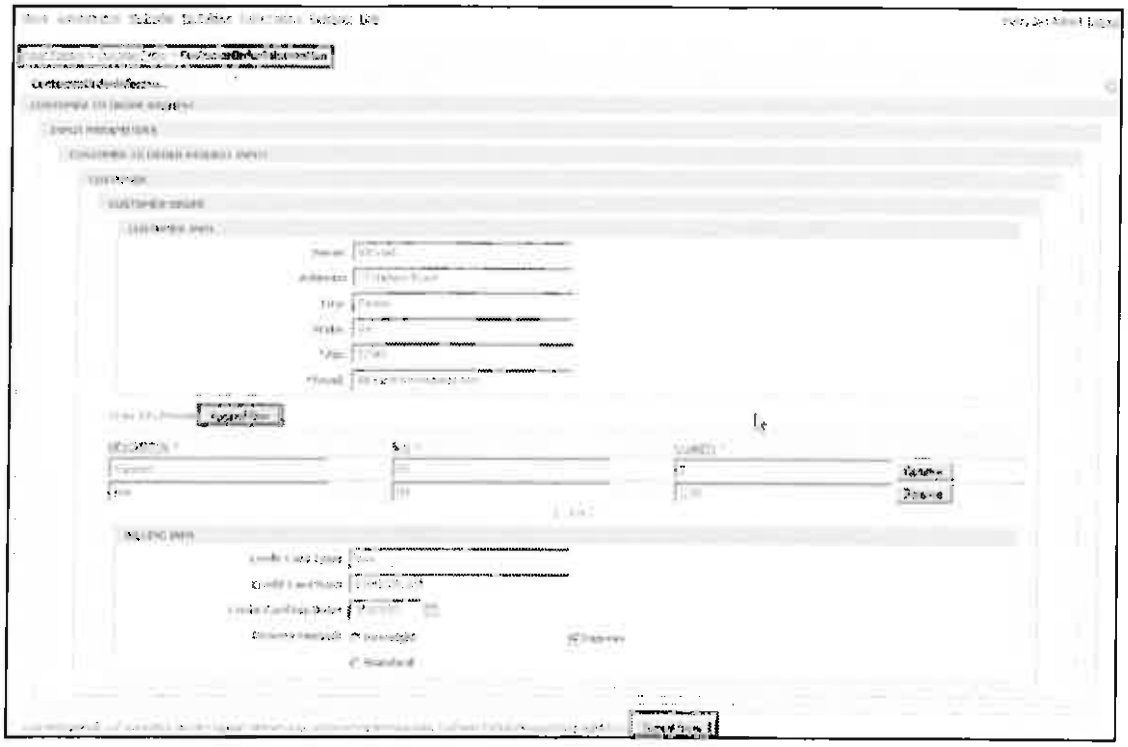
      i) Master Card: **5105105105105100**

      ii) VISA: **4111111111111111**

      iii) American Express: **378282246310005**

      iv) Discover: **6011111111111117**

e) Finally click the **Submit Order** button:



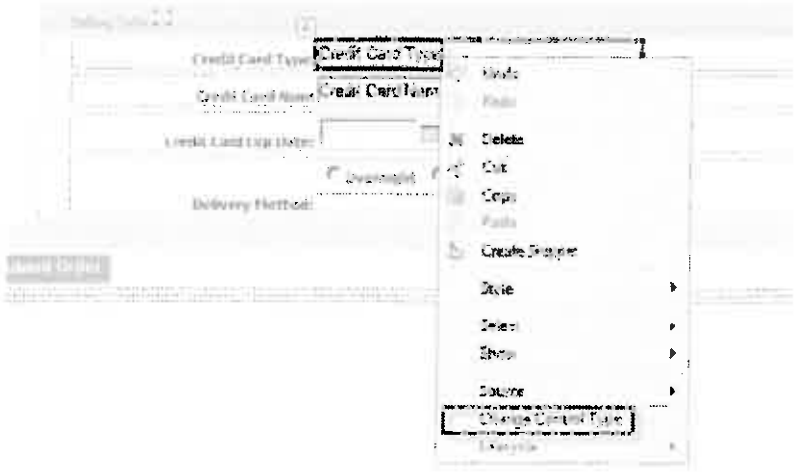
10. Login to My webMethods using **Administrator/manage**. Navigate to the **Applications** -> **Monitoring** -> **Business** -> **Process Instances** page. Ensure a new process instance has been started and completed successfully.

11. Open the IS server.log file. The server.log file should contain your submitted order data:

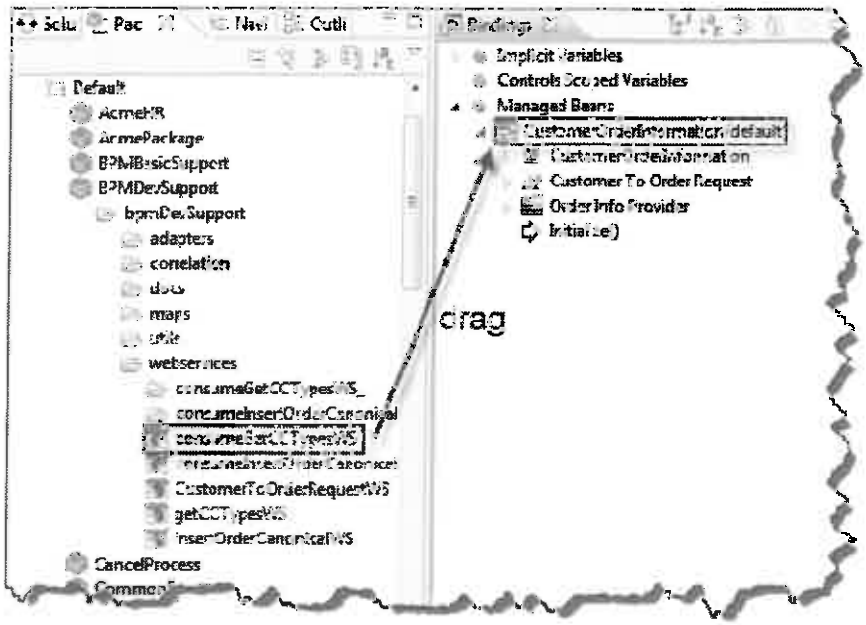
```
<PurchaseOrderRequest>
  <PurchaseOrder>
    <deliverTo>
      <PhysicalAddress>
        <cityName>
          <FreeFormText>Reston, VA</FreeFormText>
        </cityName>
        <addressLine1>
          <FreeFormText>27 Harbour Road</FreeFormText>
        </addressLine1>
        <addressLine2>
          <FreeFormText>Reston, VA</FreeFormText>
        </addressLine2>
        <addressLine3>
          <FreeFormText>Reston, VA</FreeFormText>
        </addressLine3>
        <NationalPostalCode>12345</NationalPostalCode>
        <regionName>
          <FreeFormText>VA</FreeFormText>
```

12. Customize the Text Input field **Credit Card Type** to become a dropdown with choices automatically filled by a Web service invocation:

    a) In Designer, open **CustomerOrderInformation/default.view**, right-click Text Input **Credit Card Type** contained in the Property Group **Billing Info.**
    Change its control type to **Input/Dropdown:**

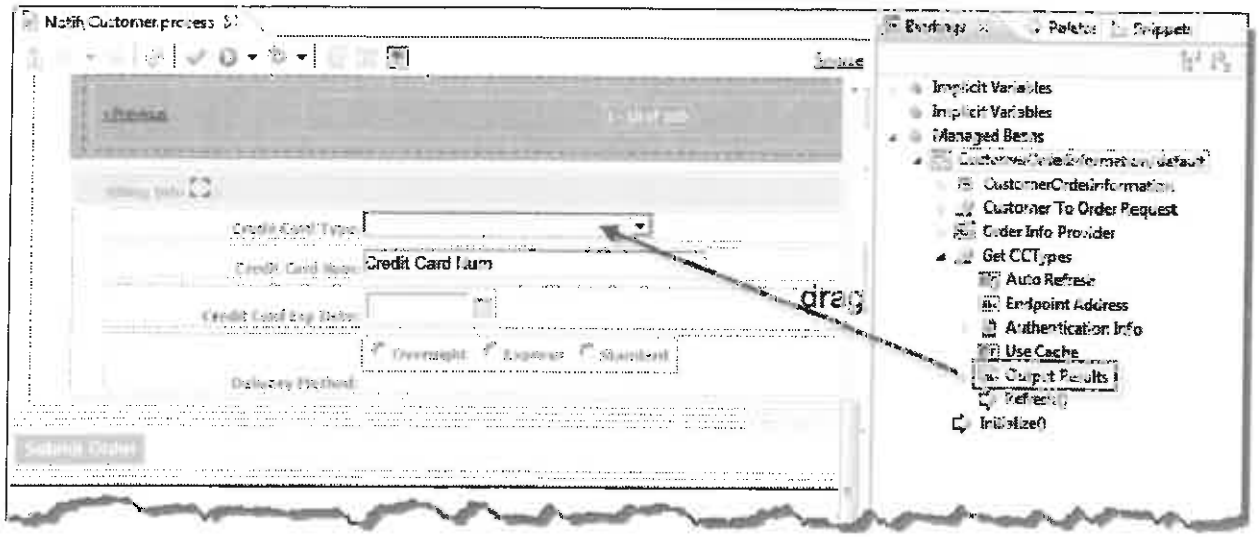


    b) Locate the Web Service Descriptor **bpmDevSupport.webservices:getCCTypesWS** in the Package Navigator view. Drag and drop the Web Service Descriptor **getCCTypesWS** from the Package Navigator view onto the **CustomerOrderInformation/default** Managed Bean in the portlets Bindings view:



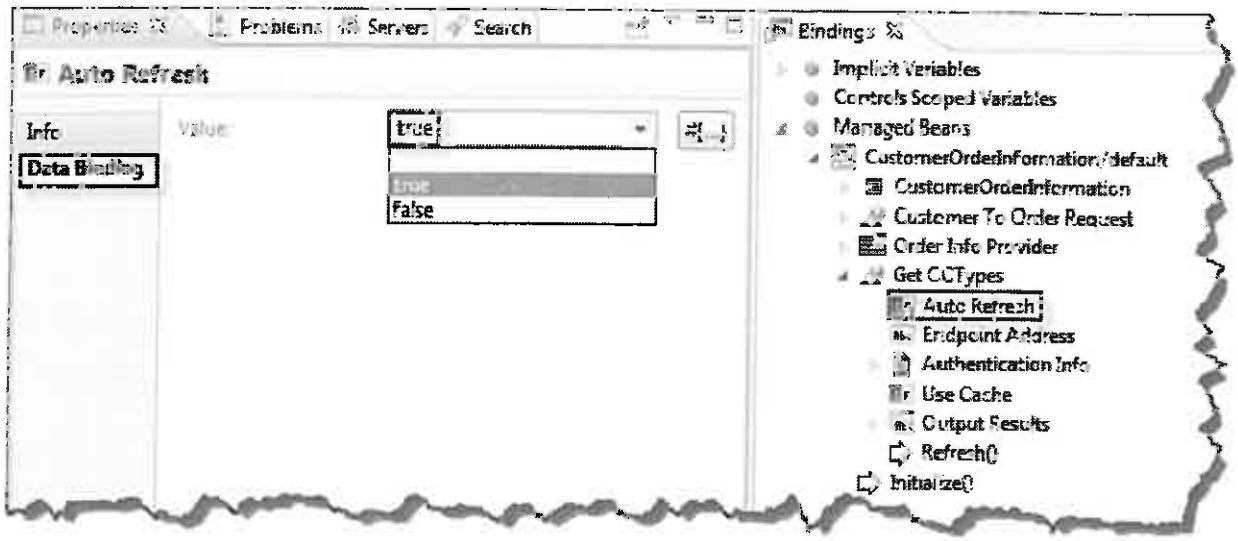
    In the appearing wizard, click the Next button, select the Web service operation **IgetCCTypesWS_PortType/getCCTypes(getCCTypes)**, and click **Finish.**
    Note: This will add another Web service connector to the portlets Managed Beans.

c) From the Bindings view, drag the **Output Results** of the **GetCCTypes** Web service connector to the **Credit Card Type** dropdown control on the view **CustomerOrderInformation/default.view** in the editor pane.



This is to fill the dropdown with the outputs of the Web service.

d) Configure the Web service to be called automatically before the view is rendered: From the Bindings view, open the property **Auto Refresh** of the **GetCCTypes** Web service connector and set its **Data Binding Value** in the Properties view to **true**:



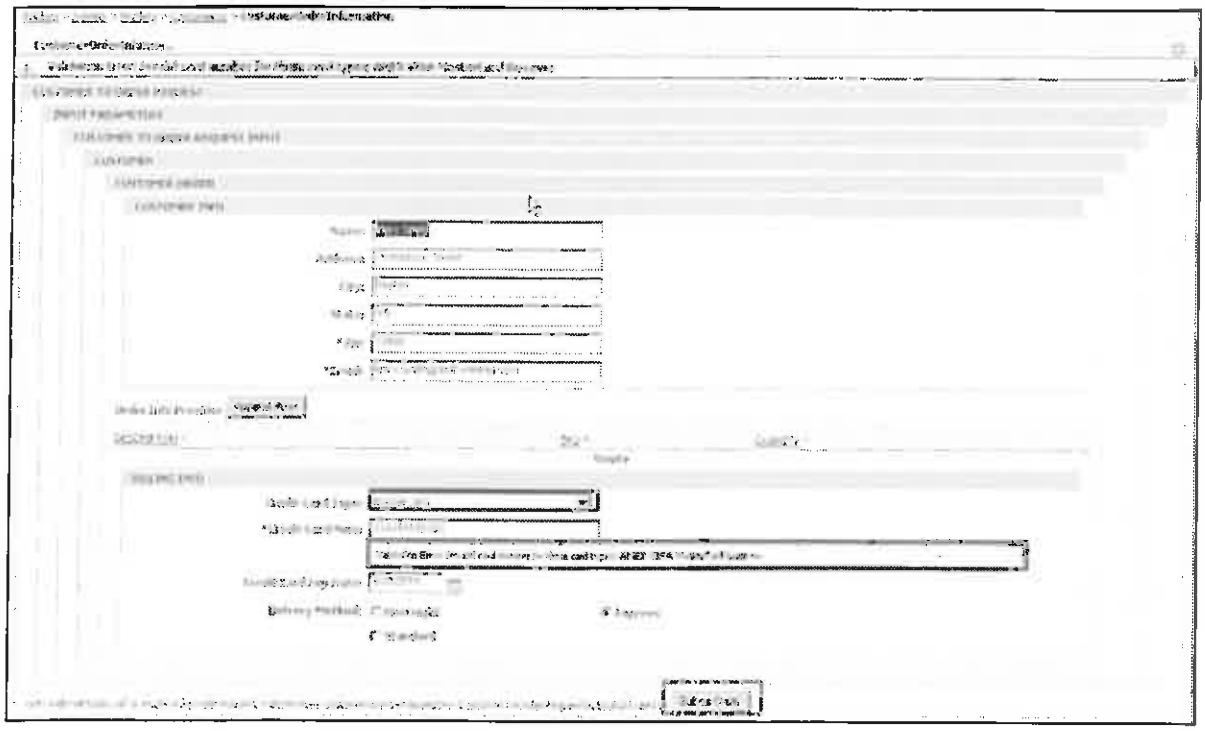
13. Save your work.

14. Using the Servers view, republish your **CustomerUI** project to My webMethods Server.

15. Test your portlet UI from Designer:

    a) Click the dropdown from the menu bar of the design canvas. Select **My webMethods Server (Remote) at localhost** from the dropdown choices. Now click on the green button to run the portlet on the MWS. Use **Sysadmin/manage** when prompted for authentication.

    b) Notice the entries in the Credit Card Type dropdown input box. Enter some data including an invalid credit card number. Click the **Submit Order** button and confirm there is a credit card number validation error message at the top of the page and under the field:



    c) Change the credit card number into a valid one. For testing you can use one of the valid sample numbers mentioned in step 9d. Then click **Submit Order** again.

16. Login to My webMethods using **Administrator/manage**. Navigate to the **Applications** -> **Monitoring** -> **Business** -> **Process Instances** page. Ensure a new process instance has been started and completed successfully.

