

IBM WEBSHERE 6.0 – LAB EXERCISE

Preparing a Process with WebSphere Business Modeler for WebSphere Integration Developer

What this exercise is about	1
Lab Requirements.....	1
What you should be able to do	1
Introduction	2
Exercise Instructions	2
Part 1: Initialize workspace for WebSphere Business Modeler	3
Part 2: Prepare Process for Export	9
Part 3: Initialize workspace for WebSphere Integration Developer	21
Part 4: Rewire Process for Components.....	23
What you did in this exercise	35
Solution Instructions.....	36

What this exercise is about

The objective of this lab is to provide you the basic steps for exporting a project from WebSphere Business Modeler V6. This lab will also guide you on how to import a project into WebSphere Integration Developer V6 and rewire the process in a best practice way to use the imports/exports.

Lab Requirements

- WebSphere Business Modeler V6
- WebSphere Integration Developer V6.0.1
- Sample code in the directory C:\

What you should be able to do

At the end of this lab you should be able to:

- Import a project into WebSphere Business Modeler V6.
- Set Implementation type to Java for the tasks in WebSphere Process Server modeling mode.
- Add customize information to the process.
- Export the project from modeler.

- Import a model into WebSphere Integration Developer V6.0.1.
- Create modules using library dependency.
- Create implementations and bindings in modules.
- Rewire the process to use imports and exports.

Introduction

In this lab exercise you will learn how to import a project into WebSphere Business Modeler V6. This lab shows you the procedures of setting the implementation type to Java for a particular task in the WebSphere Process Server modeling mode, adding customization information to the process, and exporting the project from WebSphere Business Modeler V6. When exported, the process is converted to a BPEL business process. The lab exercise will then show you how to import a project into WebSphere Integration Developer V6.0.1, create modules using library dependencies, create implementation and binding, and rewire the process to use imports/exports. The rewiring of the process to use imports/exports rather than components is a best practice which will be beneficial when it comes to modifying the process in WebSphere Business Modeler and then importing it again into WebSphere Integration Developer. With the use of the Imports/Exports, changes to the individual task implementations do not have to be modified when the process is updated.

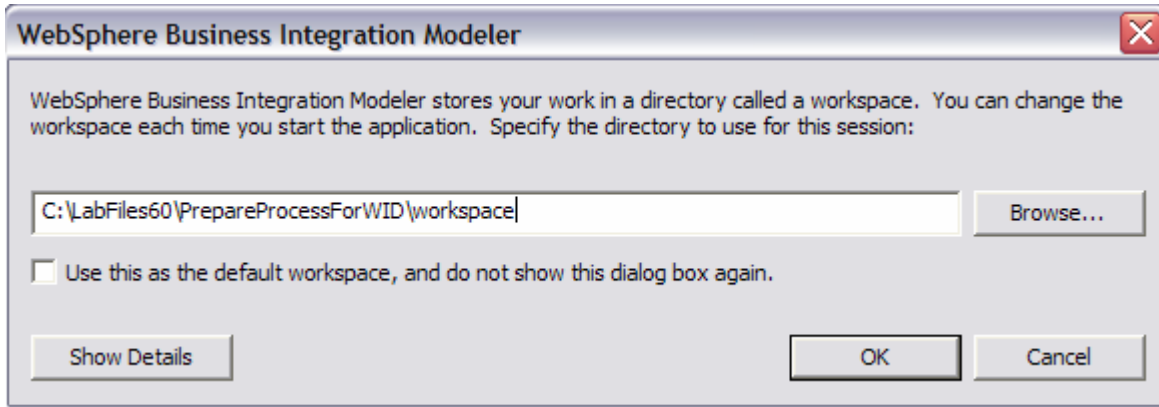
Exercise Instructions

Some instructions in this lab may be Windows operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to execute the appropriate commands, and use appropriate files (.sh vs. .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

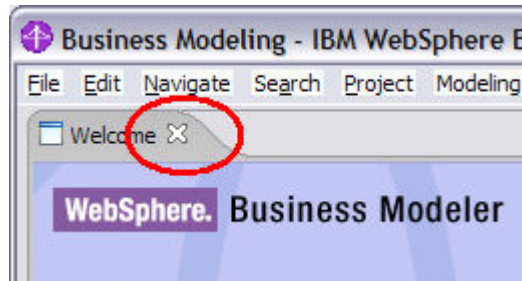
Reference Variable	Windows Location	AIX/UNIX Location
<LAB_FILES>	C:\Labfiles60	/tmp/Labfiles60

Part 1: Initialize workspace for WebSphere Business Modeler

- ___ 1. Start WebSphere Business Modeler.
 - ___ a. To start the Modeler, click **Start > Programs > WebSphere Business Modeler > WebSphere Business Modeler**.
 - ___ b. The Modeler workspace window will appear. Click the **Browse...** button and select **<LAB_FILES>\PrepareProcessForWID\workspace** as your workspace directory.

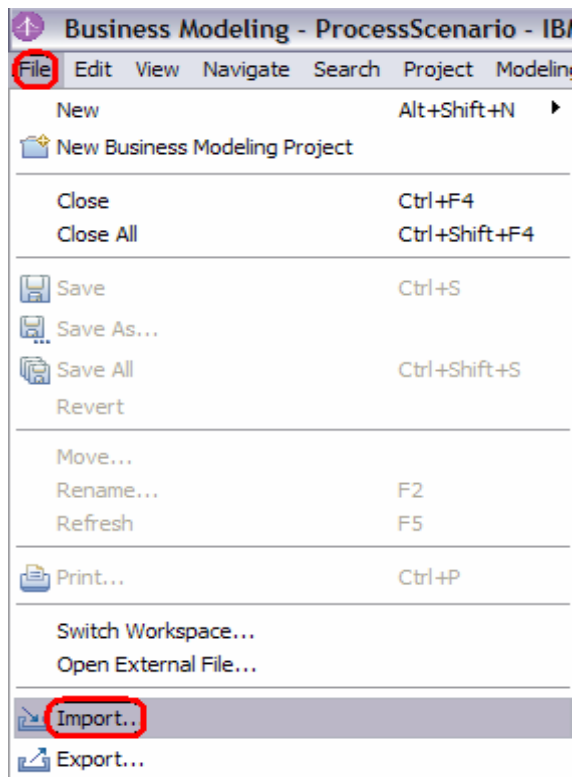


- ___ c. Click **OK**. If the Quickstart wizard opens, click **Cancel**.
- ___ d. Close the Welcome view by clicking the **X** at the top of the view.