Check Your Understanding

1. Why did you have to publish your application in step 7 before adding the portlet in the MWS Admin view?

2. What protocol was used to make the invocation that retrieved the credit card types?

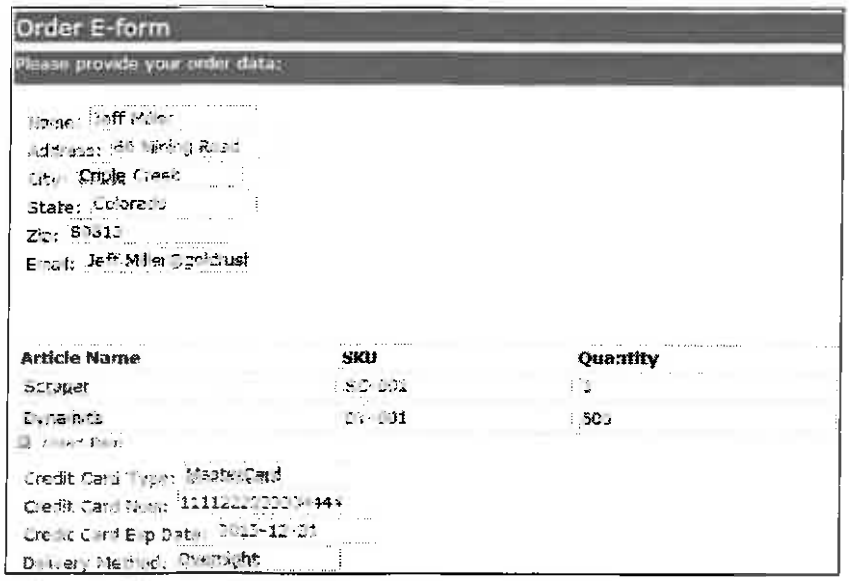
3. What was the purpose of setting the Auto Refresh property to true?

4. If you did not republish the application in step 14, what results would you have expected?

Exercise 22:
Starting a Process from an E-form

Overview

In this exercise, you will modify the **HandleNewOrder** process so that it can also be started by an InfoPath e-form that contains Order data:



To enable this scenario, you will configure an E-form environment using My webMethods, create a new IS document type by importing an e-form template and finally, add a new parallel Start Message Event to the process along with a new Order transformation step.

Steps

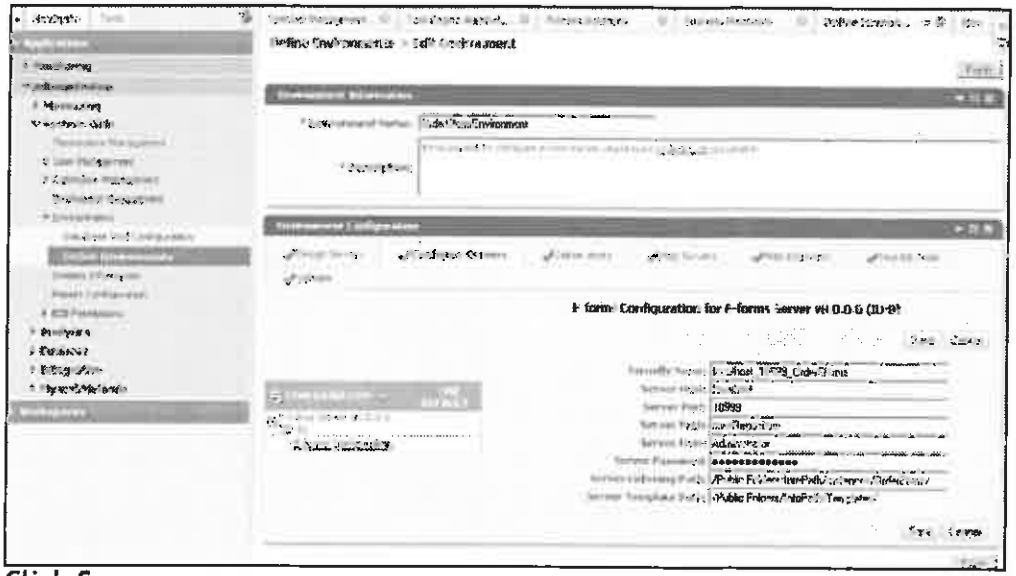
1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Start Designer and switch the UI Development perspective.
3. Create a Content Repository in the public folder structure of your MWS:
   a) Import the folder structure and e-form template:
      I. Select **File -> Import**
      II. Choose **General > Existing Projects into Workspace**. Click **Next**.
      III. Click **Select archive file** and browse for the provided zip file: ***<workshop_dir>*\Exercise22\Resources\ EformContentProject.zip**
      IV. Select the zip file and click **Open**.
      V. Click **Finish**.

      As a result, you should see a new project named **EformContentProject** displayed in the Navigator view.

   b) In the Servers view, select **My webMethods Server (Remote)** and add the project **EformContentProject** to your MWS. From the Servers view, publish the project to MWS.

c) Switch to the MWS Admin view and refresh its content. If asked for authentication, provide **Sysadmin/manage**. As a result of the step above, the following folder structure should be available:

4. Login to My webMethods as user **Administrator/manage.**
5. Navigate to **Applications -> Administration -> System Wide -> Environments -> Define Environments.** Add a new Environment with the name **OrderEformEnvironment.** Provide **"Environment used to work with InfoPath e-forms"** as description. Save your definition. Then click the environment name to edit/modify it like this:
   a) On the **Design Servers** tab, add the **E-forms Server** as associated Logical Server. To do so, click the **Add...** button and select the existing **E-forms Server v8.x.x.x.** Click **OK.**
   b) On the **Configure Servers** tab, expand the tree and select the **E-forms Configuration** link. Configure the E-forms Server with an E-forms Configuration containing the following definitions:
      - Friendly name: **localhost_10999_ OrderEforms**
      - Server Host: **localhost**
      - Server Port: **10999**
      - Server Path: **mwsRepository**
      - Server User: **Administrator**
      - Server Password: **manage**
      - Server Listening Path: **/Public Folders/InfoPath/Instances/OrderEform/**
      - Server Template Path: **/Public Folders/InfoPath/Templates/**



Click **Save.**

c) On the **Define Hosts** tab, add a host to your environment with the following values:

- Display Name: **localhost**
- Host Name or IP Address: **localhost**

Click **OK.**

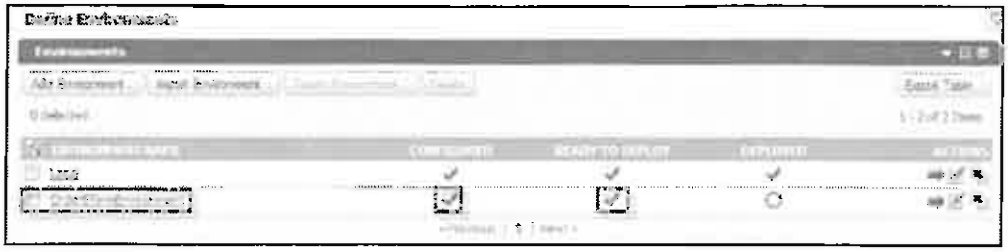
d) On the **Map Servers** tab, select the **Map All** button to map the logical E-forms Server to the physical host **localhost.**

e) On the **Map Endpoints** tab, accept all the defaults (port **15006**, which is the default Process Engine Listener port running at IS:15006). Click **Save.**

f) Select **Map DB Pools** tab and click on **Save** only.

g) On the Validate tab, you should see a green checkmark with the message "Valid Configuration". Click on **Finish.**

h) Your OrderEformEnvironment should now be in the state CONFIGURED and READY TO DEPLOY:



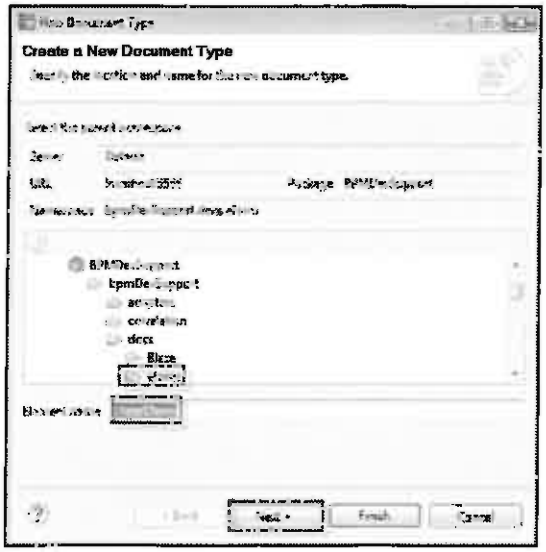
For the environment **OrderEformEnvironment** click in the ACTIONS column to start the deployment. On the next page, choose **Deploy All.** You should see a green checkmark with the message "Successfully deployed environment".

6. Create an IS document type and schema using an InfoPath e-form template. To do so:

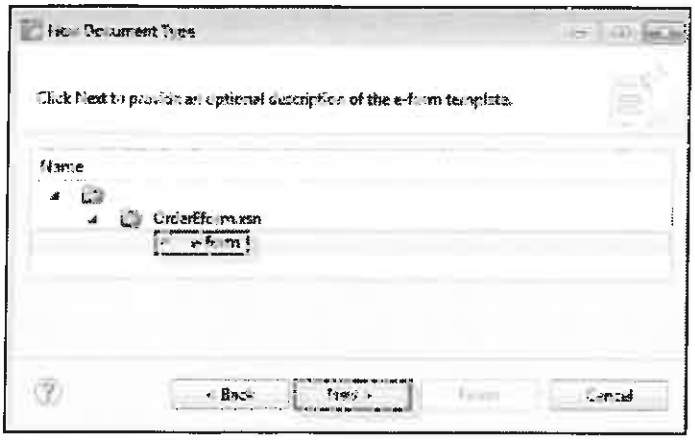
a) Open Designer and switch to the Process Development perspective.

b) In the Package Navigator view, drill down to the existing empty folder **bpmDevSupport.docs.eform.**

c) Right-click on the **eform** folder and select **New** to create a new Document Type. Provide **OrderEform** as element name for your new document type and click **Next:**

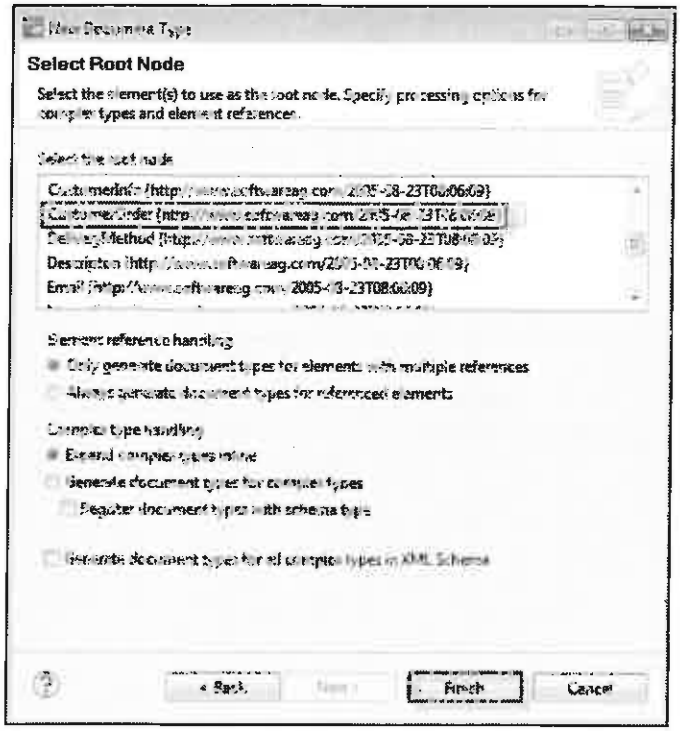


d) On the next dialog, select **Microsoft® InfoPath® E-form Template** as template type and click **Next**. Select **From Repository** and from the drop-down list select your Content Repository **localhost_10999_OrderEforms**. Click **Next**. Now drill down into the folder structure until you find **e-form** within OrderEform.xsn. Select the **e-form** row and click **Next**.



e) Skip providing an optional description. On the next panel, use the default schema, select Content model compliance **None** and leave Enable MTOM streaming unchecked.

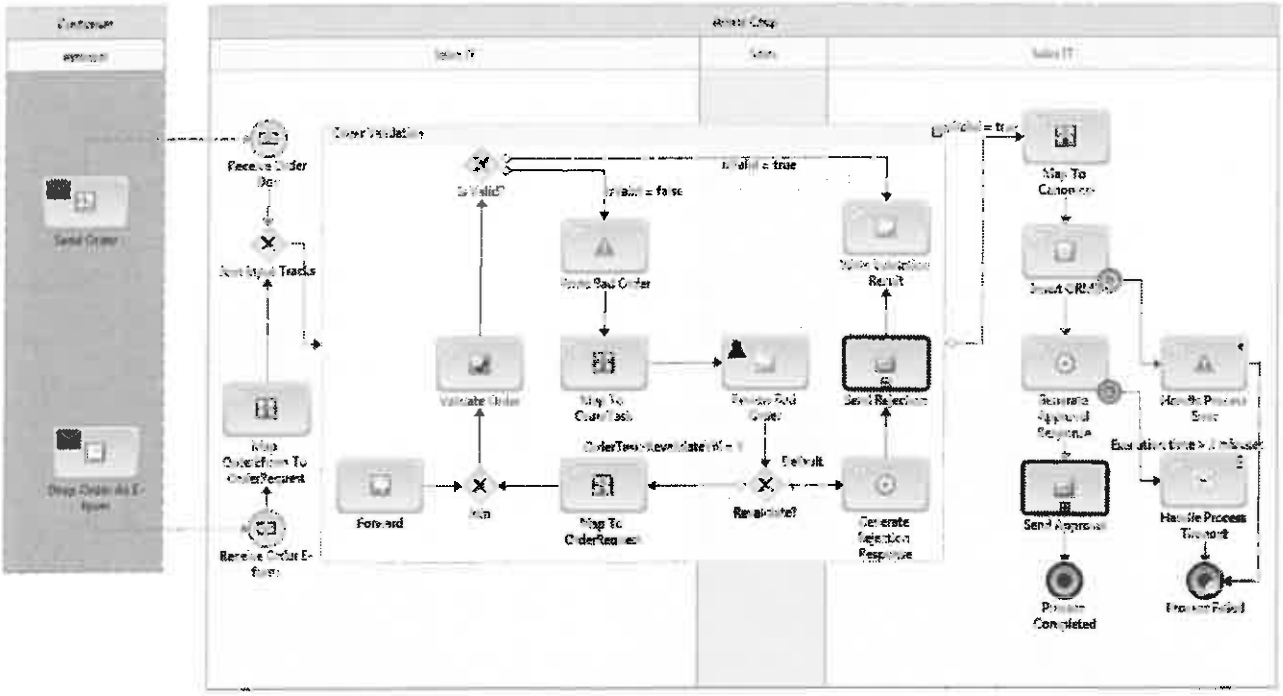
f) On the final panel, select **CustomerOrder** as the root node and leave all other settings unchanged. Click **Finish** and accept warnings.



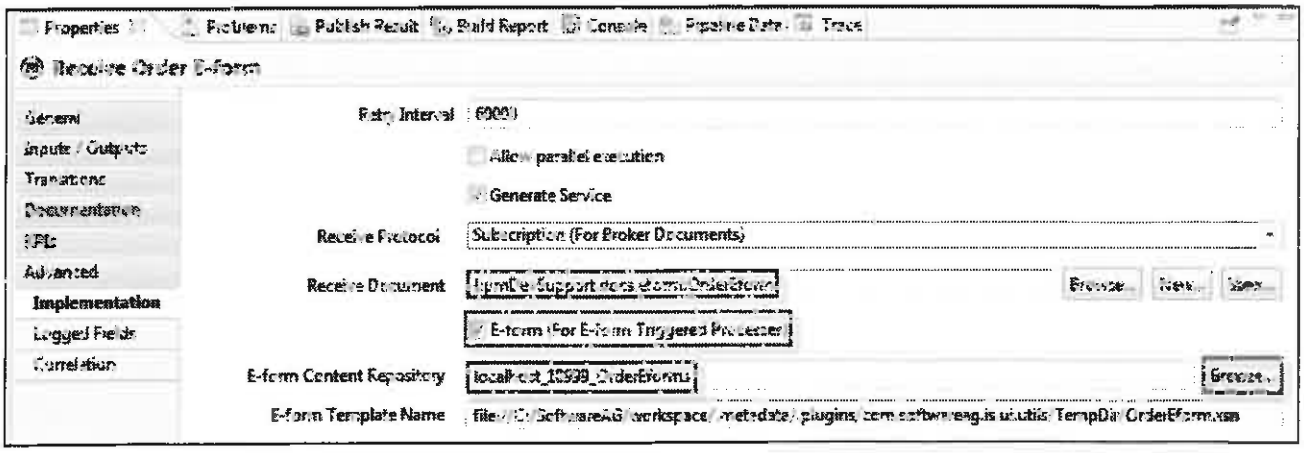
7. Open your **HandleNewOrder** process model as contained in the CorporateProcess project from the Solutions view.

8. Enhance your process model by adding:
   - a Start message Event named **Receive Order E-form**
   - a Service Task Activity named **Map OrderEform To OrderRequest**
   - a Send Task Activity named **Drop Order As E-form**
   - an Exclusive Gateway named **Join Input Tracks**

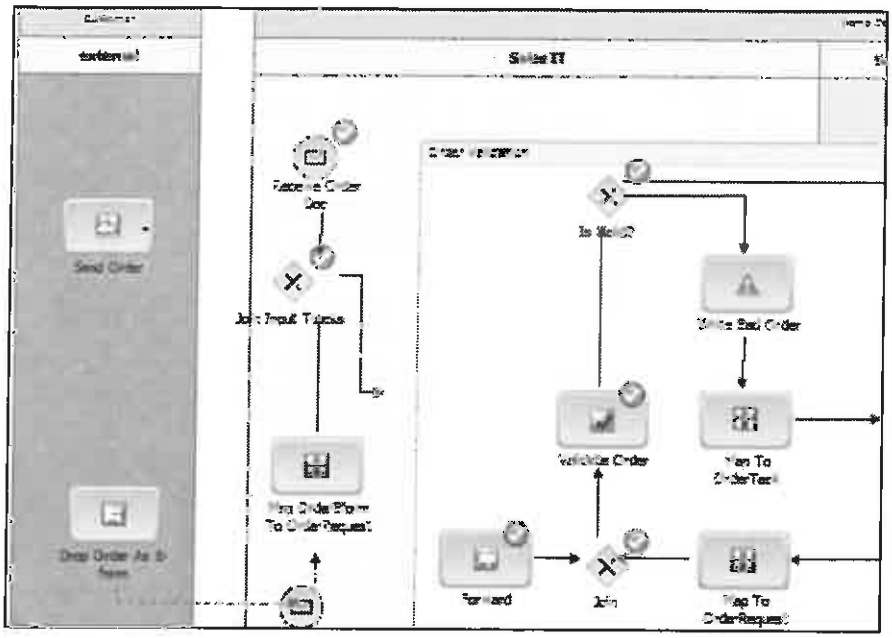
   Add transitions and adapt step images so that your process model corresponds to the following image:



9. Drag and drop the new IS document type **bpmDevSupport.docs.eform:OrderEform** onto the added Start Message Event to assign the Receive Document type and the E-form Template. Open the Start Message Event's Properties view. On the Implementation tab, click **Browse** assign your E-form Content Repository **localhost_10999_OrderEforms**:



10. Drag and drop the existing IS service **bpmDevSupport.maps:OrderEformToOrderRequest** onto the added Service Task Activity named **Map OrderEform To OrderRequest**. This is to assign the implementing IS service and to set the Inputs/Outputs.
11. Select the added Gateway step **Join Input Tracks.** Ensure its Join type is an **Unsynchronized Or** Join.
12. Save, build and upload the **HandleNewOrder** process.
13. First, ensure that your process can be started as before:
    a) Double-click ***<workshop_dir>*\Exercise22\Resources\Ex22_submit.html.** Click the Submit button. If asked for authentication use **Administrator/manage.**
    b) As the order data is valid, a new process instance should have completed successfully without any User Task interaction.
    Login to My webMethods as **Administrator/manage** and navigate to **Applications -> Monitoring -> Business -> Process Instances** to search for the process instance. Open its Details view. Check that the Event **Receive Order Doc** is the first in the trail of executed steps:

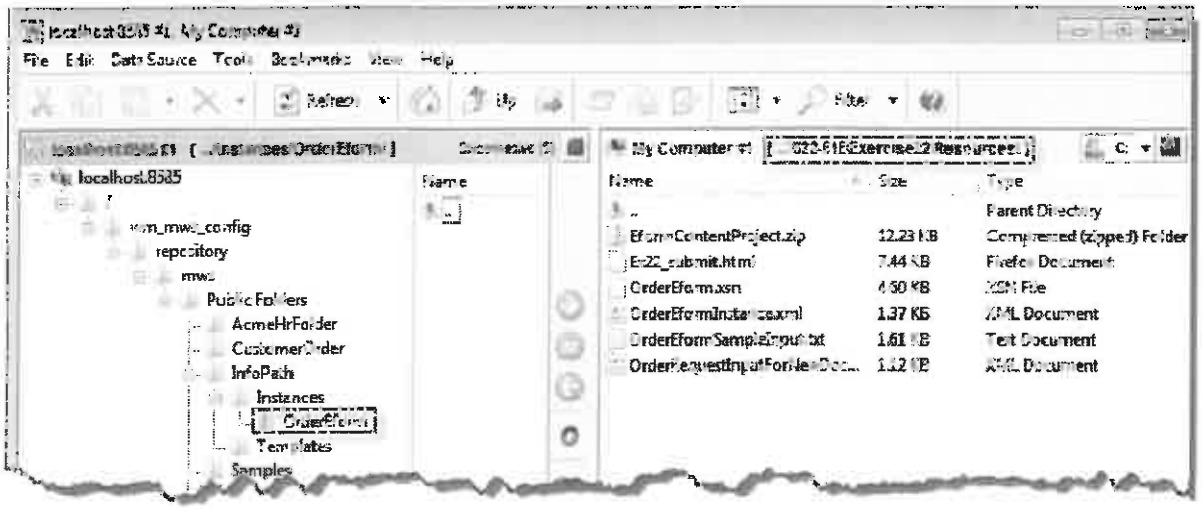


14. Now ensure that your process can be started by dropping an InfoPath e-form instance into your MWS-based Content Repository:

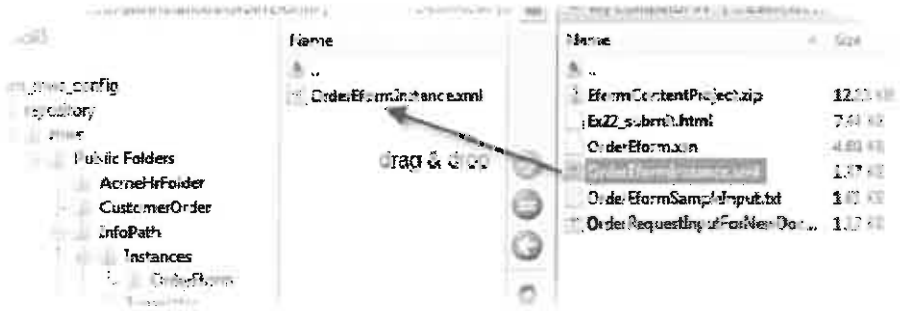
    a) Launch a WebDAV client (**Start-> All Programs -> Tools -> BitKinex -> BitKinex**).

    b) In the BitKinex tool tree view, locate and double-click the pre-configured WebDAV connection **localhost:8585** to connect to your MWS Content Repository.

    c) Use the WebDAV explorer view on the left side to expand the **Public Folders** and navigate to **Public Folders -> InfoPath -> Instances -> OrderEform.**
    Use the file explorer view on the right side to navigate to your local folder: ***<workshop_dir>*\Exercise22\Resources.**

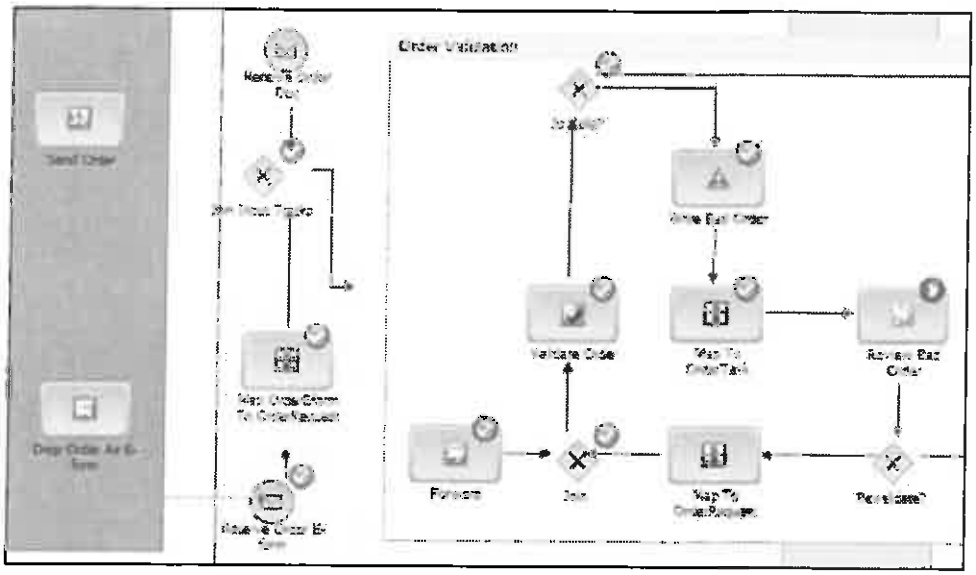


    d) Drag and drop the provided sample e-form instance **OrderEformInstance.xml** from the file explorer view on the right side into the (empty) **OrderEform** Content Repository folder on the left side.



Note: If you intend to repeat this action, use the file explorer view to rename **OrderEformInstance.xml** first (e.g. **OrderEformInstance1.xml**). Then drag and drop this e-form instance to your Content Repository folder as described above.

e) As the dropped e-form instance contains invalid order data, a new process instance should have been started and a new User Task should have been queued.
Login to My webMethods as **Administrator/manage** and navigate to **Applications** -> **Monitoring** -> **Business** -> **Process Instances** to search for the process instance. Open its Details view. Check that Event **Receive Order E-form** is the first in the trail of executed steps:



f) Open the queued User Task using the Task Management List page, provide a valid quantity and click **Revalidate** to complete the User Task as well as the process.

Check Your Understanding

1. Can you configure one Start Message Event to receive an IS Document Type and an e-form-related IS document?

2. In step 13d, why do you dragged and dropped **OrderEformInstance.xml** to the MWS public folder?

3. Instead of using a WebDAV client as above, name at least two other possibilities to start a process via an e-form when MWS is used as e-form Content Repository.

Exercise 23:
Local and Shared Metadata

Overview

In this exercise, you will use Software AG Designer to use your local metadata first by creating and running a Saved Search. Then you will publish a re-usable process to CentraSite. After publishing you are able to perform a graphical impact analysis based on the published shared metadata. Finally you will retract your shared metadata from CentraSite registry/repository.

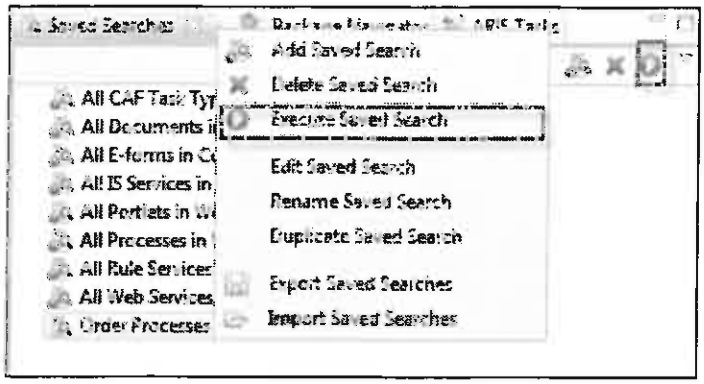
Steps

1. In this exercise, ensure that Integration Server is running as Windows service, but the Windows services **My webMethods Server** and **Optimize Analytic Engine** are stopped. *Note*: You are stopping these services to free up memory in the training VM so that you can start the CentraSite services. In a real webMethods environment you will not have to do this.
2. Launch Software AG Designer and ensure you are in the **Process Development** perspective.
3. In the webMethods Designer Preferences, under **Software AG -> Workspace Index**, ensure that **Workspace Indexing** is enabled (checkbox is unchecked). If **Workspace Indexing** was previously disabled, enable it and restart Designer.
4. Show the **Saved Searches** view (if not displayed, use **Window -> Show View -> Other... -> Software AG -> Saved Searches** to open it). Add a new Saved Search:

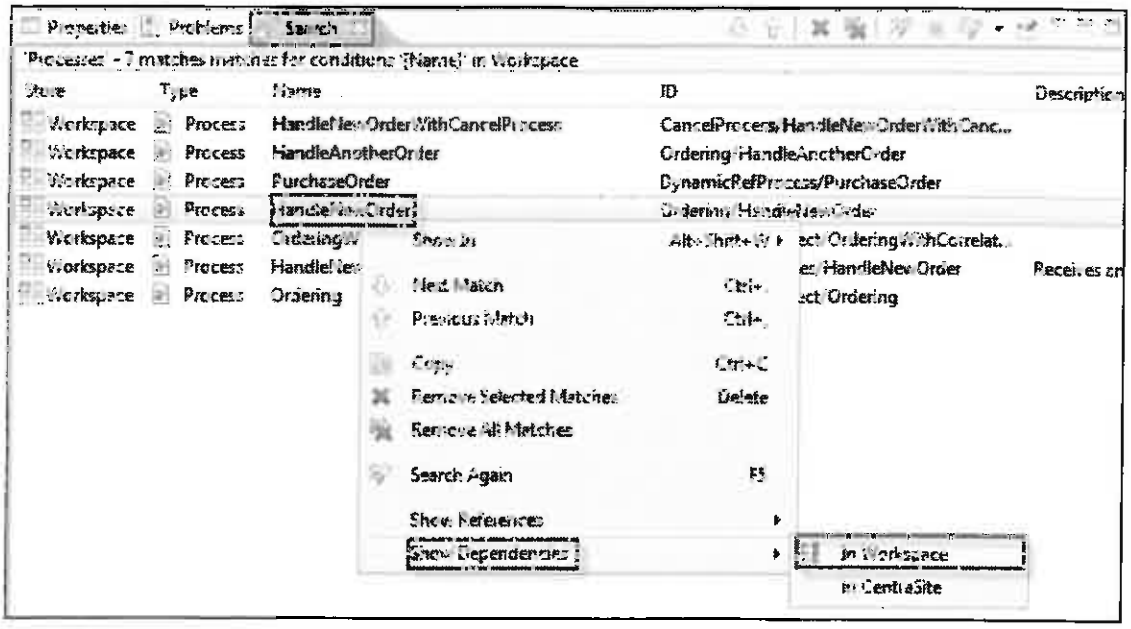
Provide the Saved Search details mentioned in the table below. Finally save your definitions:

Data	Value	
Search Name	**Order Processes**	
Asset Type	**Process**	
Properties	Property	**Name**
	Condition	**contains**
	Value(s)	**order**
Match Condition	**Any**	
Search In	**Workspace**	

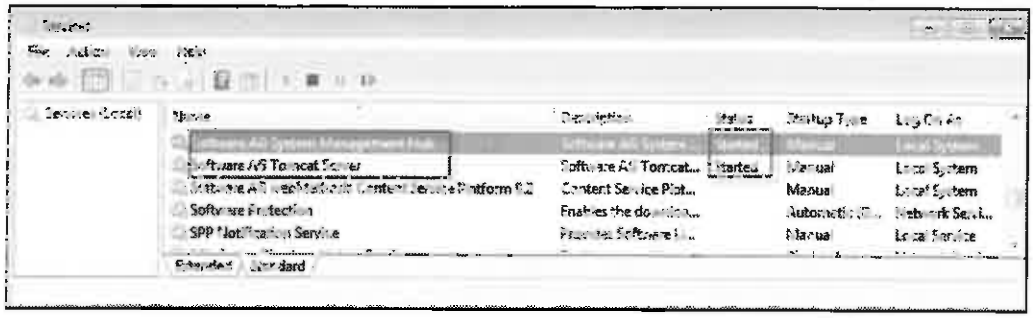
5. From the Saved Searches view, execute the **Order Processes** Saved Search.