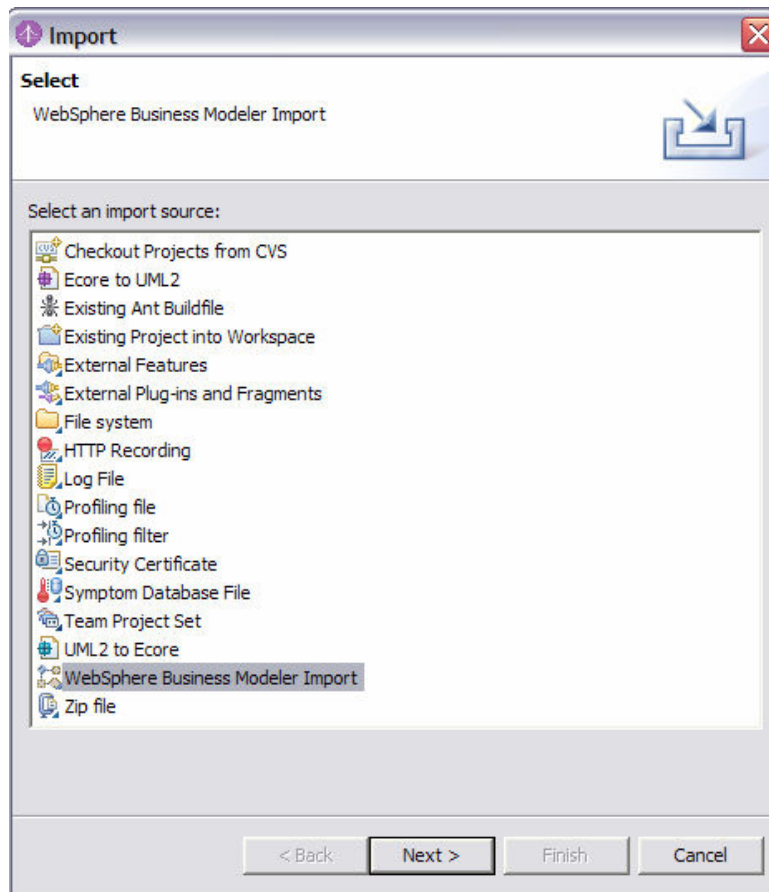


___ 4. Import Solution.

___ a. Select **File > Import**.

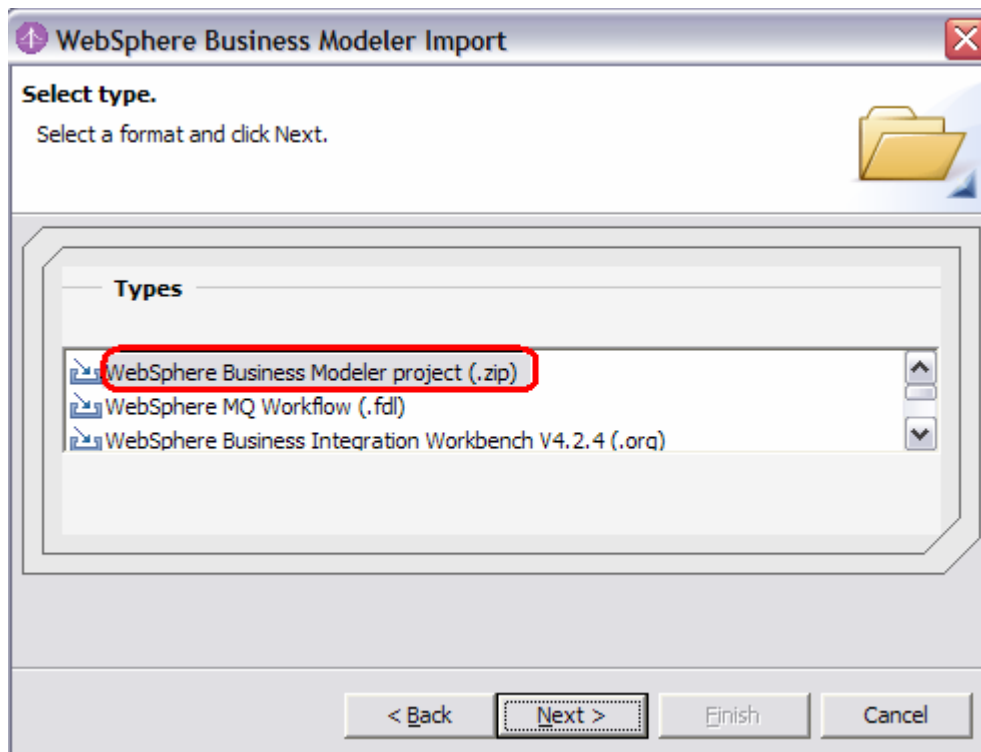


___ b. From the **Import** window, select **WebSphere Business Modeler Import** and click **Next**.



WebSphere Business Modeler Import window will be opened.

__ c. Select **WebSphere Business Modeler Project (.zip)** and click **Next**.



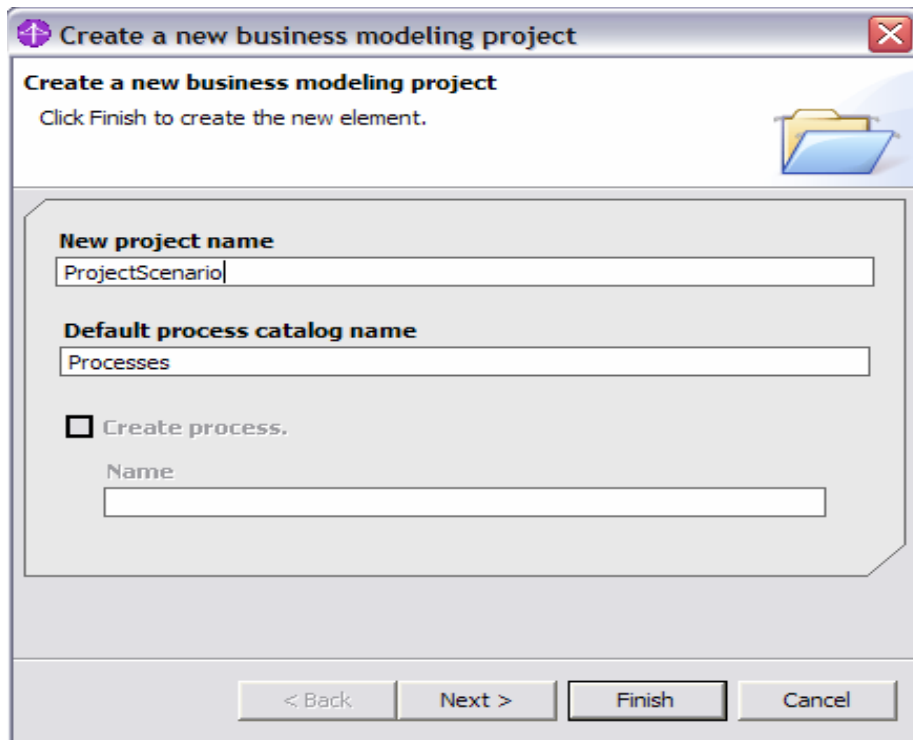
__ d. Click **Browse** and select **<LAB_FILES>\PrepareProcessForWID\import** as the **Source directory**.

__ e. Select the file **ProjectScenario_start.zip**.

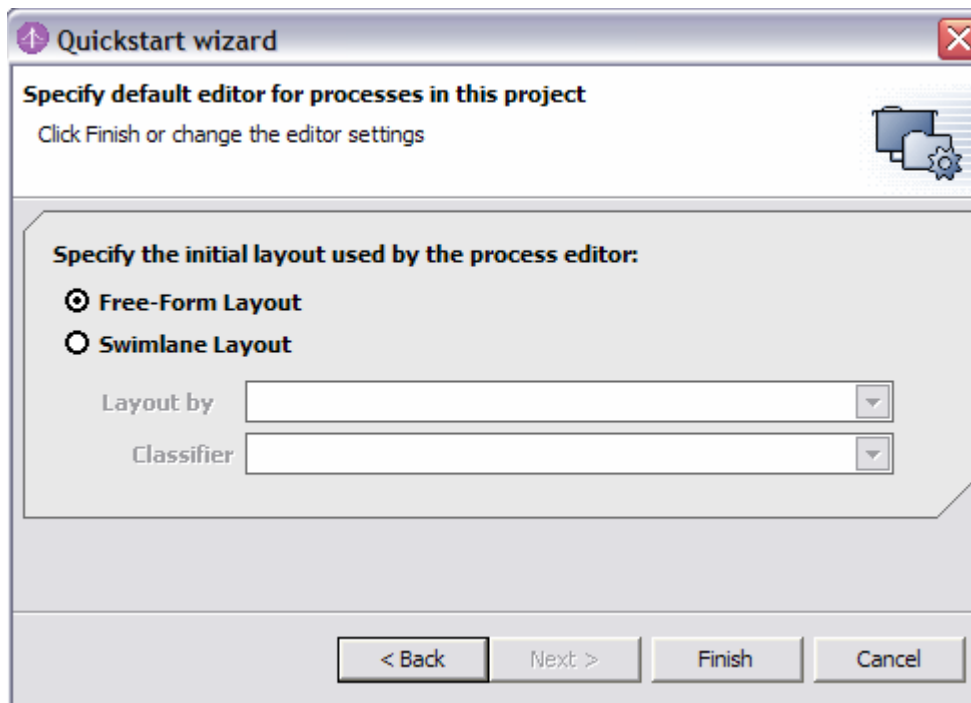
__ f. Click **New** to create a **Target Project**.

Create a new business modeling project window will be opened.

___ g. Enter **ProjectScenario** as **New project name** and click **Next**.

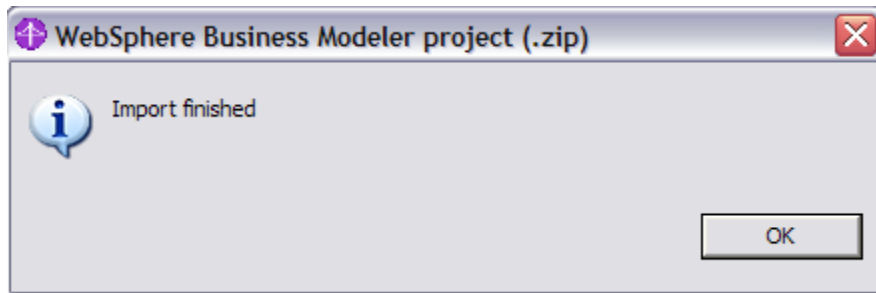


___ h. Accept the default **Free-Form Layout** and click **Finish**.



___ i. Click **Finish** from the **WebSphere Business Modeler Import** window.

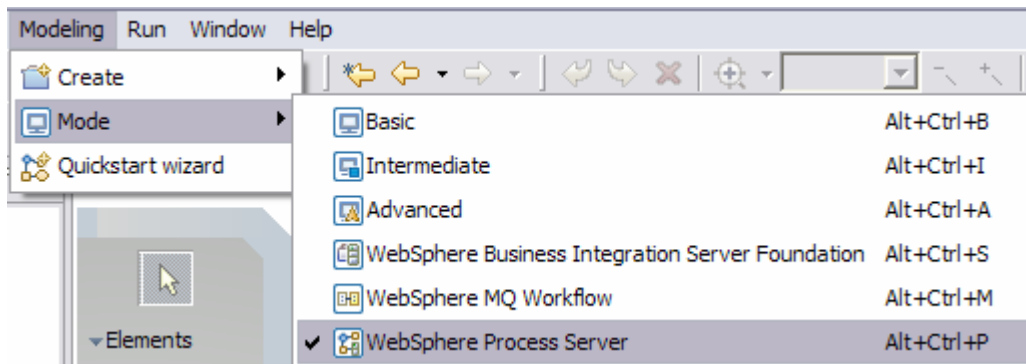
___ j. Click **OK** from **WebSphere Business Modeler project (.zip)** window.



Part 2: Prepare Process for Export

In this section, you will perform the steps necessary to prepare the process for assembly in WebSphere Integration Developer and deployment to WebSphere Process Server.

1. WebSphere Business Modeler provides a number of modeling modes which correspond to different destination runtimes. Each mode provides and restricts different options depending on what is supported by the runtime. Since the ProcessScenario process is destined for WebSphere Process Server, change to this mode accordingly.
 - a. Select the WebSphere Process Server modeling mode by selecting **Modeling > Mode > WebSphere Process Server** from the main menu.



You may notice some changes as different options are displayed or removed inline with the support available for WebSphere Process Server.