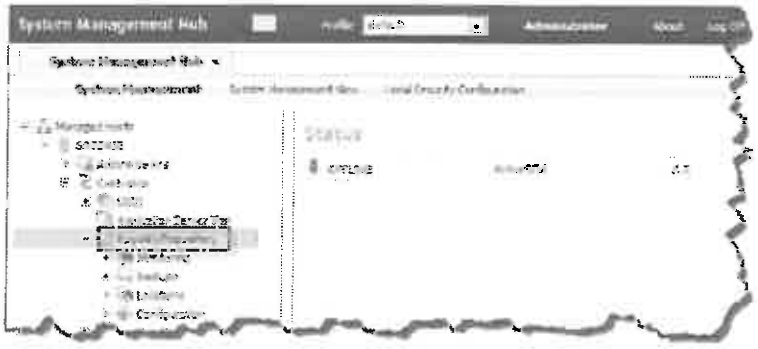
6. Inspect the resulting **Search** view. Select the match which is related to the **HandleNewOrder** process. Based on the local metadata, retrieve the process dependencies in the Workspace.



7. Start your local CentraSite to work with Shared Metadata also:
   a) Use the Windows Service Control Panel to start the following services, if not already started:
      i) CentraSite 8.2 Apache
      ii) Software AG Tomcat Server
      iii) Software AG System Management Hub



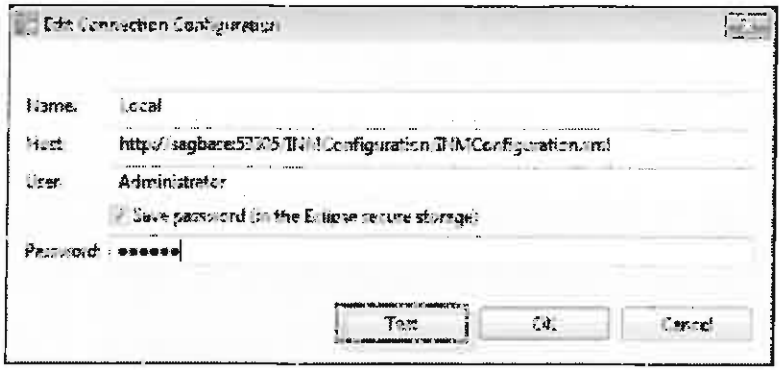
   b) Launch the **CentraSite System Administration** tool called System Management Hub (**Start -> All Programs -> Software AG -> Administration -> CentraSite System Administration 8.2**). Authenticate with **Administrator/manage**.
   c) In System Management Hub, drill down to **Managed Hosts -> SAGBASE -> CentraSite -> Registry/Repository**:

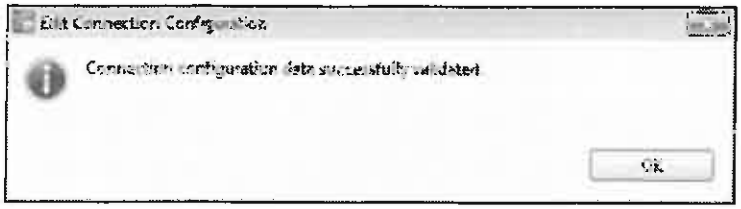


   Right-click on **Registry/Repository** and **select Start** from the context menu.

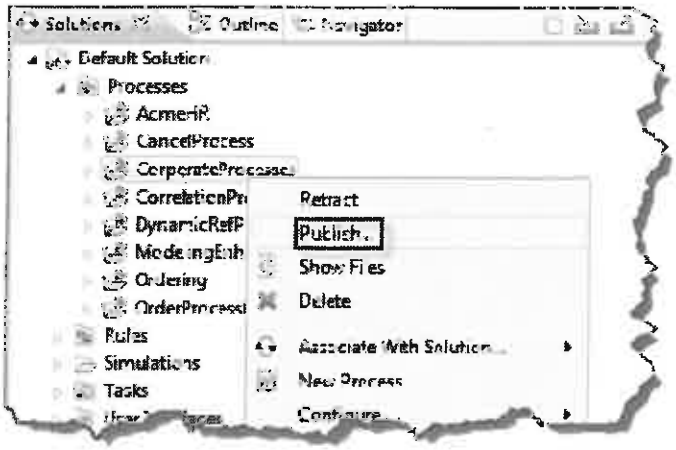
   d) Wait until the job status switches from Running to Completed, then select the **Back** button. Status should be ONLINE now:

8. Back to Software AG Designer, use **Windows -> Preferences -> CentraSite -> Connections** to check that your CentraSite connection has already been configured in your Designer environment. **Edit** the existing connection, and use **Test** to check the connectivity:

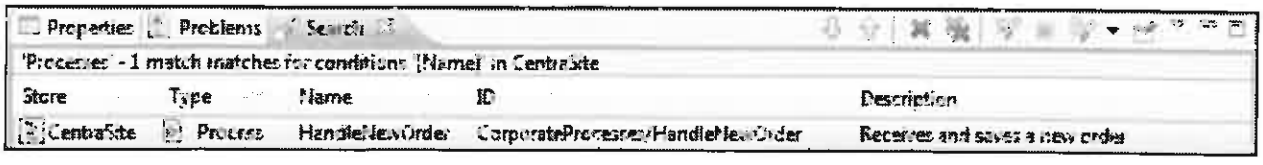




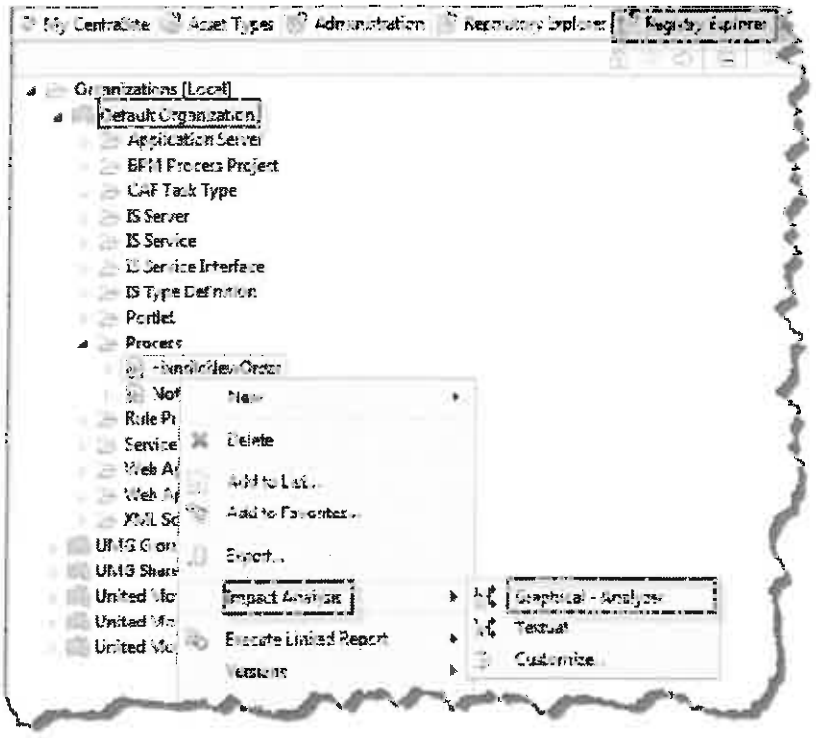
9. In the Solutions view, publish the meta-data of the process project **CorporateProcesses** to CentraSite. On the subsequent panel, publish the meta-data for all the processes.
Note: Click the **Run in Background** button, and wait for "Publish Action Completed" to appear:



10. Right-click on your Custom Saved Search **Order Processes** in the Saved Searches view. Edit the custom search to search only in **CentraSite**. Save your modifications.
11. Re-execute your **Order Processes** Custom Saved Search form the Saved Searches view.
12. In the Search view, ensure that **HandleNewOrder** is found from the **CentraSite** Store:

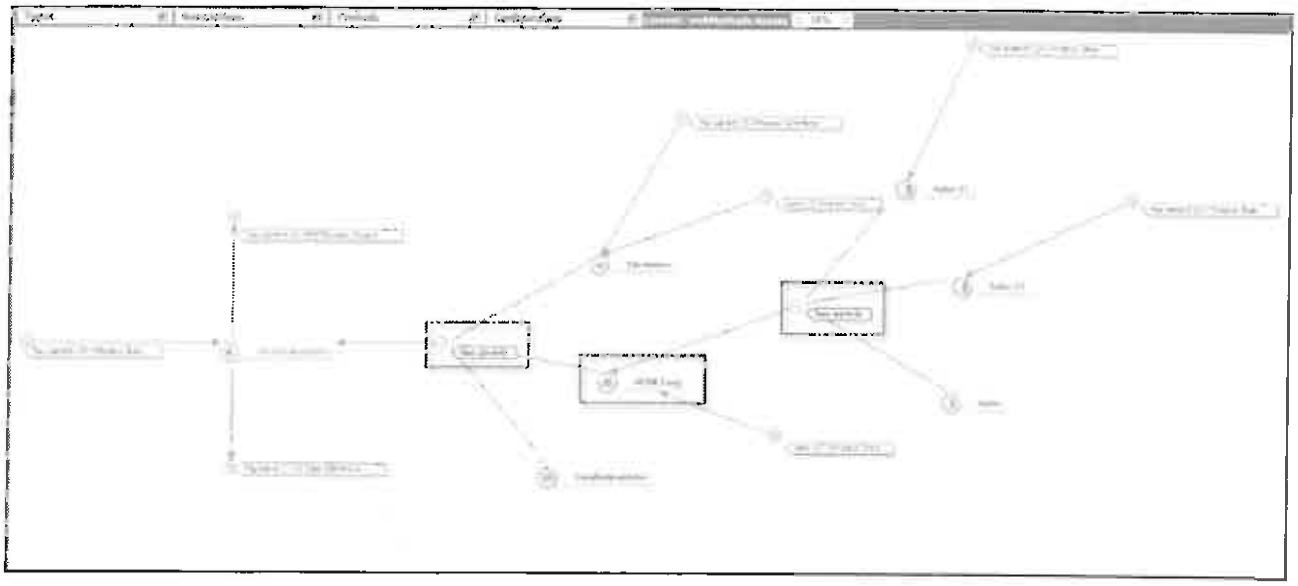


13. Switch to the **CentraSite Search and Browse** perspective. Open the Registry Explorer view.
14. In the Registry Explorer view drill down into **Organizations** -> **Default Organization** -> **Process**. Right-click on the **HandleNewOrder** process and select **Impact Analysis** -> **Graphical - Analyzer**.
    Note: Because user **Administrator** belongs to the **Default Organization** in CentraSite, your assets published from Designer to CentraSite will belong to this organization.

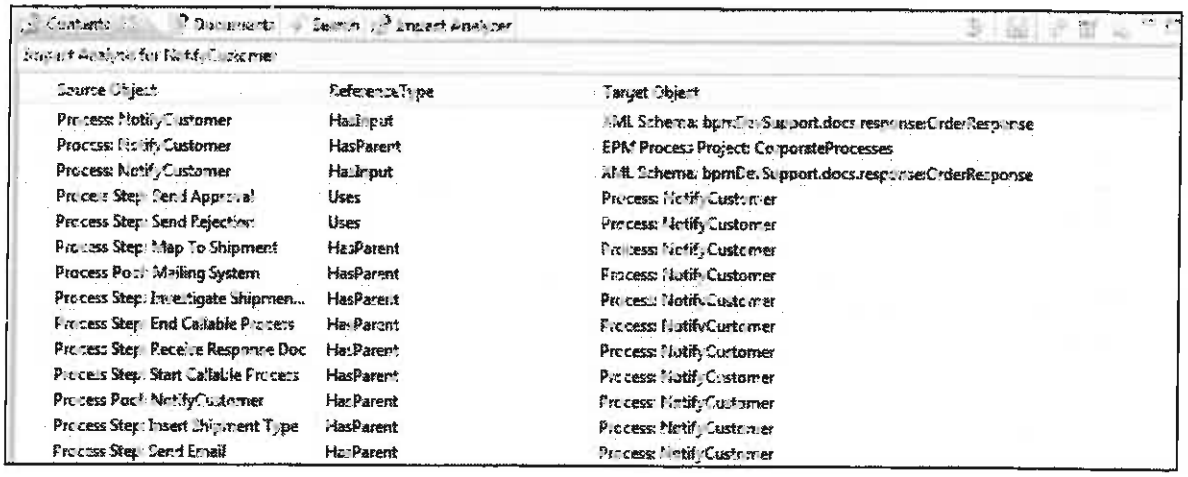


15. Maximize the **Impact Analyzer** view in Designer (double-click the view tab).
16. To filter for BPM-relevant asset types, select **webMethods Assets** from the **Configuration** dropdown list. Click **Save** to reload the data:

17. In the graphical representation, first expand **has parent (*n*) Process Pool**. To the right of ACME Corp, expand **has parent (*n*) Process Swimlane**. Continue to drill into the process. You will notice you will get a lot of detail. You can use the drop-downs at the top of the view to filter what you see.



18. Use the Registry Explorer view and select the asset related to your **NotifyCustomer** process. Choose **Impact Analysis -> Textual** from the context menu.
Inspect the results of the generated textual impact analysis in the Contents view:



Source Object	ReferenceType	Target Object
Process: Notify Customer	HasInput	XML Schema: bpmDe.Support.docs responseOrderResponse
Process: Notify Customer	HasParent	EPM Process Project: CorporateProcesses
Process: Notify Customer	HasInput	XML Schema: bpmDe.Support.docs.responseOrderResponse
Process Step: Send Approval	Uses	Process: Notify Customer
Process Step: Send Rejection	Uses	Process: Notify Customer
Process Step: Map To Shipment	HasParent	Process: Notify Customer
Process Pool: Mailing System	HasParent	Process: Notify Customer
Process Step: Investigate Shipmen...	HasParent	Process: Notify Customer
Process Step: End Callable Process	HasParent	Process: Notify Customer
Process Step: Receive Response Doc	HasParent	Process: Notify Customer
Process Step: Start Callable Process	HasParent	Process: Notify Customer
Process Pool: Notify Customer	HasParent	Process: Notify Customer
Process Step: Insert Shipment Type	HasParent	Process: Notify Customer
Process Step: Send Email	HasParent	Process: Notify Customer

19. *Housekeeping*:

a) Stop the CentraSite registry/repository.
To do so, launch the **CentraSite System Administration** tool (if not still opened) (**Start -> All Programs -> Software AG -> Administration -> CentraSite System Administration 8.2**). If necessary, authenticate with **Administrator/manage**.
In the appearing System Management Hub, drill down to **Managed Hosts -> SAGBASE -> CentraSite -> Registry/Repository**. Right-click on **Registry/Repository** and select **Stop** from the context menu.
Wait until the job status switches from Running to Completed, then select the **Back** button. Status should be **OFFLINE** now.