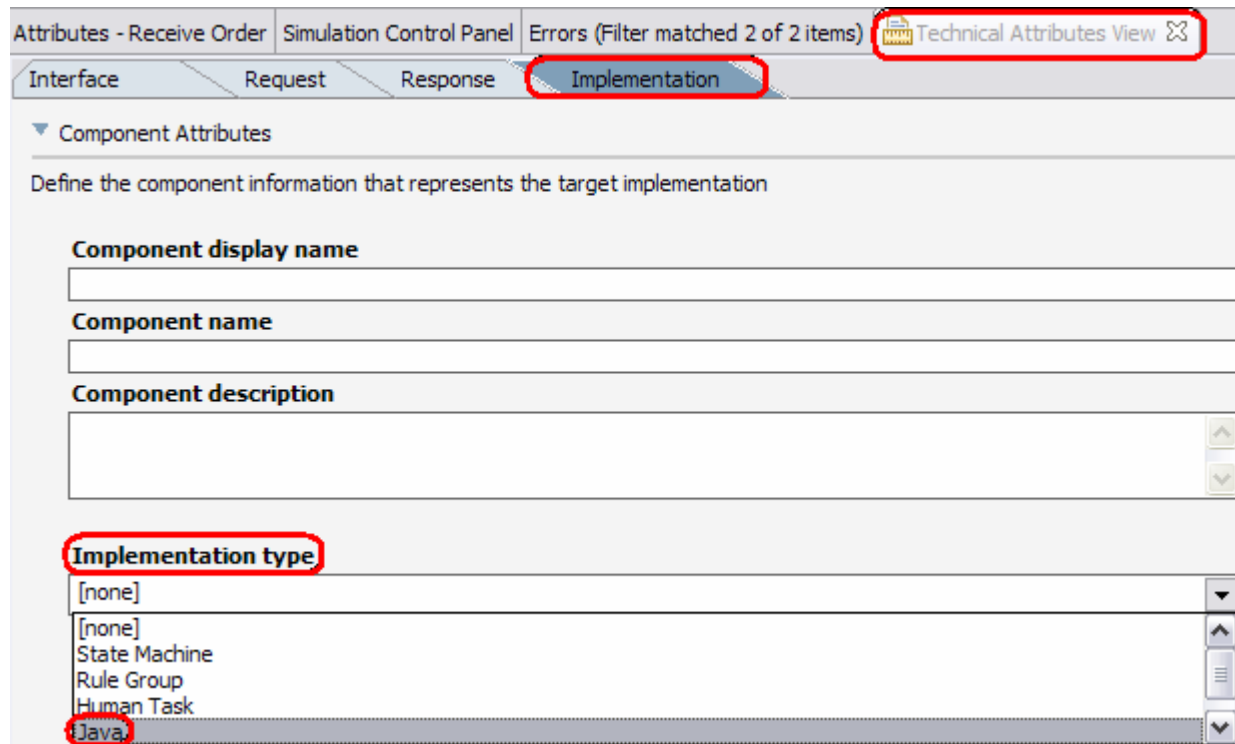
Note: You might see some warnings after changing the mode to WebSphere Process Server but you should not see any errors.

2. For a process destined for WebSphere Process Server, the different tasks in the process will be converted into activities in a BPEL business process. Set the tasks to use a Java implementation.

A resource assignment on a task can dictate what type of activity will be created in the BPEL business process. For example, if a role or person is assigned to a task and the task is local, then an inline human task will be created. If the task is global, then a participating human task that is a separate service component will be created. If the resource assigned to the task is a machine, then a Java component will be created. If no resource is specified, a Java component will also be created. Besides this behavior, you can also specify the specific implementation for a task. The implementation setting will cause an empty or skeleton component implementation to be created. The empty implementation can be completed or act as a hint for the assembler who will connect the process up to the appropriate existing component. For simplicity, a Java component will be used for this lab exercise. This will be further explained later in the lab.

- a. In the Project Tree view, select **ProjectScenario > Processes** and open the process by double-clicking the **ProcessScenario** process.
- b. Select the **Receive Order** task.
- c. From the **Attributes View** of the Receive Order task, select **Technical Attributes View**.
- d. Select **Implementation** in the Technical attributes view.

___ e. Select **Java** as the **Implementation Type** from the drop down list.



___ f. Select the **Package and Ship Order** task in the process diagram and repeat steps from 3-b through 3-d to change the Implementation type to Java.

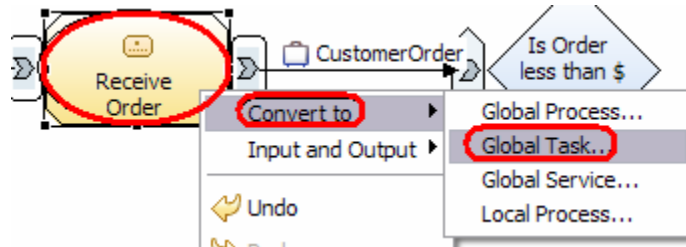
___ g. Save your work (**Ctrl+S**).

___ 3. Convert the tasks of the ProcessScenario to Global tasks.

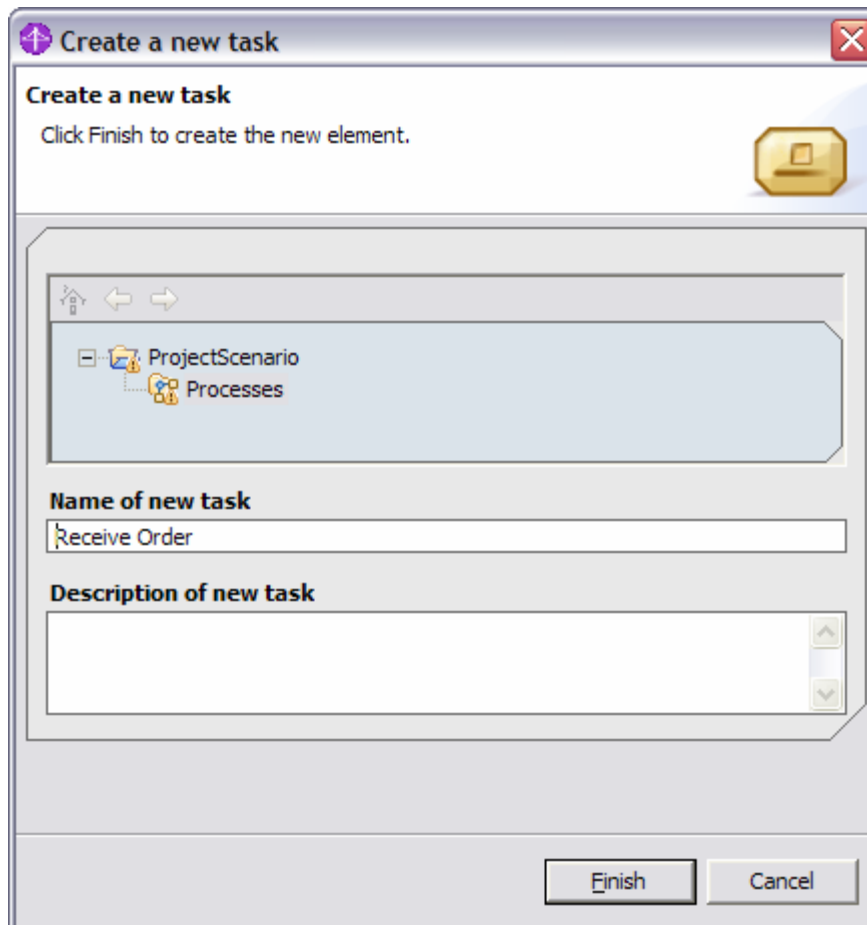
When preparing a process for WebSphere Process Server, it is a best practice to make your tasks global tasks; this will cause the interfaces for the tasks to be packaged into a library and available to other components in other modules. This will be further emphasized once you import the process into WebSphere Integration Developer. It is a best practice to have a backup of your process prior to this step.

___ a. Right click on the **Receive Order** task.

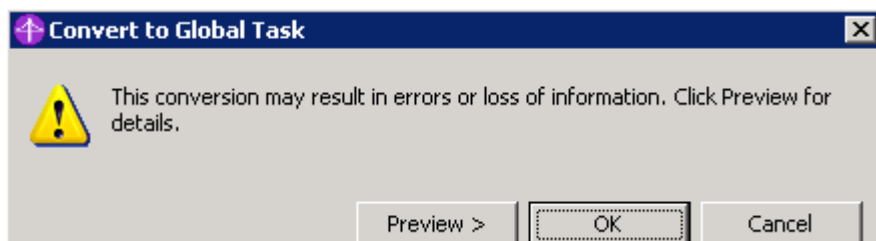
___ b. Select **Convert to > Global Task**.



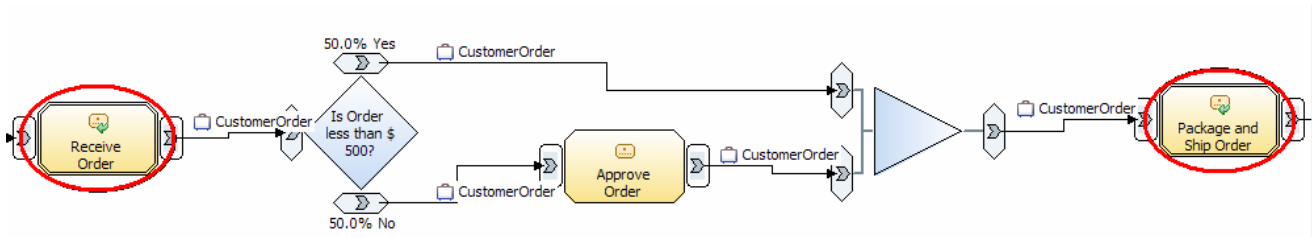
___ c. The window **Create a new task** will be opened. Make sure that the **Name of new task** is **Receive Order** and click **Finish**.



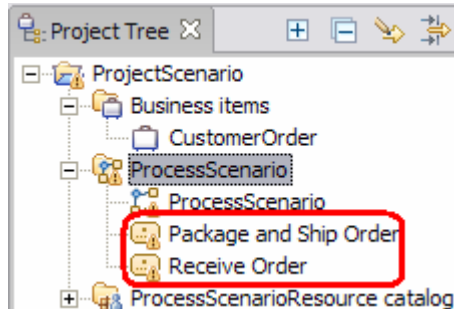
___ d. Right click on the **Package and Ship Order** task and repeat steps b and c to convert it to global task. Click **OK** if you get the following window:



__ e. Observe the difference between the local and global task in the process diagram:



__ f. From the Project Tree, you can observe two new global tasks under the ProcessScenario.



__ g. Save your work (**Ctrl+S**).

Note: You will not set the implementation on the Approve Order task or convert it to a global task. The Approve Order task has a Role (Account Manager) defined as a resource and will be implemented as a human task. If the task is local, then the human task will be an inline human task inside the BPEL process. For the process, this tight integration will be used since the human task step is a critical part of the process. There is no need to make the task a separate service component that will be called outside of a business process. Converting the Approve Order task to a global task would generate a separate human task service component which would have to be managed separately from the BPEL process.

___ 4. The interfaces and BPEL activities that will be generated for the process can be customized prior to exporting the project. It is recommended that careful consideration is used when specifying these values since they will be exposed as part of the service definition for the implementation.

__ a. Open the **ProcessScenario** process diagram and click on the white space on the diagram.

__ b. Select **Technical Attributes View** from the Attributes View.

___ c. Select **General** tab and enter the following under **BPEL Attributes**:

- Target namespace: <http://com/asdfparts/projectscenario>
- Process display name: **OrderProcessScenario**
- Process name: **OrderProcessScenario**

Attributes - ProcessScenario | Simulation Control Panel | Errors (Filter matched 2 of 6 items) | **Technical Attributes View**

General | Interface | Request | Response | Implementation

▼ BPEL Attributes

Define BPEL process information below

Target namespace

Process display name

Process name

___ d. Select **Interface** tab and enter the following under **WSDL Interface Attributes**:

- Target namespace:
<http://com/asdfparts/projectscenario/OrderProcessScenarioInterface>
- PortType name: **OrderProcessScenario**

General | **Interface** | Request | Response | Implementation

▼ WSDL Interface Attributes

Define WSDL interface information below

Target namespace

PortType name

Callback PortType target namespace

Callback PortType name

___ e. Select **Request** tab.

___ f. Select the **Input criterion** and enter the following:

- Operation name: **ReceiveOrder**
- Message name: **InputMessage**
- Part name: **CustomerOrder**
- Activity display name: **OrderReceived**
- Activity name: **OrderReceived**

These settings customize the interface of the BPEL process.

General Interface **Request** Response Implementation

▼ Input criterion section

Select an input criterion in the table to see the WSDL and BPEL information in the sections below

Input criterion name	Corresponding output crit...	Valid operation type	Selected operation type
Input Criterion	Output Criterion	Request / response, one ...	Request / response

▼ Operation type

Select the type of operation

Request / Response operation

One way operation

▼ WSDL Attributes

Define the WSDL interface information

Operation name
ReceiveOrder

▼ WSDL Message Details

Define the WSDL message that represents the input of an operation

Message name
InputMessage

Part name
CustomerOrder

▼ BPEL Attributes

Define the BPEL receive activity information

Activity display name
OrderReceived

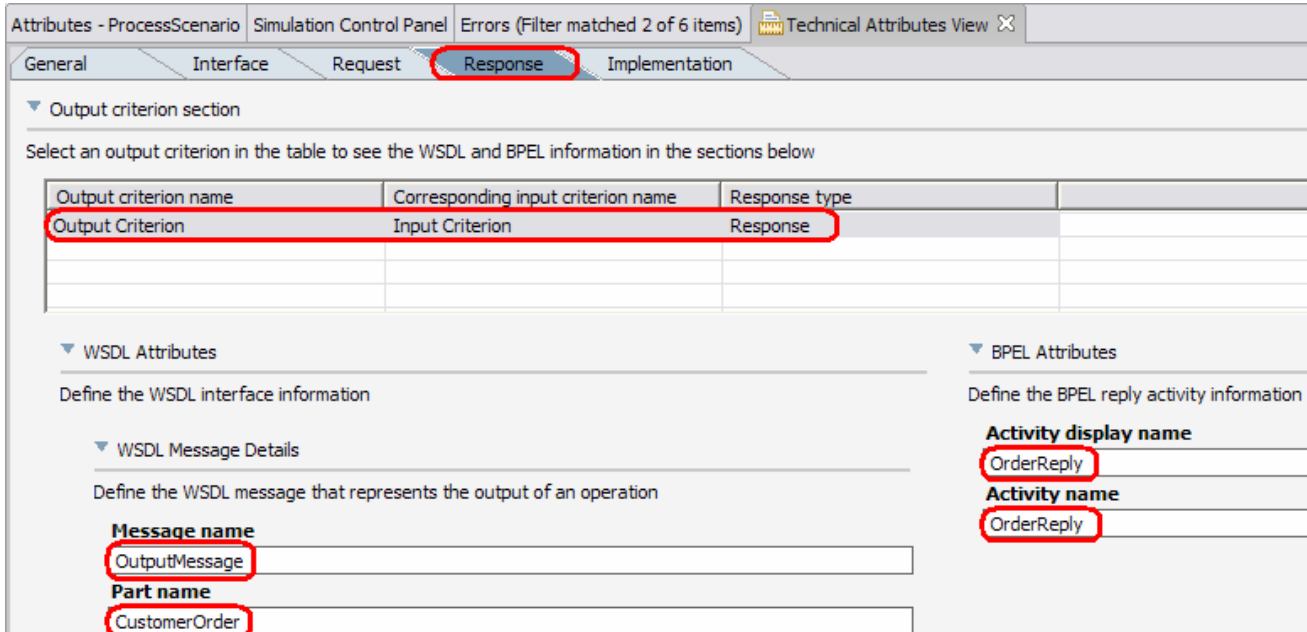
Activity name
OrderReceived

___ g. Select **Response** tab.

___ h. Select **Output criterion** and enter the following:

- Message name: **OutputMessage**
- Part name: **CustomerOrder**
- Activity display name: **OrderReply**
- Activity name: **OrderReply**

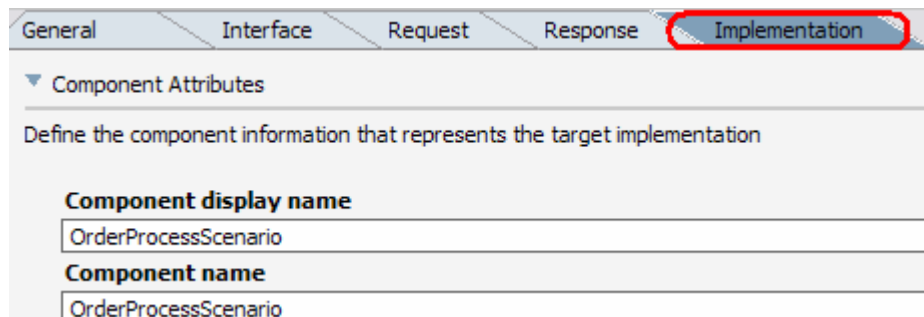
These settings customize the response portion of the interface.



___ i. Select **Implementation** tab and enter the following:

- Component display name: **OrderProcessScenario**
- Component name: **OrderProcessScenario**

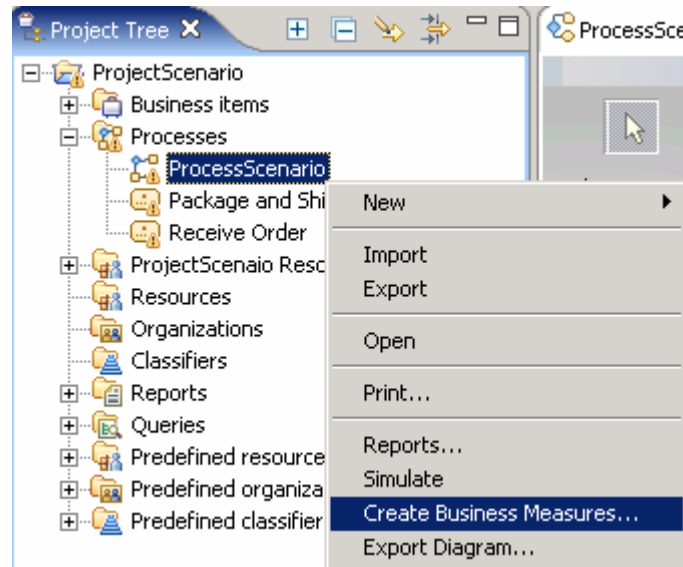
These settings customize the component that will represent the BPEL process.



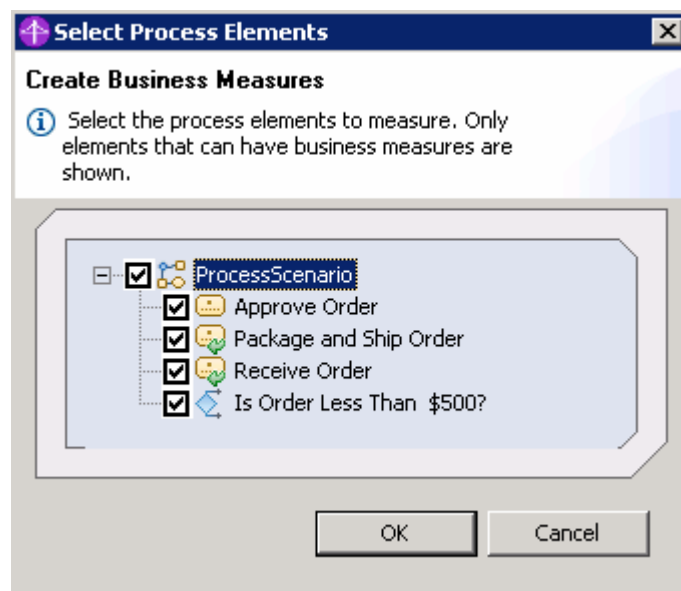
___ j. Save your work (**Ctrl+S**).