b) Use the Windows Service Control Panel to stop the services you started in step 7a:

i) CentraSite 8.2 Apache

ii) Software AG Tomcat Server

iii) Software AG System Management Hub

Check Your Understanding

1. In which view would you create a Custom Saved Search?

2. Why would you need to consider an asset's references and dependents?

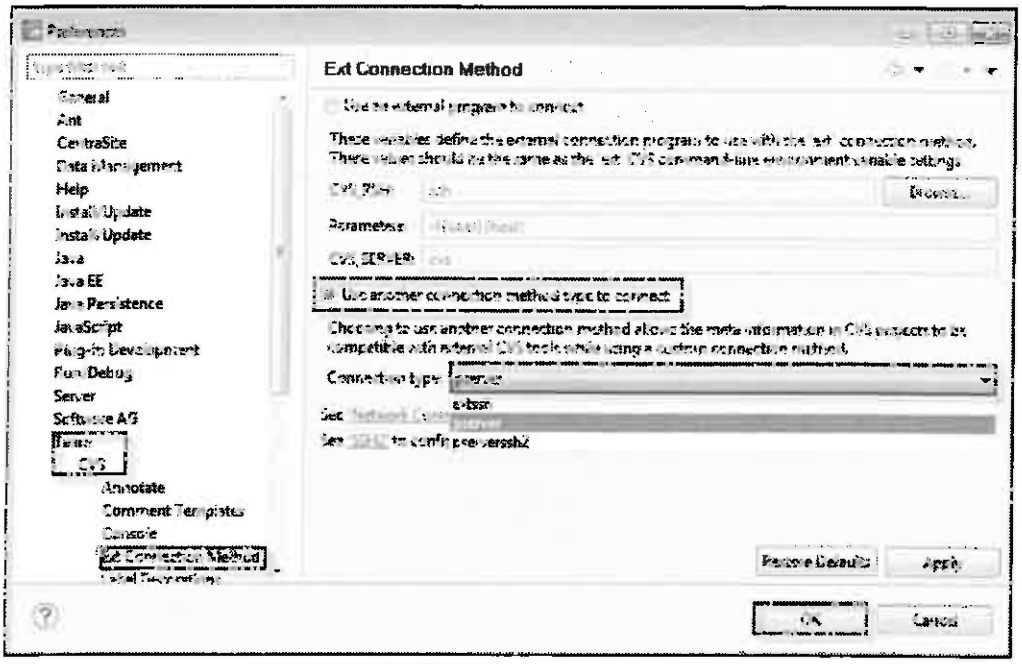
Exercise 24:
Collaborative Development Using CVS

Overview

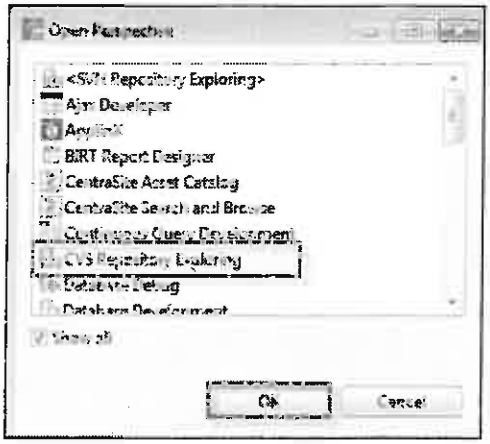
In this exercise, you will change the configuration of Software AG Designer to enable collaborative development using CVS.

Steps

1. Launch the Windows Services Control Panel. Ensure that the **CVSNT Dispatch service** and **CVSNT Locking** services are started. Start them if they are not started.
2. Launch Software AG Designer (if not already started).
3. From Designer, select **Window -> Preferences** and navigate to the **Team -> CVS** preferences.
   Note: If the **Team -> CVS** preferences are not visible in Designer, select **Window -> Open Perspective -> Other...** .Check **Show all** in the appearing dialog and select perspective **CVS Repository Exploring.** Click **OK** to allow the enablement of "CVS Support". Then select **Window -> Preferences** and navigate to the **Team -> CVS** preferences.
   Under **Ext Connection Method** check **Use another connection method type to connect** and select Connection type **pserver.** Click **OK** to save your settings.



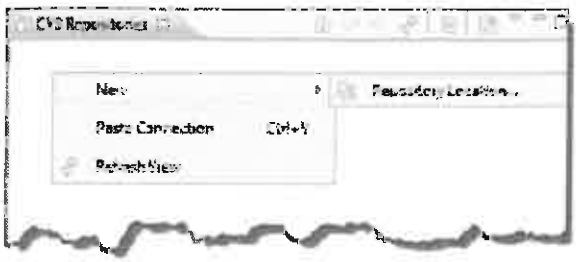
4. Open the **CVS Repository Exploring** perspective by clicking on **Window -> Open Perspective -> Other...** .

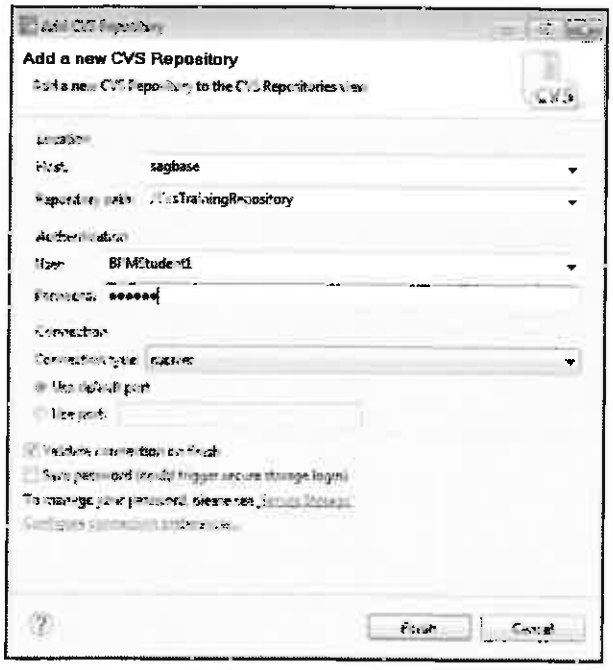


5. In the CVS Repositories view, right-click to add a new connection to the existing CVS Repository Location. For the location specify:

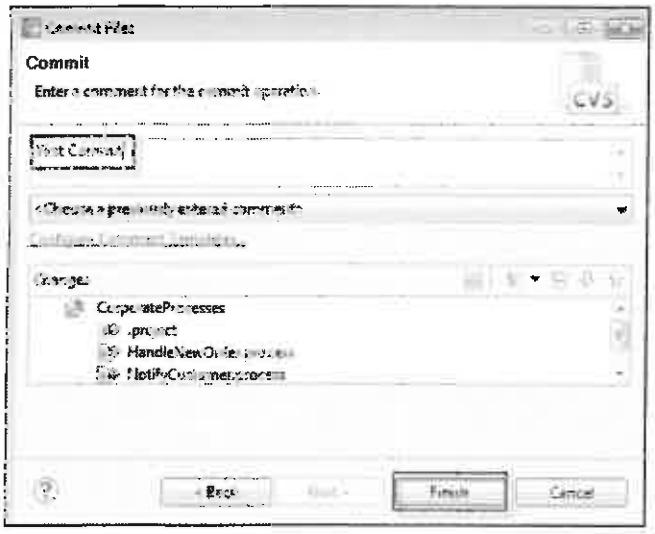
CVS Connection	Value
Host	**sagbase**
Repository path	**/CvsTrainingRepository**
User	**BPMStudent1**
Password	**manage**
Connection type	**pserver**

Accept the rest of the defaults and click **Finish**.



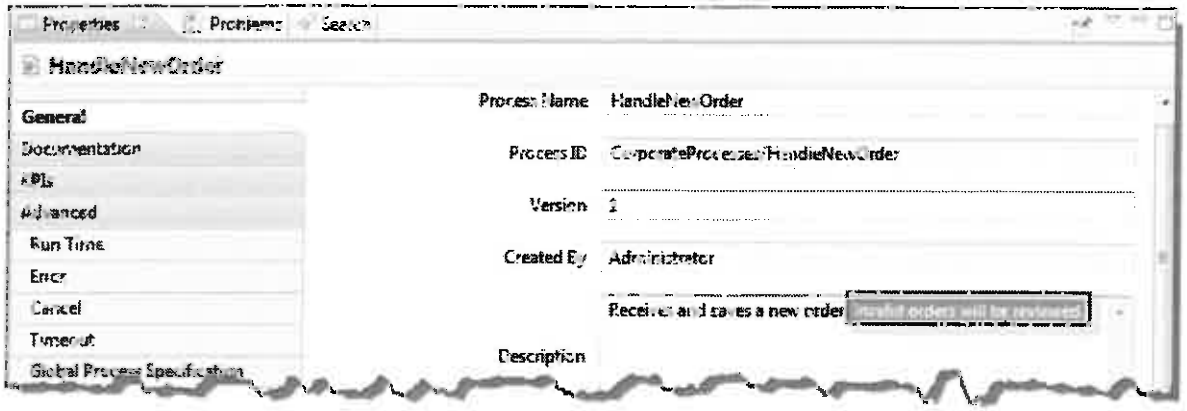


6. Switch to the **Process Development** perspective. Right-click your **CorporateProcesses** project in the **Navigator** view and select **Team -> Share Project...** from the context menu. In the appearing wizard, choose repository type **CVS**, stay on the preselected existing repository location, and accept all the other defaults until you reach the Commit Files dialog.
   On the Commit Files panel, just click **Next**, then specify **First Commit** as comment, and click **Finish** to share all contained project processes.



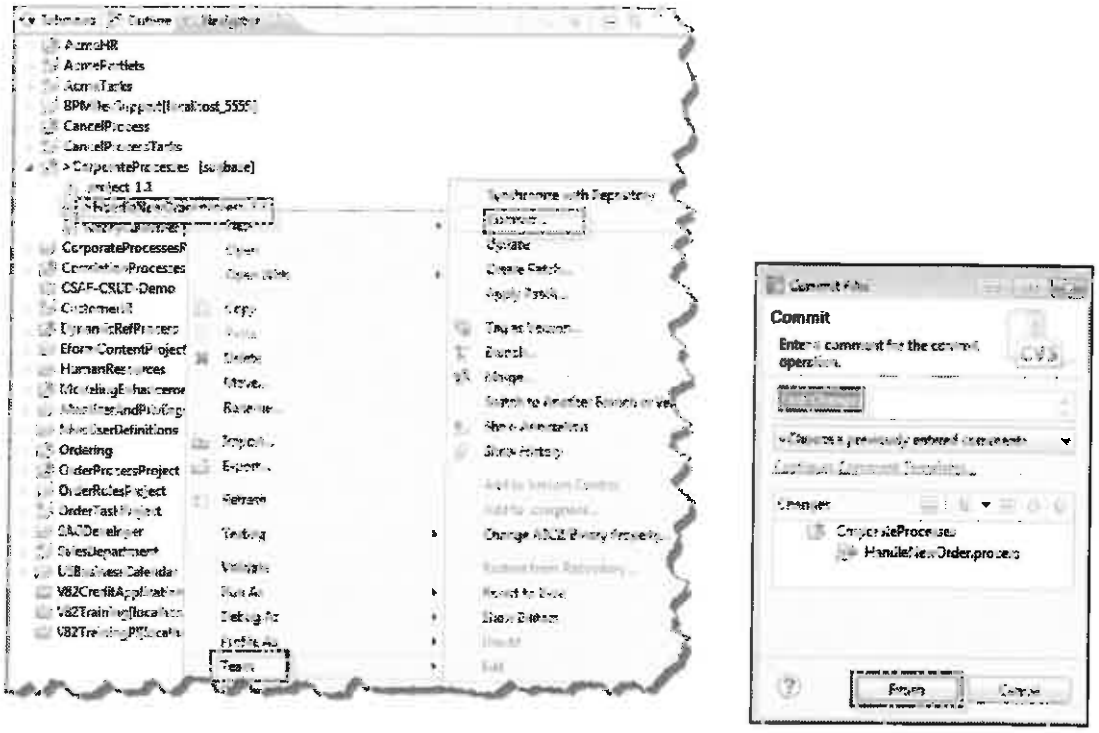
   As a result, your project assets should be flagged with a version **1.1** in the Navigator view.

7. Open the **HandleNewOrder** process from the Navigator view. Enhance the process description in the properties view like this:

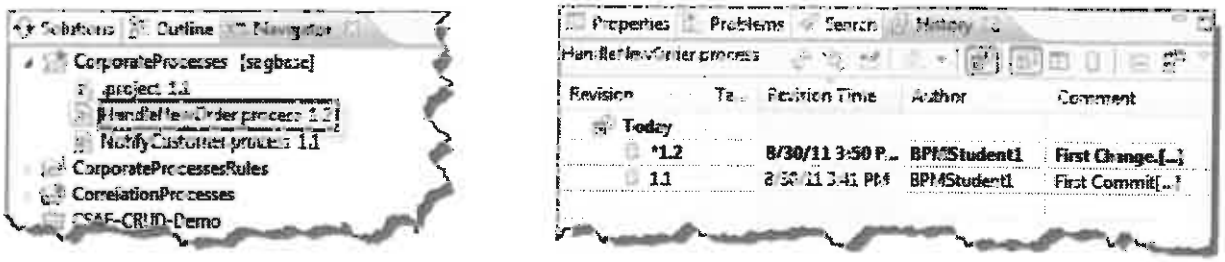


   Save your process model.

8. **HandleNewOrder** should now be prefixed in the Navigator view with an outgoing change > flag. Right-click the process in the Navigator view and choose **Team -> Synchronize with Repository** to show the differences between the latest version in the CVS Repository and the current version in the workspace.
9. Right-click the process **HandleNewOrder** in the Navigator view and choose **Team -> Commit...** to create a new version with your changed description in the CVS. Specify **First Change** for the comment and click **Finish**.



10. Double-check that the outgoing change flag vanished and the source is now of version **1.2**. Right-click the process **HandleNewOrder** in the Navigator view and choose **Team -> Show History** to bring up its source history in the History view:



Check Your Understanding

1. What benefits does CVS add to team development of BPM, CAF, and Rules projects?

2. Is CVS the only source code management system that could be used from Designer?

3. Can you use CVS for source code control of your IS packages?

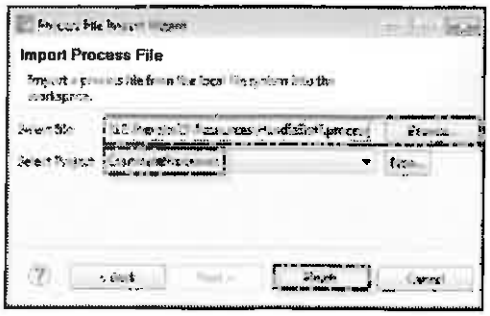
Exercise 25:
Process Versioning

Overview

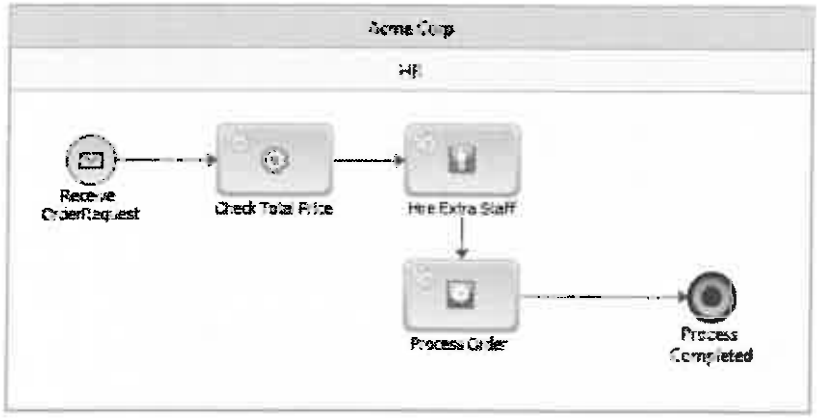
In this exercise, you will import and modify a process called **HandleStaff** to test process versioning. During your testing you will see how a process instance that is already running will continue to use a previous process version even after a new version has been released.