Note: You can also customize each task of the process individually. Click the Receive Order or Package and Ship Order task in the ProcessScenario process diagram. From the Technical Attributes View select General, Interface, Request, Response, and Implementation tabs and enter the required information as explained in step 4.

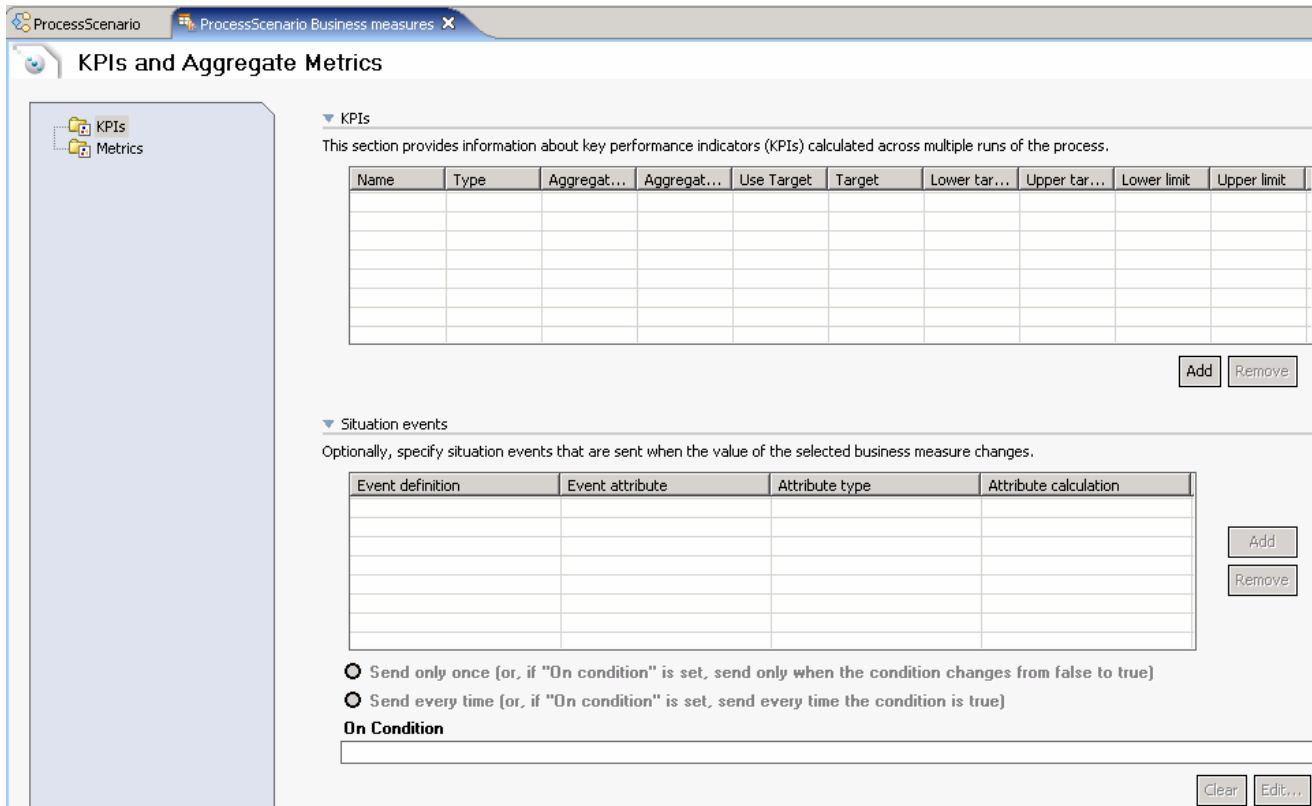
- ___ 5. In order to correctly export out the process, the default business measures will need to be created.
Create business measures.
 - ___ a. From Project Tree, select **Processes > ProcessScenario**.
 - ___ b. Right click on **ProcessScenario** and select **Create Business Measures...**



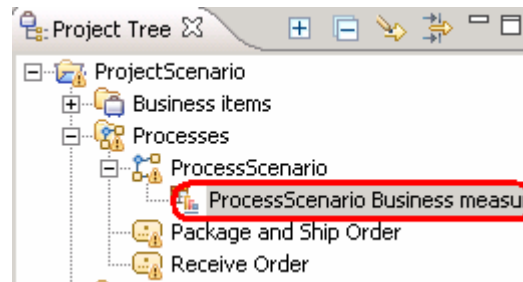
- ___ c. Click **OK** to accept all selected elements for creating business measures.



___ d. ProcessScenario Business measures window will be opened. Review and close that window.



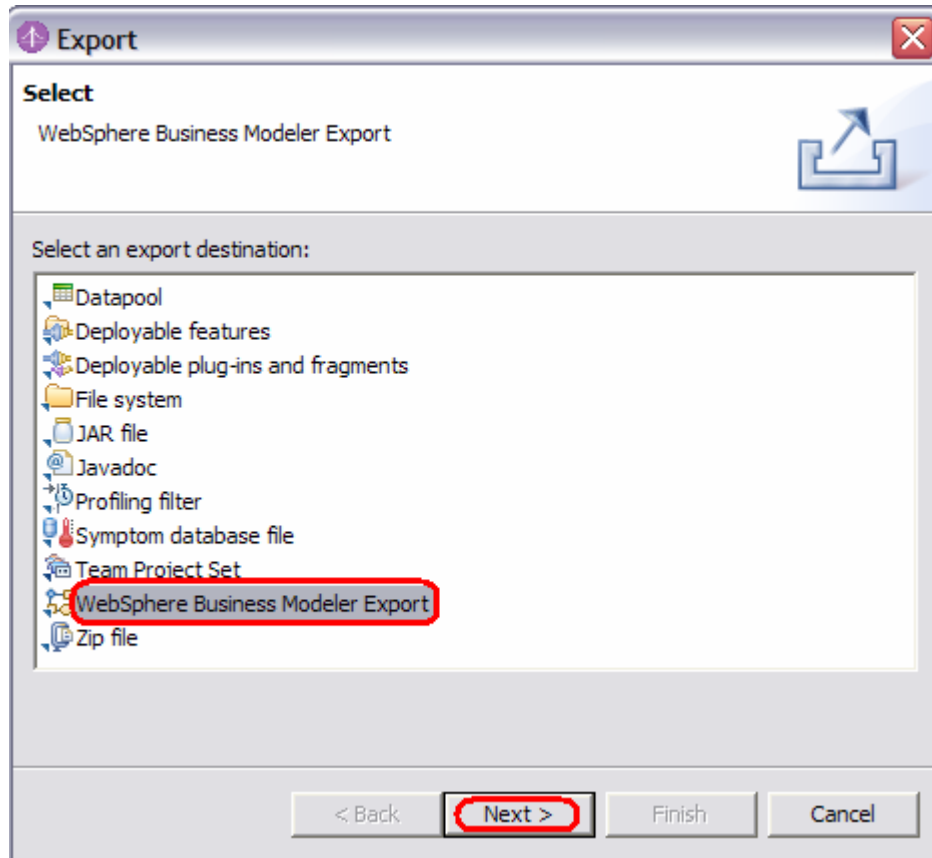
___ e. **ProcessScenario Business measures** will be added under the ProcessScenario process in the Project tree:



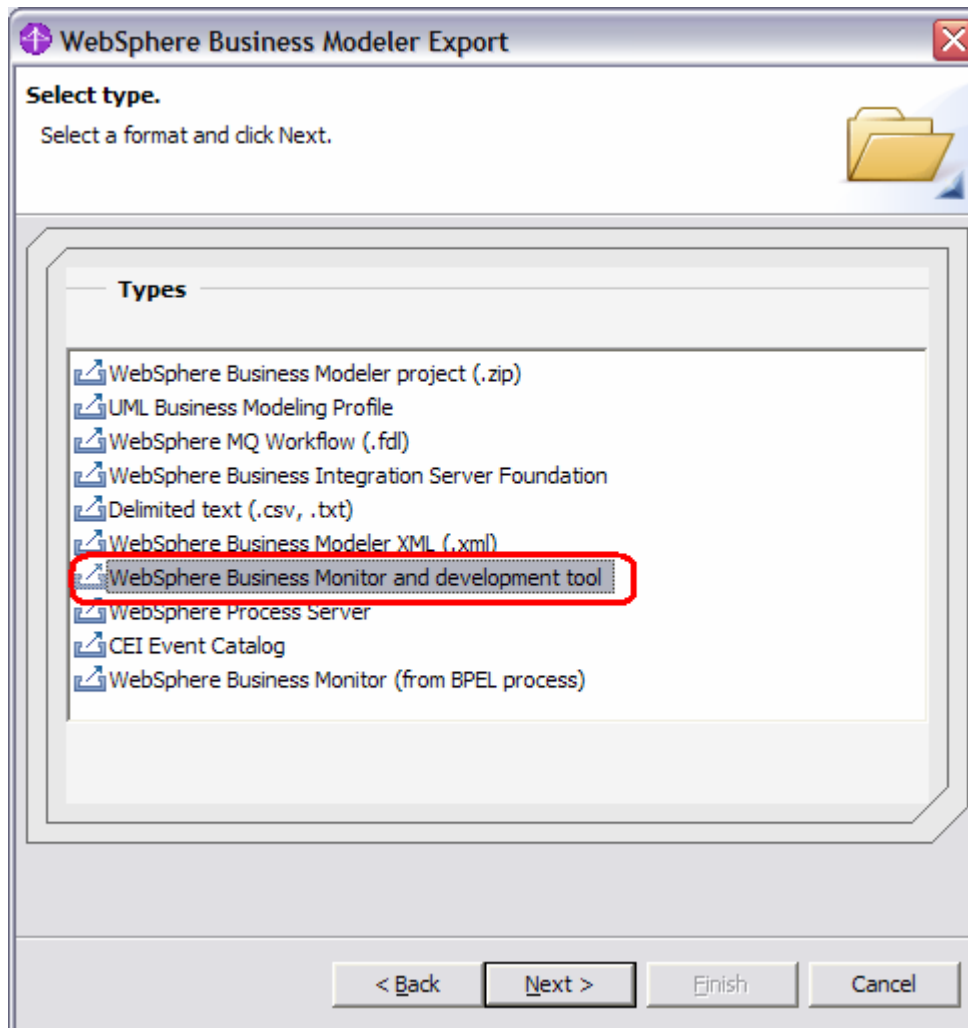
___ 6. The process is customized and ready to be imported into WebSphere Integration Developer. Export the ProjectScenario project from Modeler.

___ a. Select **File > Export**.

__ b. From the **Export** window select **WebSphere Business Modeler Export** and click **Next**.

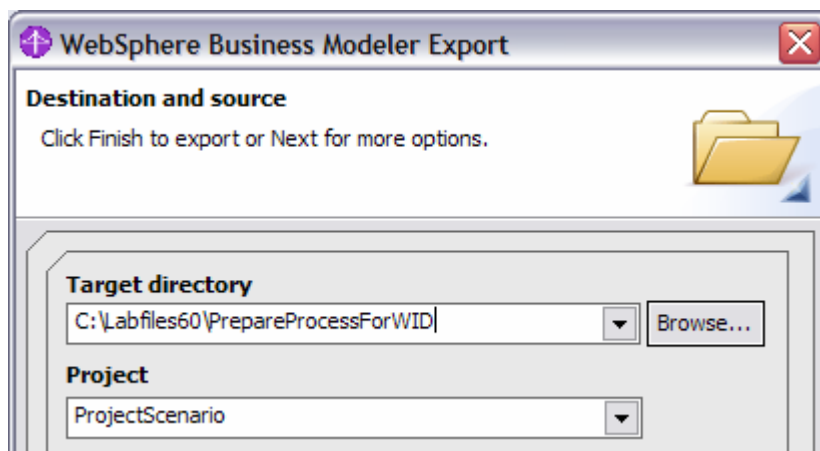


- ___ c. The WebSphere Business Modeler Export window is opened. Select Type as **WebSphere Business Monitor and development tool** and click **Next**.

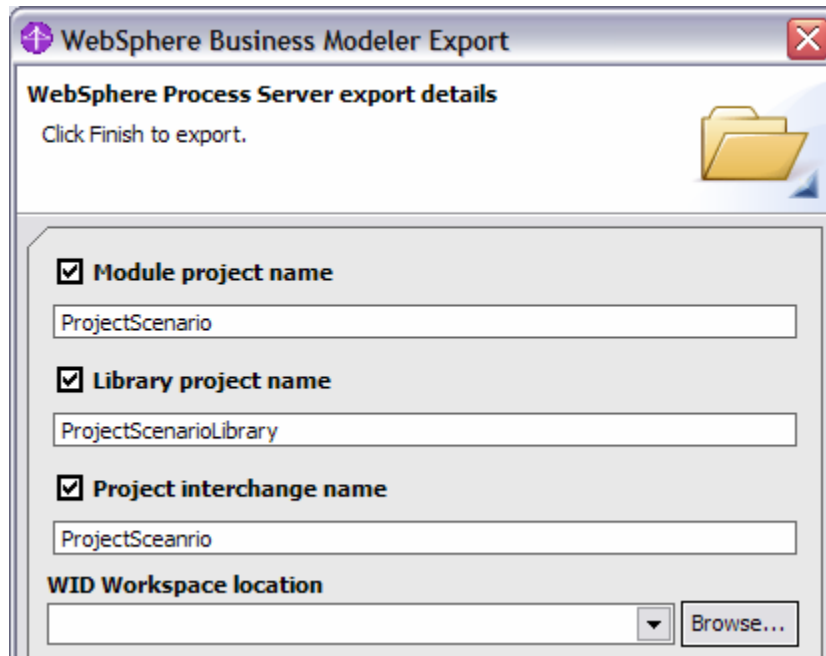


- ___ d. Click **Browse...** and select **C:\<LabFiles60>\PrepareProcessForWID** as the **Target directory**.

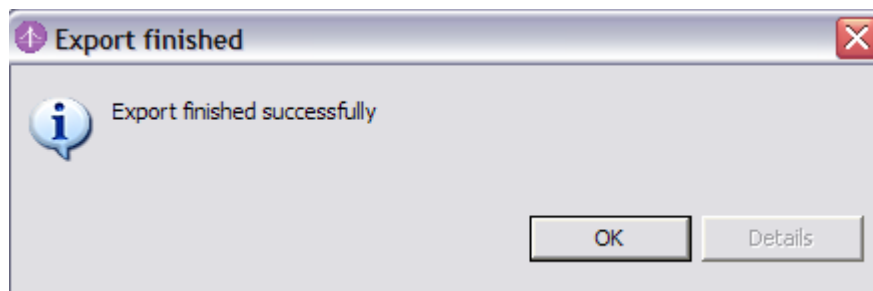
- ___ e. Select the **Project** as **ProjectScenario**.



- ___ f. Click **Next**.
- ___ g. Click **Module project name** and enter **ProjectScenario**.
- ___ h. Click **Library project name** and enter **ProjectScenarioLibrary**. All interfaces for the process and global tasks will be placed in the library and available for use by other components not in the module containing the process.
- ___ i. Click **Project interchange name** and enter **ProjectScenario**.



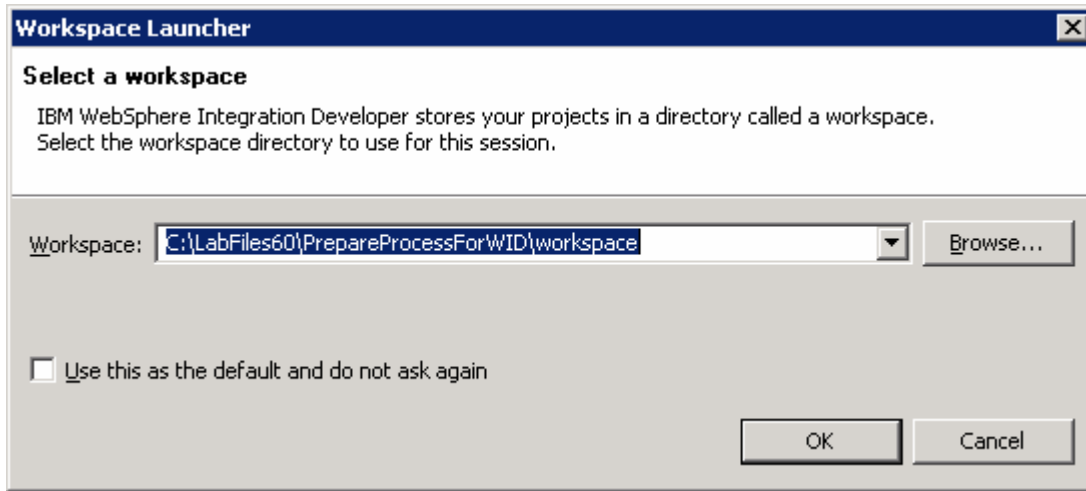
- ___ j. Click **Finish** from the **Export finished** window.



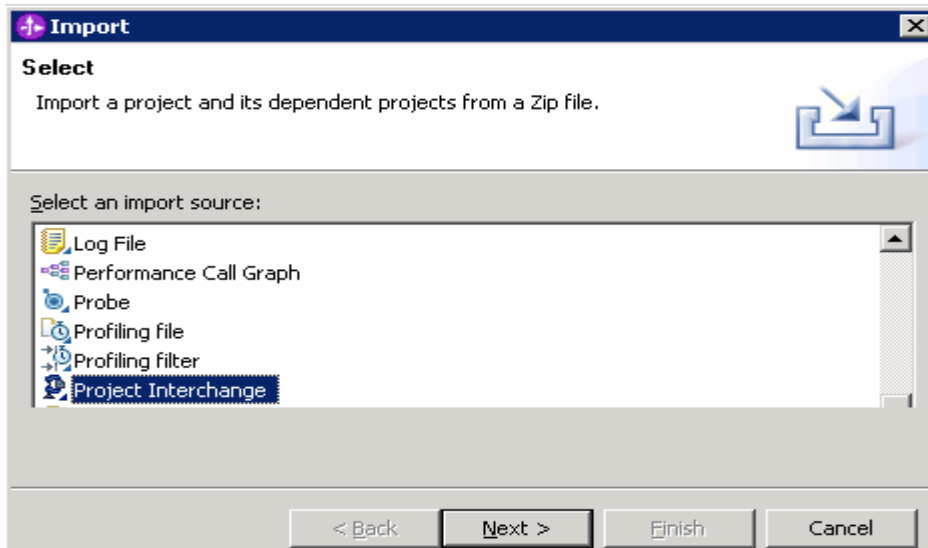
- ___ 7. Select **File > Exit** to exit from the modeler.

Part 3: Initialize workspace for WebSphere Integration Developer

- ___ 1. Start WebSphere Integration Developer V6.0.1.
 - ___ a. Select **Start > Programs > IBM WebSphere > Integration Developer V6.0.1 > WebSphere Integration Developer 6.0.1**.
 - ___ b. Workspace Launcher window will be displayed. Click the **Browse** button and select **<LabFiles60>\PrepareProcessForWID\workspace** as your workspace directory.