Steps

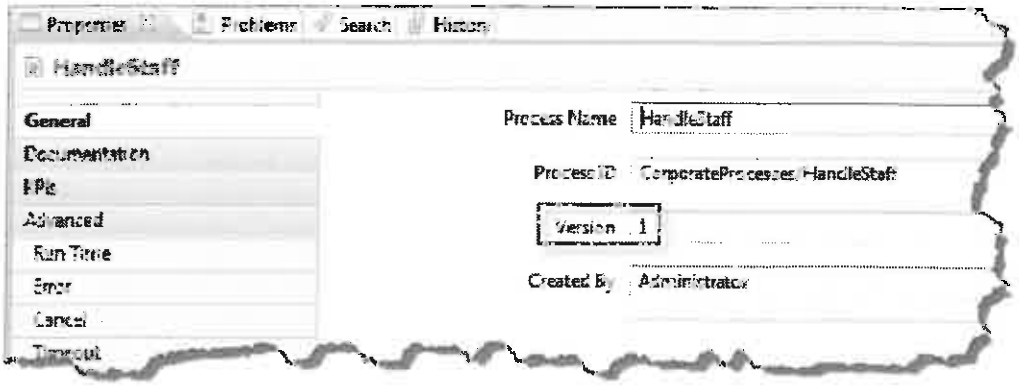
1. Ensure Integration Server, My webMethods Server, and Optimize Analytic Engine are running as Windows services.
2. Start Software AG Designer and ensure you are in the **Process Development** perspective.
3. Choose **File -> Import** from the menu bar to import a new **Process File**. In the wizard, select process file ***<workshop_dir>*\Exercise25\Resources\HandleStaff.process** and make sure Project is set to **CorporateProcesses.**



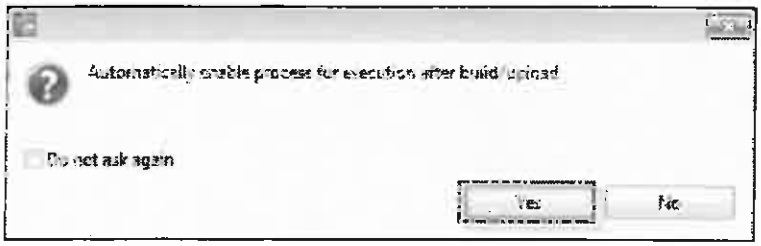
4. If not opened, open the **HandleStaff** process model in the design canvas:



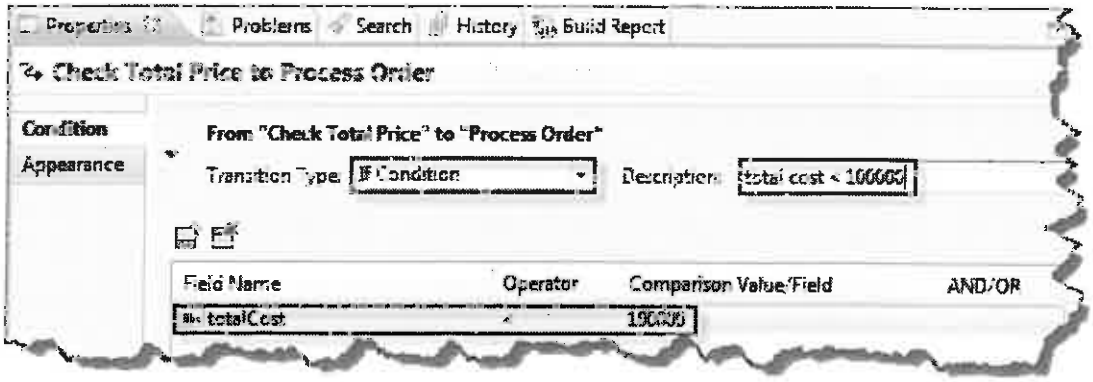
5. Double-check that the Start Message Event named **Receive OrderRequest** receives a document of type **bpmDevSupport.docs.request:OrderRequest** and step **Hire Extra Staff** invokes the IS Service **bpmDevSupport.utils:sixMinuteDelay**.
6. In the Properties view of the process model **HandleStaff**, ensure its process version is **1**:



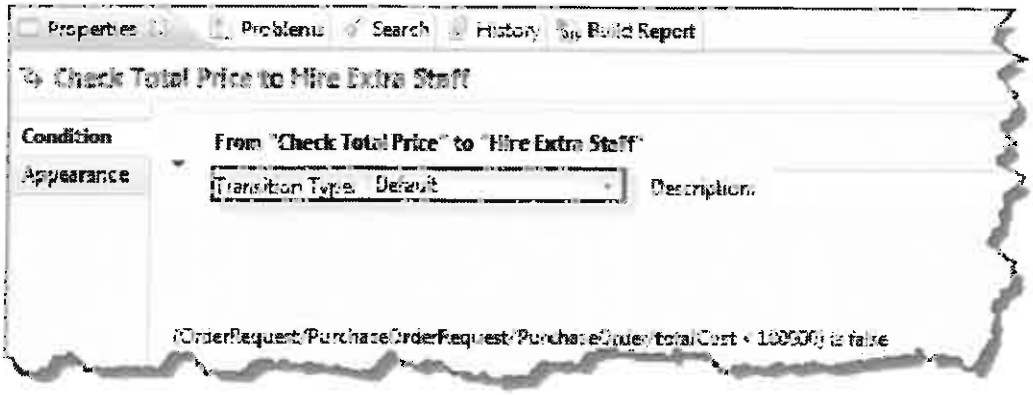
7. Build and upload the **HandleStaff** process. If asked for "Automatically enable process for execution...", confirm with **Yes**.



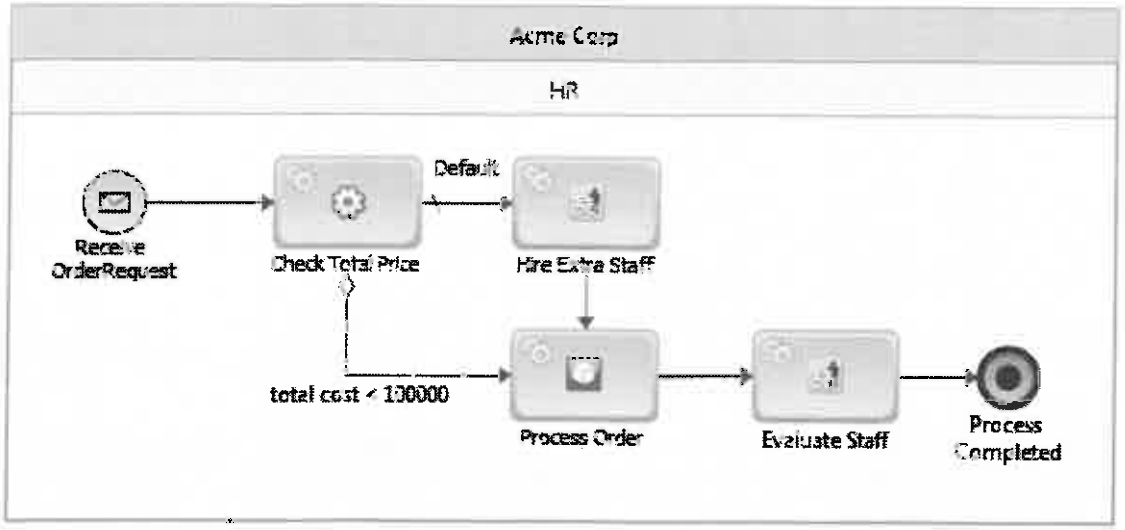
8. In the Properties view of the process model **HandleStaff**, change the process version from 1 to **2**.
9. Add a transition from step **Check Total Price** to step **Process Order** with an If Condition **OrderRequest/PurchaseOrderRequest/PurchaseOrder/totalCost < 100000**. Set the transition Description to **total cost < 100000**:



10. Change the transition type of the other transition leaving step **Check Total Price** to type **Default**.

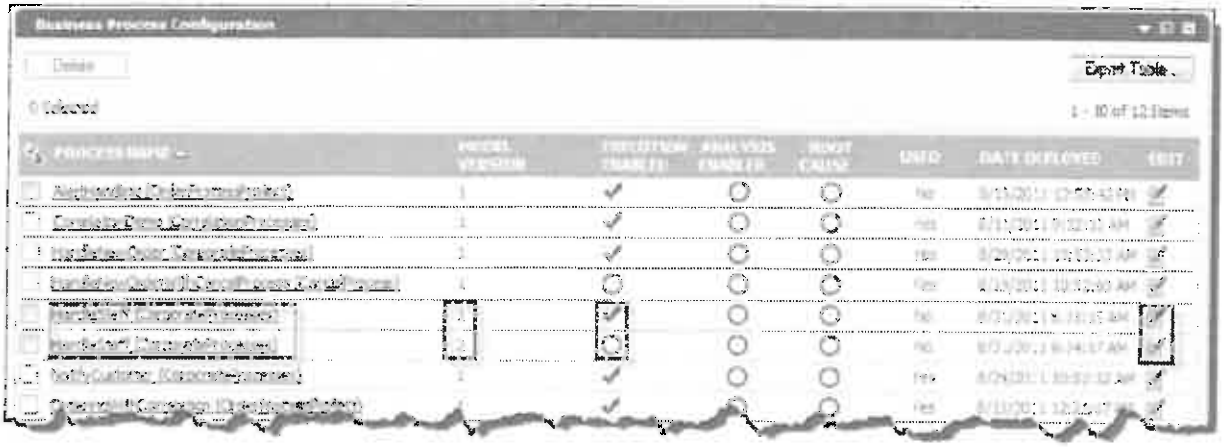


11. Remove the transition between **Process Order** and **Process Completed.**
12. Insert a Service Task Activity named **Evaluate Staff.** Add a transition from **Process Order** to **Evaluate Staff** and a transition from **Evaluate Staff** to **Process Completed.** Add an image to the **Evaluate Staff** step using the picture below as a reference. The HandleStaff process should now look like the following:

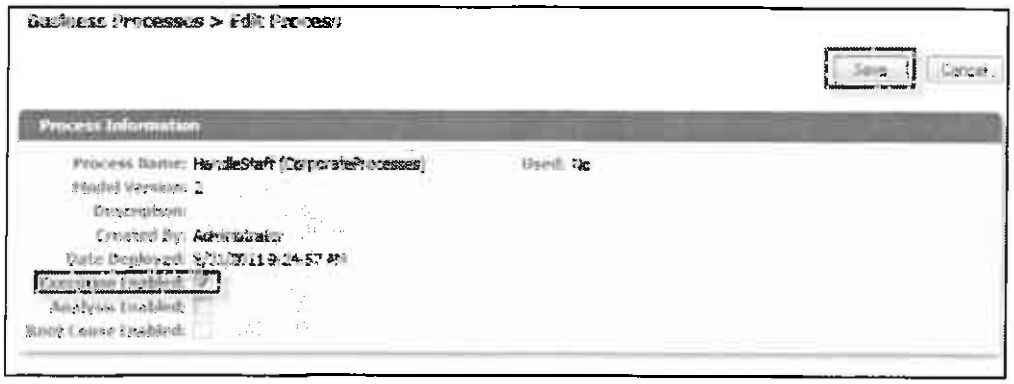


13. Using the Properties tab of the **Process Order** step, under Joins, select the join type **Unsynchronized Or.**
14. Save, build and upload the **HandleStaff** process. If asked this time for "Automatically enable process for execution..." confirm with **No.**

15. Login to My webMethods as **Administrator/manage**.
16. Navigate to **Applications** -> **Administration** -> **Business** -> **Business Processes**. Ensure both process model versions of **HandleStaff** are visible, version 1 is enabled for execution, and version 2 is currently disabled for execution (if not, click on the Edit icon to change):

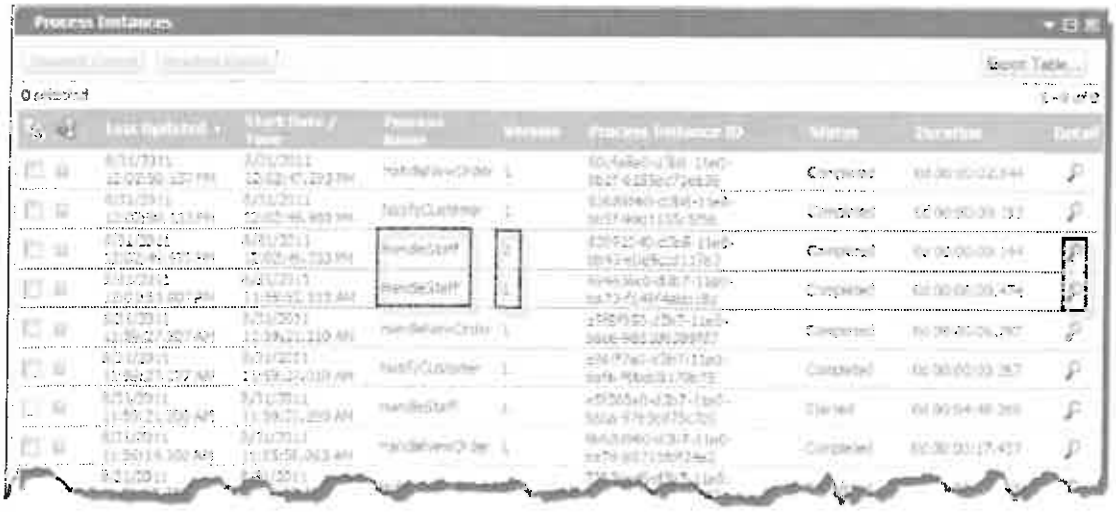


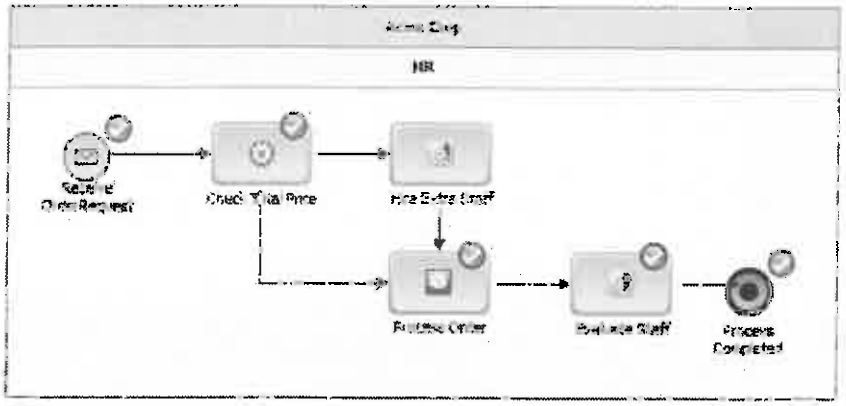
17. To start a process instance, double-click the provided form ***<workshop_dir>*\Exercise25\Resources\Ex25_submit.html**. Click the **Submit** button. If asked for IS authentication, provide **Administrator/manage**.
18. Within My webMethods, navigate to **Applications** -> **Monitoring** -> **Business** -> **Process Instances**. On the appearing page, confirm that the HandleStaff process instance is in a **Started** state.
    The process should run approximately six minutes because of step **HireExtraStaff** invoking service **bpmDevSupport.utils:sixMinuteDelay**.
19. From the **Applications** -> **Administration** -> **Business** -> **Business Processes** page, enable the second version of **HandleStaff** for execution. This will automatically disable the previous version:



If asked for upgrading running process instances, select **No**.

20. Again, double-click *<workshop_dir>*\Exercise25\Resources\Ex25_submit.html and click the **Submit** button to start a new process instance.
21. On the **Applications** -> **Monitoring** -> **Business** -> **Process Instances** page, click **Search** to refresh the list of listed process instances. You should now see two process instances of type **HandleStaff**. Ensure both used a different version number.
22. Monitor both instances until their completion and note the steps executed in each.