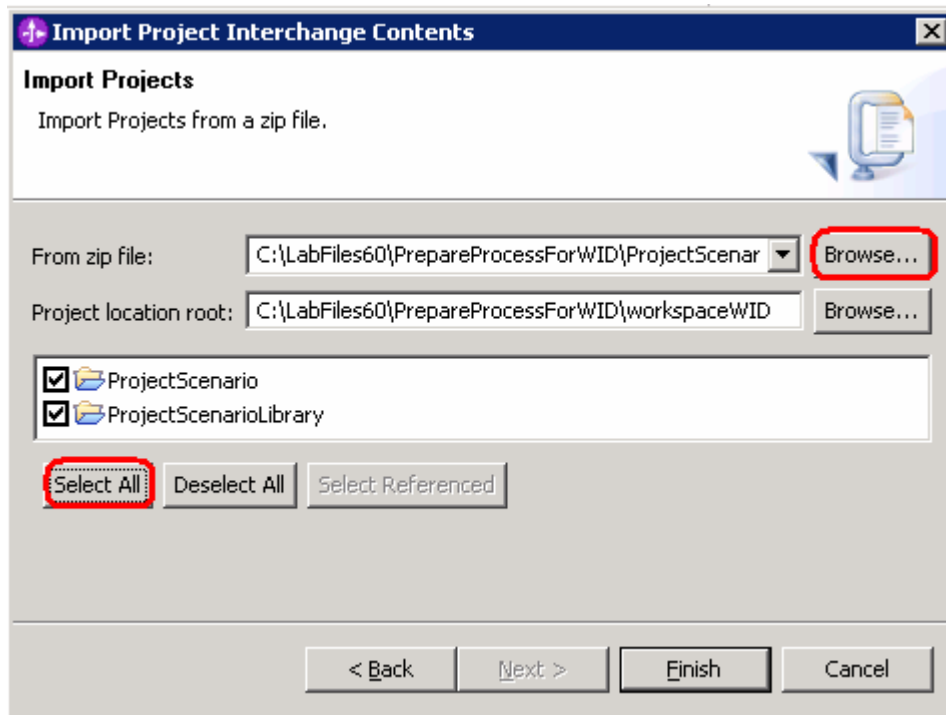


- ___ c. Click **OK**.
- ___ 2. Import the Model into WebSphere Integration Developer.
 - ___ a. Select **File > Import** from the IBM WebSphere Integration Developer.
 - ___ b. From the **Import** window select **Project Interchange** and click **Next**.



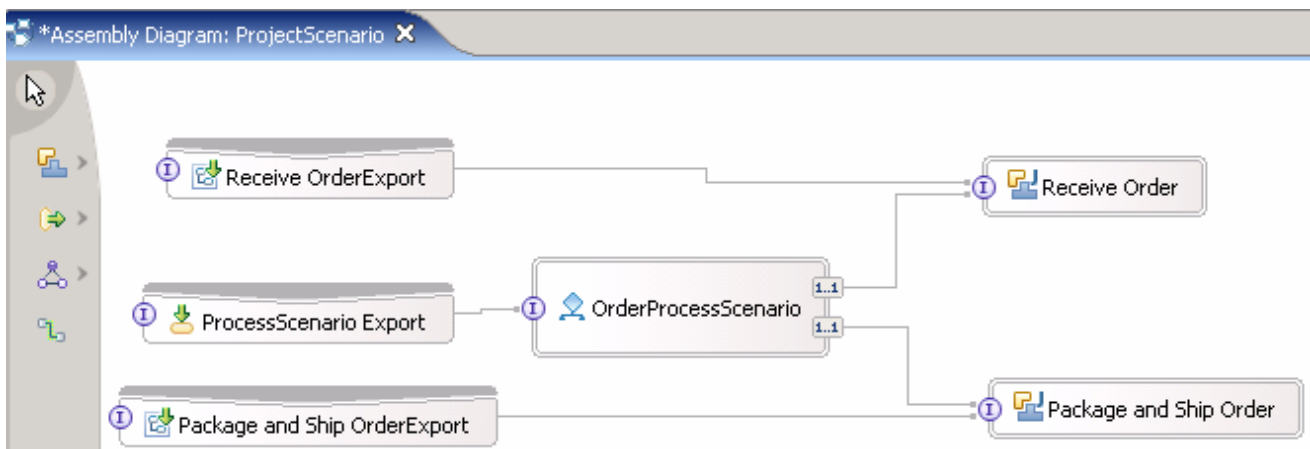
Import Project Interchange Contents window will be opened.

- ___ c. Click **Browse...** and select **C:\<LabFiles60>\PrepareProcessForWID\ProjectScenario.zip** as source zip file.
- ___ d. Click **Select All** and click **Finish**.

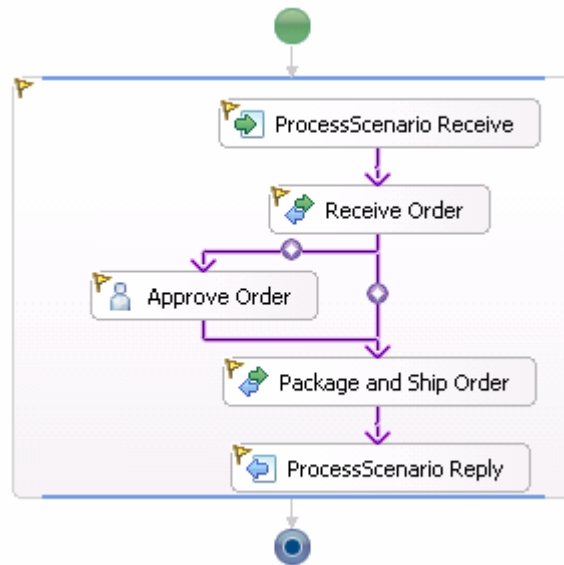


Part 4: Rewire Process for Components

- ___ 1. Review the imported artifacts.
 - ___ a. In the Business Integration view, notice the two projects. One (ProjectScenario) is the module containing the process and the other is a library (ProjectScenarioLibrary).
 - ___ b. Expand **ProjectScenarioLibrary > Interfaces**. Notice there are 3 interfaces in the library. One for the process (ProcessScenario) and one for each task (PackageandShipOrder and Receive Order). With the interfaces in the library, they can be used for components defined in other modules.
 - ___ c. In the Business Integration view, expand **ProjectScenario** and then double-click **ProjectScenario**.
 - ___ d. The Assembly editor will be opened. Notice how the different tasks of the process are exposed as Java components. The Approve Order task is not exposed as a separate component since it was a local task and it had a resource assigned to it that was a Role (Account Manager). Even though the Receive Order and Package and Ship Order tasks are separate components, they are in the same module as the process. This packaging structure is simple and fine for the testing and deployment of small applications. When processes become larger and iterative development of the process occurs, it becomes problematic to keep the components updated when they are in the same module. Many times the implementation of the component remains the same, or changes to it are independent of changes to the process. Changes to the process are also independent of changes to the individual task implementation. For these reasons, it is better to move these components and their implementation to a different module and use a SCA Import/Export to connect the components to the process. In the future, when changes are made to the process, the task implementation, which is a separate component in a separate module, can be left unchanged.



- ___ e. Double-click on the **OrderProcessScenario** component to open the process. Notice the Approve Order task is an inline human task.



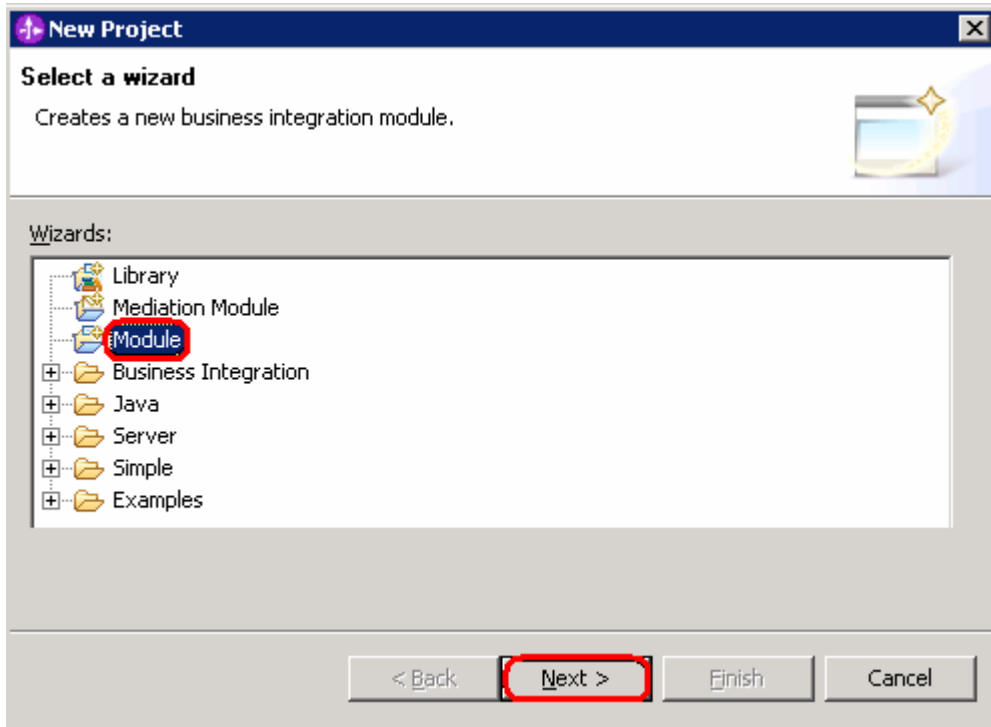
- ___ f. Close the Process editor and the Assembly editor.