Check Your Understanding

1. Can instances of multiple process versions be running simultaneously?

   yes

2. Is the version set in the process properties or in step properties?

   process properties

3. Why you were asked for upgrading running process instances when enabling version 2 of your HandleStaff process model? A process was still running in version 1

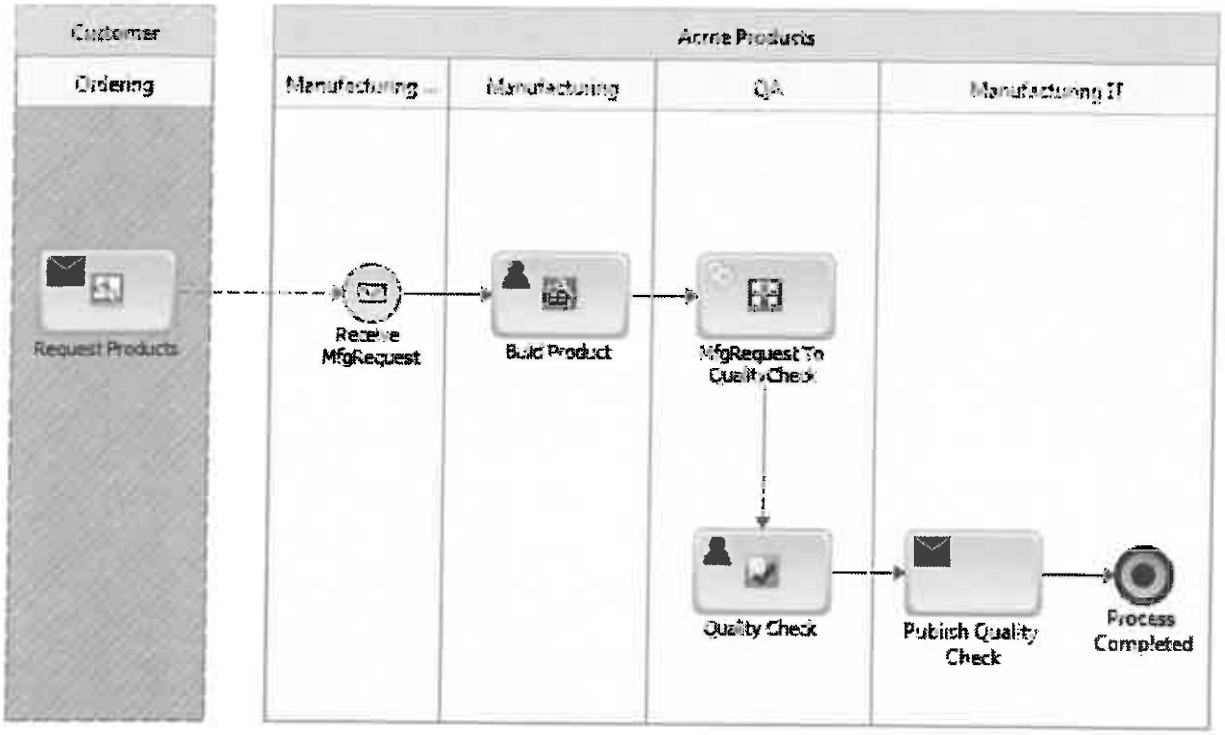
Exercise 26 *(optional)*: Process Simulation

Overview

In this exercise, you will build and examine a simulation of the Manufacturing and QA process. You will specify resources and assign them to task steps. Moreover, you will optimize the resources by using the internal optimization engine, and finally export your simulation results as an Excel report.

Steps

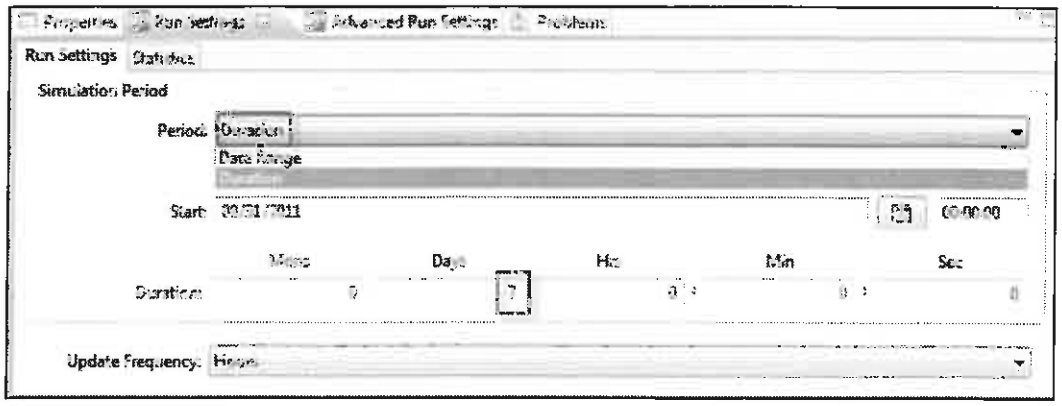
1. Ensure both Integration Server and My webMethods Server are running.
2. Launch webMethods Designer and ensure you are in the **Process Simulation** perspective.
3. Import the **MfgQATasks** CAF Project by selecting **File -> Import... -> Software AG -> Existing CAF Projects into Workspace**. Select **Archive File** and browse to the **MfgQATasks.zip** file as provided in the directory ***<workshop_dir>*\Exercise26\Resources**.
4. Also, import the process project **AcmeManufacturing** from the archive file provided as ***<workshop_dir>*\Exercise26\Resources\AcmeManufacturing.zip** by selecting **File -> Import... -> General -> Existing Projects into Workspace**.
5. Using the Solutions view, locate and open the **MfgQA** process contained in the process project **AcmeManufacturing**. If asked, stay in the Process Simulation perspective.
6. Ensure the process resembles the following figure:



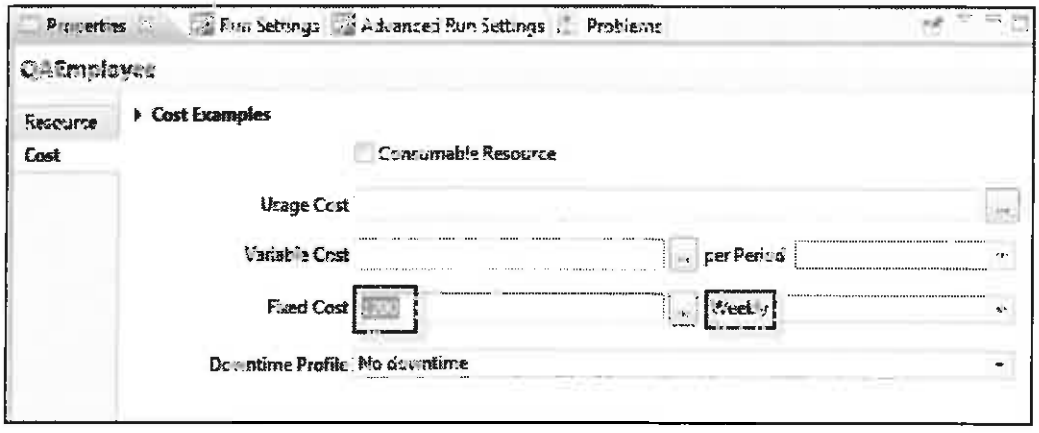
7. Click in the white space around your process model, and then use the Simulate Process button in the menu bar of Designer to create a new process simulation based on your **MfgQA** process model. In the appearing wizard accept all defaults and click **Finish**.

8. The previous step should have created and opened a simulation file called **MfgQA.simulation**. Before running the simulation the first time, you should first configure Run Settings, define and assign Resources.

   a) Ensure the **Run Settings** view is visible in Designer. Add it, if necessary (**Window -> Show View**). From the Run Settings view, set the **Simulation Period** as a **Duration** of **0 Mons** and **7 Days** starting at the *<current>* date:



   b) Ensure the Resources view is visible in Designer. Add it, if necessary (**Window -> Show View**). From the Resources view, add a resource named **QA Employee**, with a **constant** number of **3** available units at a **fixed cost** of **1200/week.**

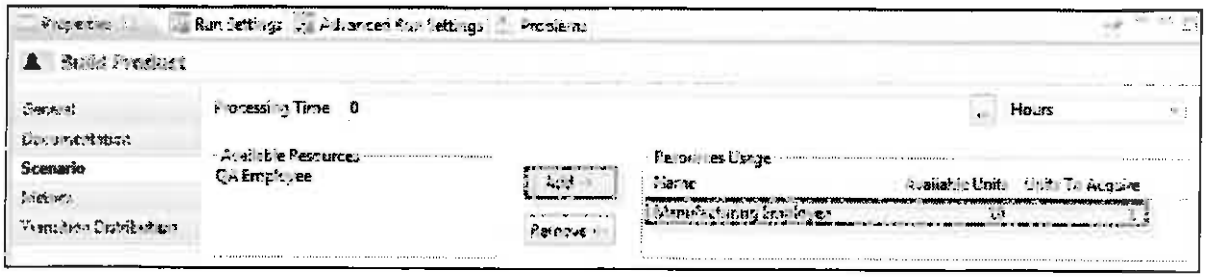


   c) Add another resource named **Manufacturing Employee**, with a **constant** number of **10** available units at a **fixed cost** of **1000/week.**

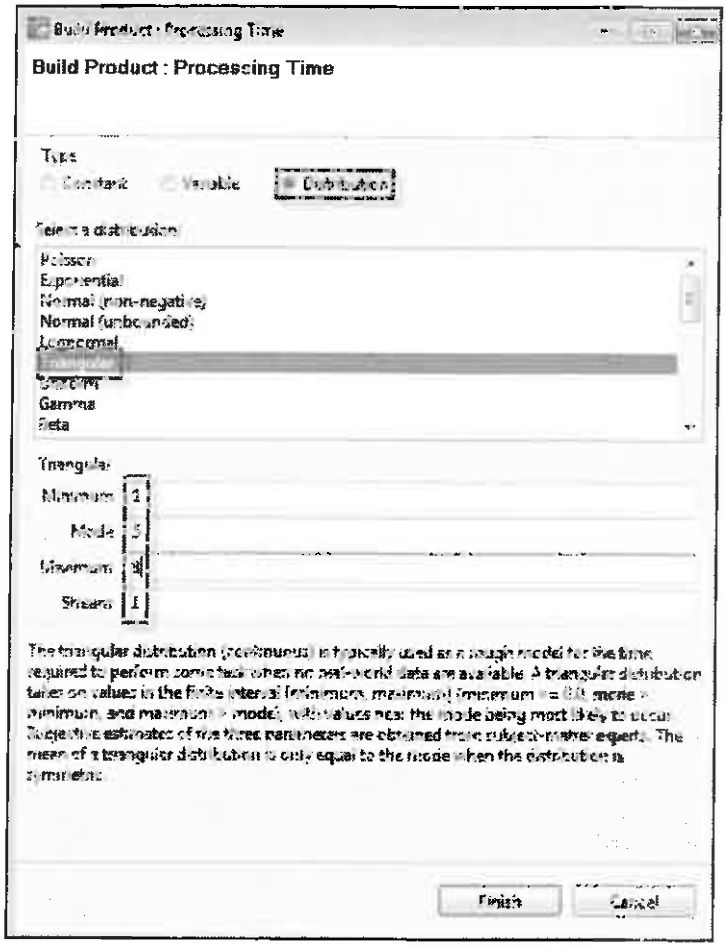
   d) Save all your work.

e) Select the User Task Activity **Build Product** in the Process Simulation editor and open the Scenario tab in the corresponding Properties view.

i) Assign **Manufacturing Employee** as the resource to be used at this User Task Activity.



ii) Click the [..] button (next to Processing Time) and set the User Task steps Processing Time to correspond to a **Distribution** type of **Triangular** with **Minimum 1, Mode 5, Maximum 8, Stream 1 hours.** This assumption will give us a more realistic data model:

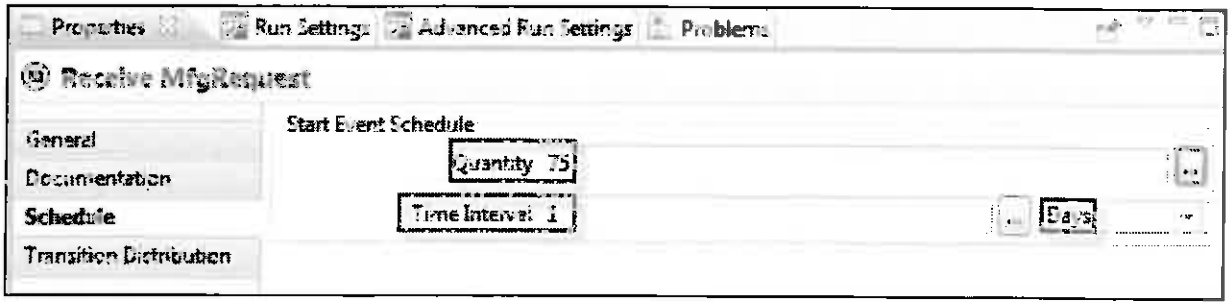


f) Select the second User Task step named **Quality Check** in the Process Simulation editor and open the Scenario tab in the corresponding Properties view.

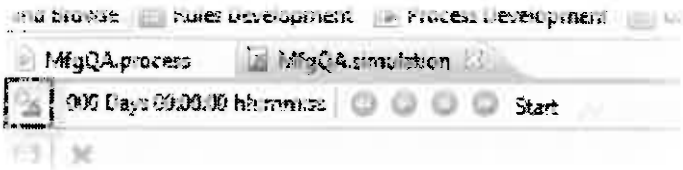
i) Assign **QA Employee** as the resource to be used.

ii) Click the [..] button to set the task steps Processing Time to correspond to a **Triangular** distribution with **Minimum 1, Mode 2.5, Maximum 3, Stream 1 hours.**

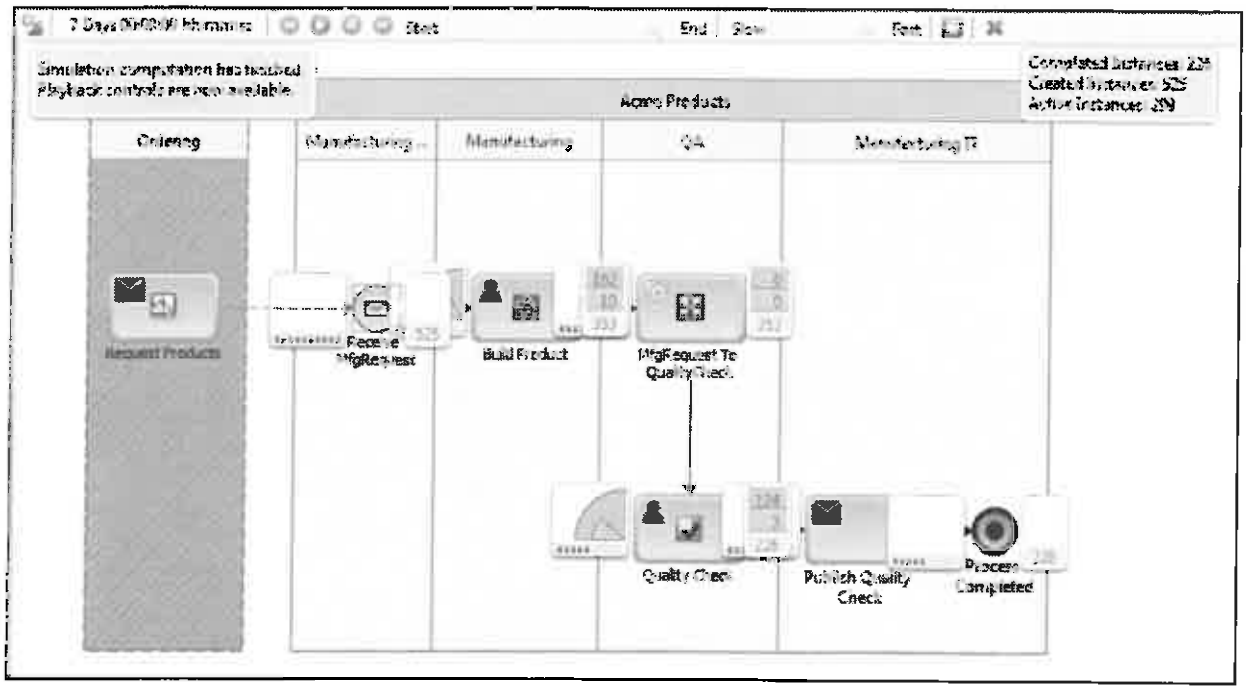
g) Select the Start Message Event **Receive MfgRequest** in the Process Simulation editor and open the Schedule tab in the corresponding Properties view. Define that **75** documents will arrive **per day:**



9. Save and run your simulation by clicking the **Simulate and animate** button:

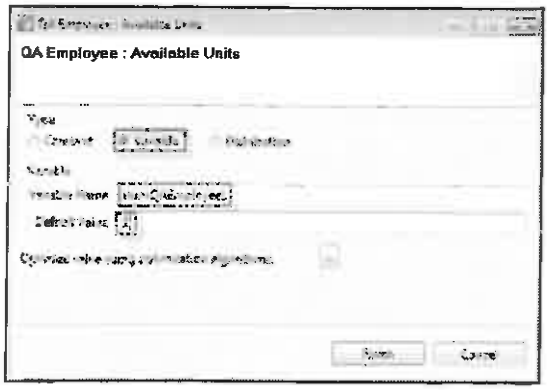


10. Look for potential bottlenecks in your simulation. The red number should always approximate zero, but may jump to 1 or 2 occasionally as your simulation runs. If the red number never goes above zero, you have too many resources. If at the end of the simulation the red number is greater than 2, you have too few:



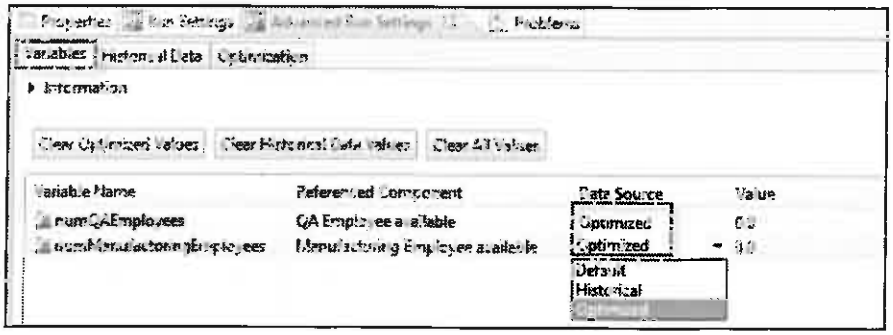
11. The simulation shows that we have a shortage of resources assigned.
Instead of guessing a reasonable number of required resources to avoid bottlenecks, we will calculate their number by using the internal Optimization Engine:

a) Select resource **QA Employee** from the Resources view and open the corresponding Properties view. On the Resource tab, click on the ... button and change the resource to be of type **Variable** with a **Default Value** of **1**. Specify **numQAEmployees** as variable name:



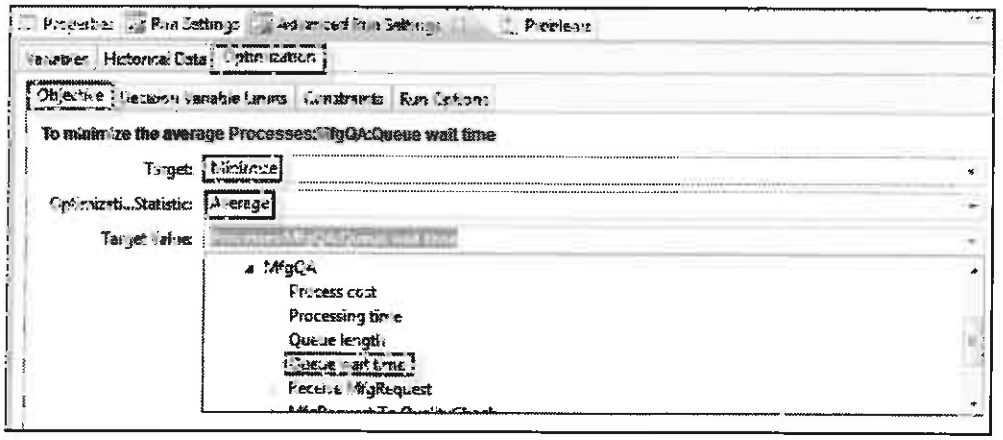
b) Select resource **Manufacturing Employee** from the Resources view and open the corresponding Properties view. On the **Resource** tab, click on the ... button and change the resource to be of type **Variable** with a **Default Value** of **1**.
Specify **numManufactoringEmployees** as variable name.

c) Open the **Advanced Run Settings** view and open the **Variables** tab. For both variable names displayed here, change **Data Source** to a value **Optimized**:

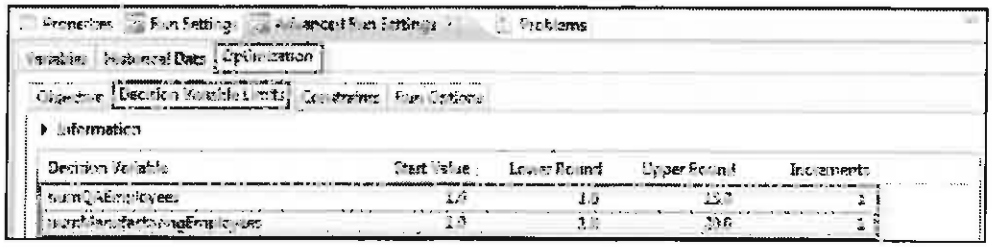


d) In the **Advanced Run Settings** view, switch to the **Optimization** tab.

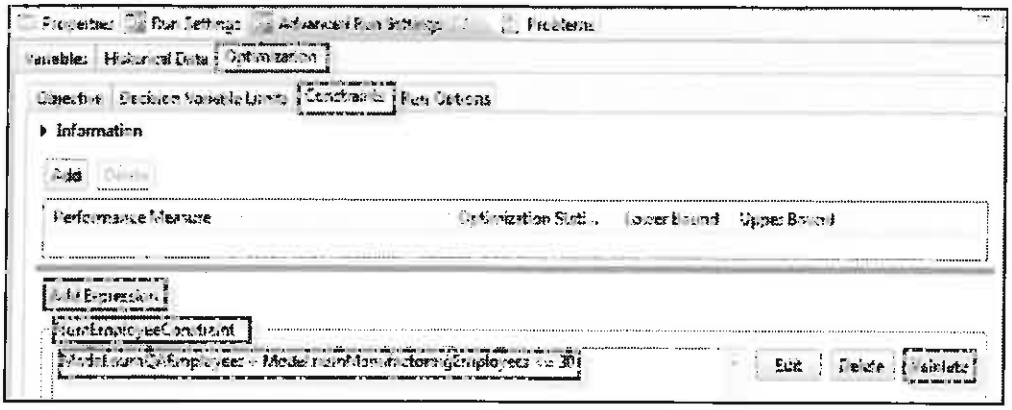
i) On the **Objective** sub-tab specify Target **Minimize**, Optimization Statistic **Average**, and select **Processes:MfgQA:Queue wait time** as Target Value to be minimized:



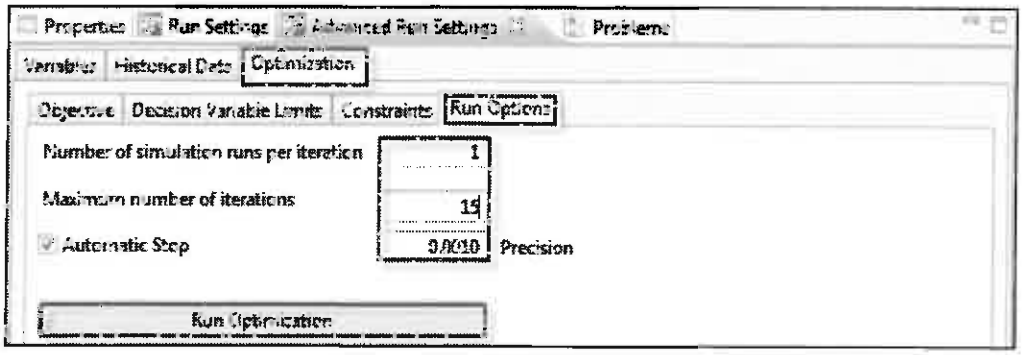
ii) On the **Decision Variable Limits** sub-tab provide the following boundaries:



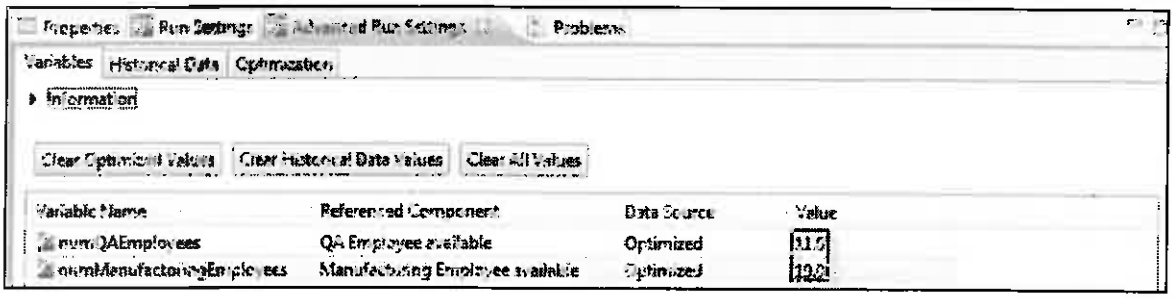
iii) On the **Constraints** sub-tab click **Add Expression** to create a constraint by using the offered Expression Builder. The Constraint Name should be **NumEmployeeConstraint** and should limit the total number of resources to a value less than or equal 30. **Validate** your expression.



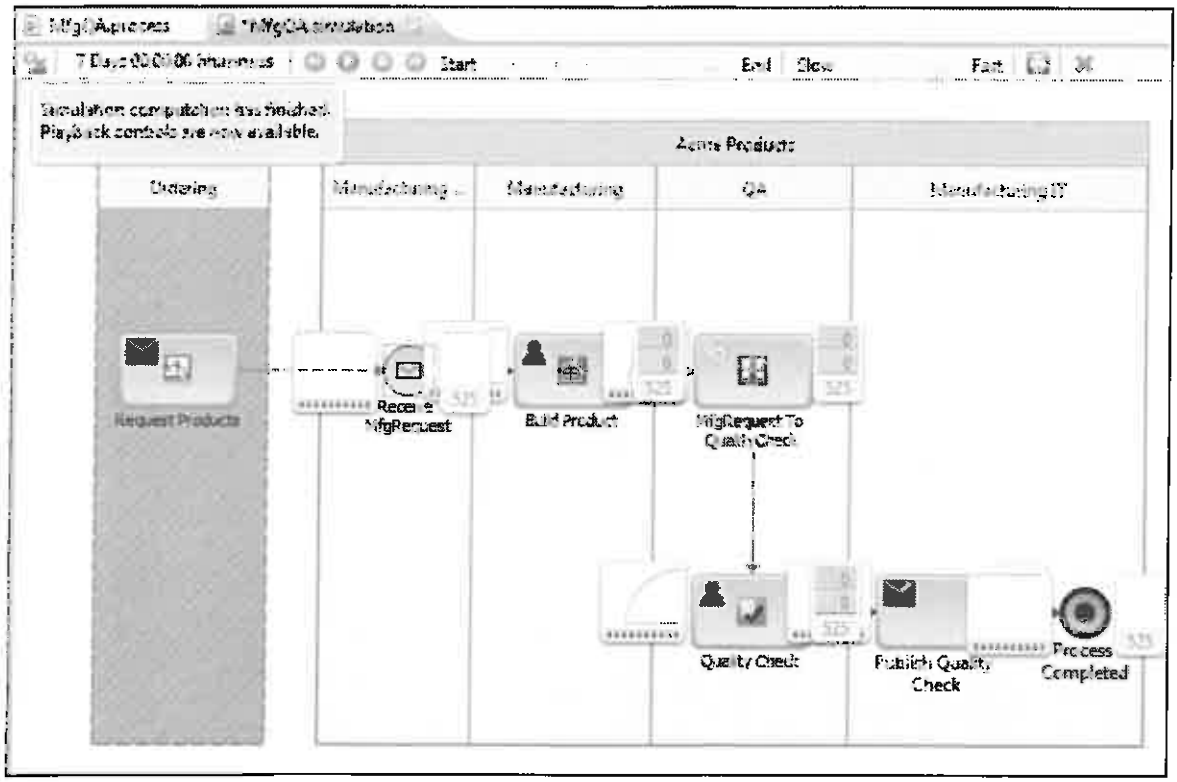
iv) On the final **Run Options** sub-tab specify the Run Options as displayed below and click on **Run Optimization**. If prompted for saving your changes, click **Yes.**



v) Optimization should calculate best fitting values for both resource variables. Monitor the optimization progress displayed in the Optimize Statistics view.
Finally, optimized results are visible on the Variables tab of the Advanced Run Settings view and are stored in each resource definition.

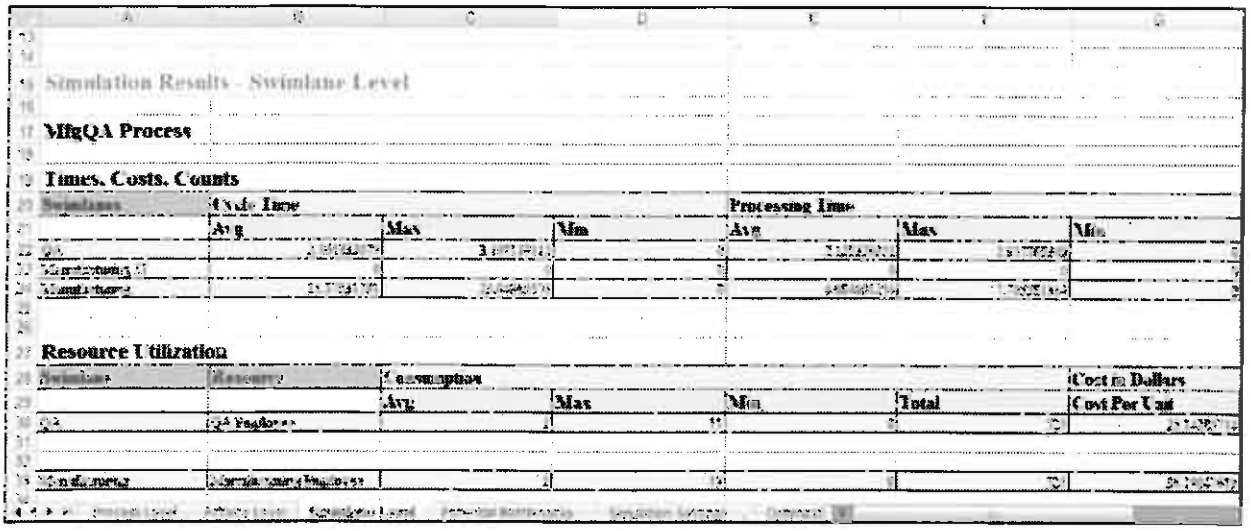


12. Save and rerun your simulation by clicking the **Simulate and animate** button. Look again for bottlenecks:



13. Generate and save a report in Excel format from your simulation. Specify **C:\temp** as target folder and provide **MfgQA_SimulationReport** as report name.

14. Open the saved report using Microsoft Excel Viewer. Locate the total cost of your simulation on a per-swimlane basis.



Check Your Understanding

1. Why did we create two resources in the Resources view?

2. Could you change the flow of the model in the Simulation view?

Appendix:
Help Information

When You Get Lost...

If you are unable to complete any of the exercises, you may use the following instructions to import the needed projects from the Solution folder of the exercise (***<workshop_dir>*\Exercise##\ Solution**). You need to delete your existing projects first, and then import the CAF/Task project, followed by the Process and/or Business Rules project.

To delete a project from workspace:

1. In the Navigator view, right-click the project name and click **Close Project.**
2. *CAF/Task projects only*:
   If a CAF or Task project is published to the MWS, you will need to right-click the **My webMethods Server** in the Servers view and click **Add and Remove Projects.**
   Select the project (e.g. SalesDepartment or CustomerUI), select **Remove**, and click **Finish.**
3. In the Navigator view, right-click the project name and select **Delete.**
4. Select the option **Also delete contents under...** . Click **Yes.**

To import a CAF project into a workspace (e.g. SalesDepartment.zip):

1. In the File menu, click **Import.**
2. In the Import window, click **Software AG -> Existing CAF Projects into workspace.** Click **Next.**
3. Enable **Select archive file** and browse to the Exercise's setup folder (***<workshop_dir>*\Exercise##\SetupExercise**)
4. Select the archived file of the CAF project and click **Open.** Click **Finish.**
5. Right-click the **My webMethods Server** in the Servers view and click **Add and Remove Projects.**
   Select the CAF project (e.g. SalesDepartment), select **Add>**, and click **Finish.**

To import a Process or Business Rules project into a workspace (e.g. CorporateProcesses.zip):

1. In the File menu, click **Import.**
2. In the Import window, click **General -> Existing Projects into workspace.** Click **Next.**
3. Enable **Select archive file** and browse to the exercise's setup folder (***<workshop_dir>*\Exercise##\SetupExercise**)
4. Select the archived file of the Process/Business Rules project and click **Open.** Click **Finish.**
5. *Process projects only*: Build and Upload the desired process(es).
   Business Rules projects only: Export the Rules project to an IS and optionally to a MWS content repository.