- ___ 2. In order for the Receive Order and Package and Ship Order component implementations to be isolated from changes in the process, separate modules for each component should be created. Create the modules.

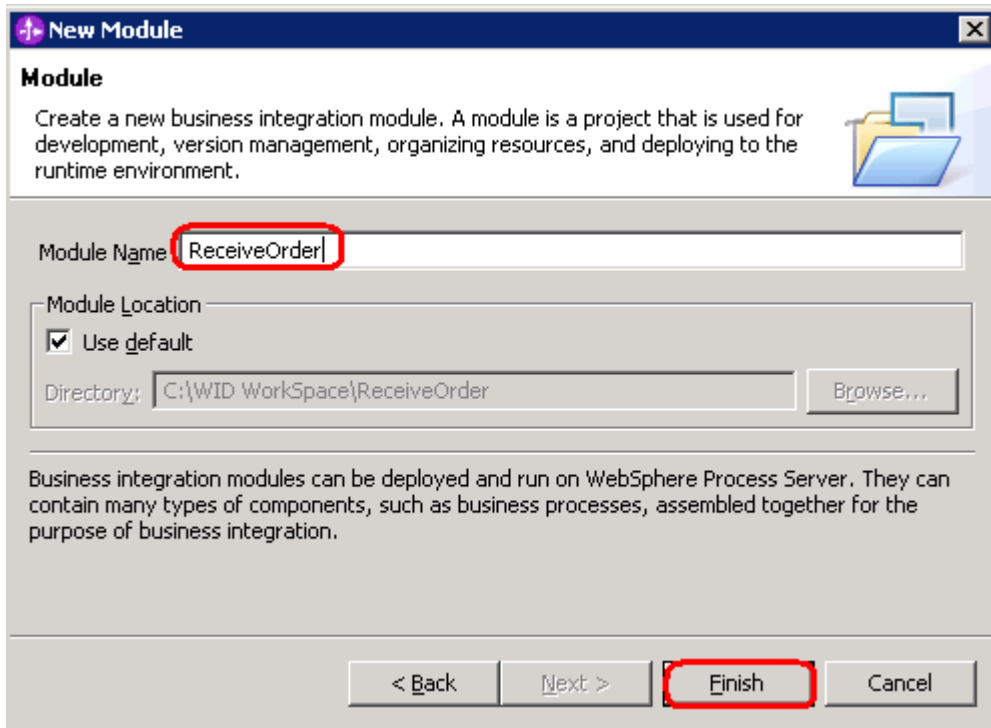
- ___ a. Select **File > New > Project** from the main menu.



__ b. Select **Module** from the **New Project** window and click **Next**.



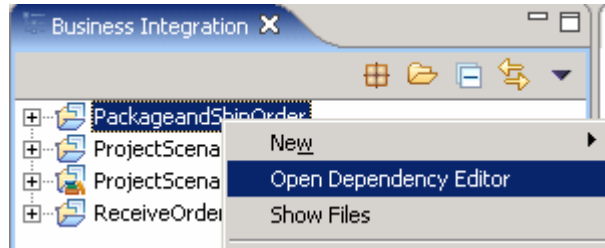
__ c. From **New Module** window enter **ReceiveOrder** as **Module Name** and click **Finish**.



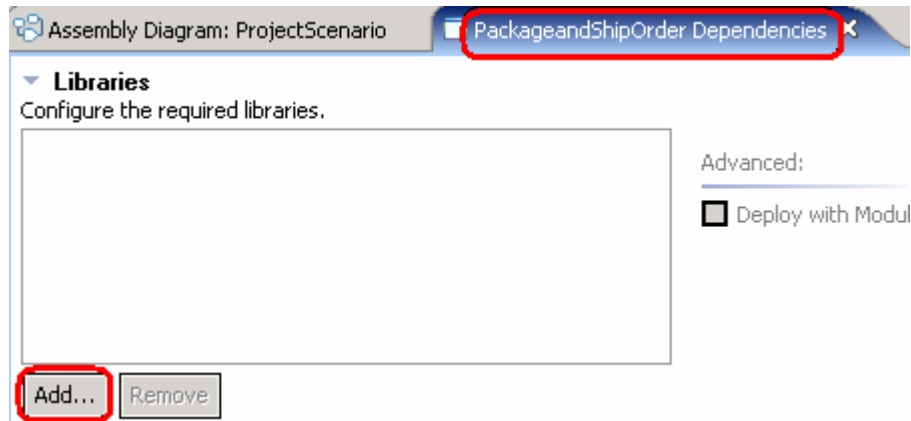
__ d. Repeat the above steps to create another module: **PackageandShipOrder**.

___ 3. In order for components in the new modules to be able to provide the implementation for the different tasks, the appropriate interfaces in the ProjectScenarioLibrary task must be used. A dependency between the module and the library must be established before the interface can be used for a component. Create the dependencies and add libraries to modules.

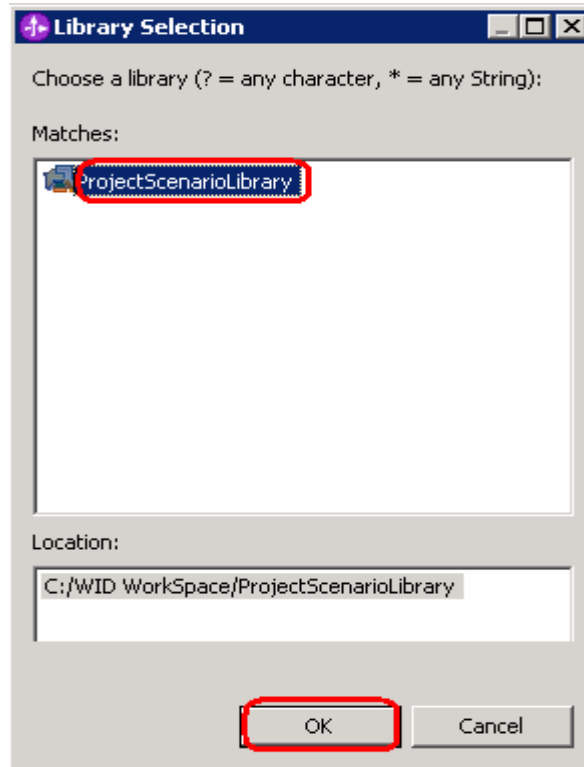
___ a. Right click on the **PackageandShipOrder** and select **Open Dependency Editor**.



___ b. Click **Add** from the **PackageandShipOrder Dependencies**.



- ___ c. Select **ProjectScenarioLibrary** from **Library Selection** window and click **OK**.



- ___ d. Save (**Ctrl+S**) and close the Dependency Editor.

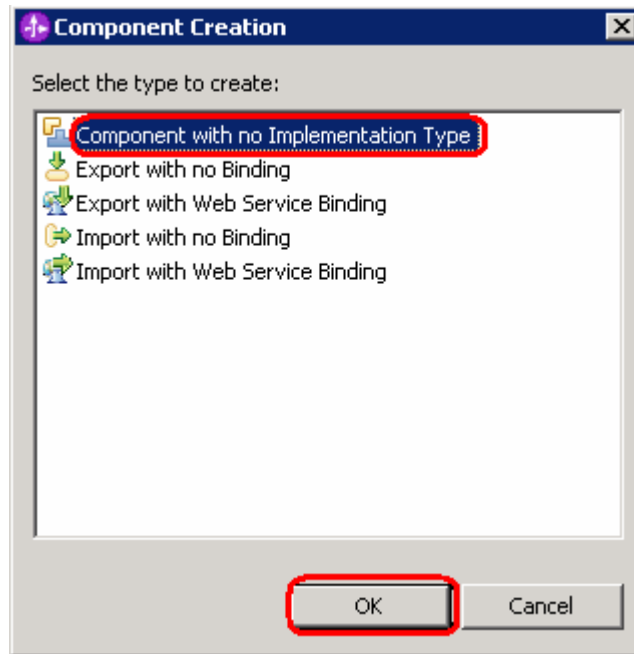
- ___ e. Repeat the above steps for **ReceiveOrder** module.

___ 4. Generate Implementation and Binding.

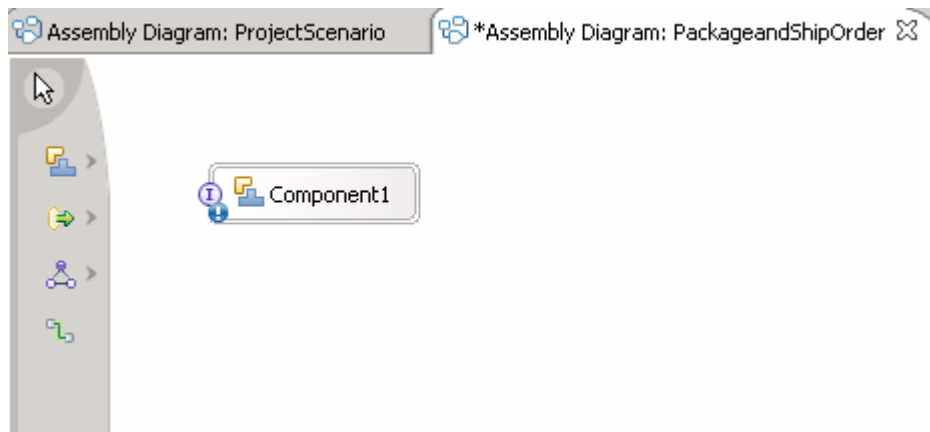
- ___ a. Double-click the **PackageandShipOrder** module and you will see an Assembly diagram for PackageandShipOrder opened on the right side window.

- ___ b. Drag the **PackageandShipOrder** interface from **ProjectScenarioLibrary > Interfaces** and drop it onto the Assembly diagram.

- ___ c. Select **Component with no Implementation Type** from the **Component Creation** window and click **OK**.

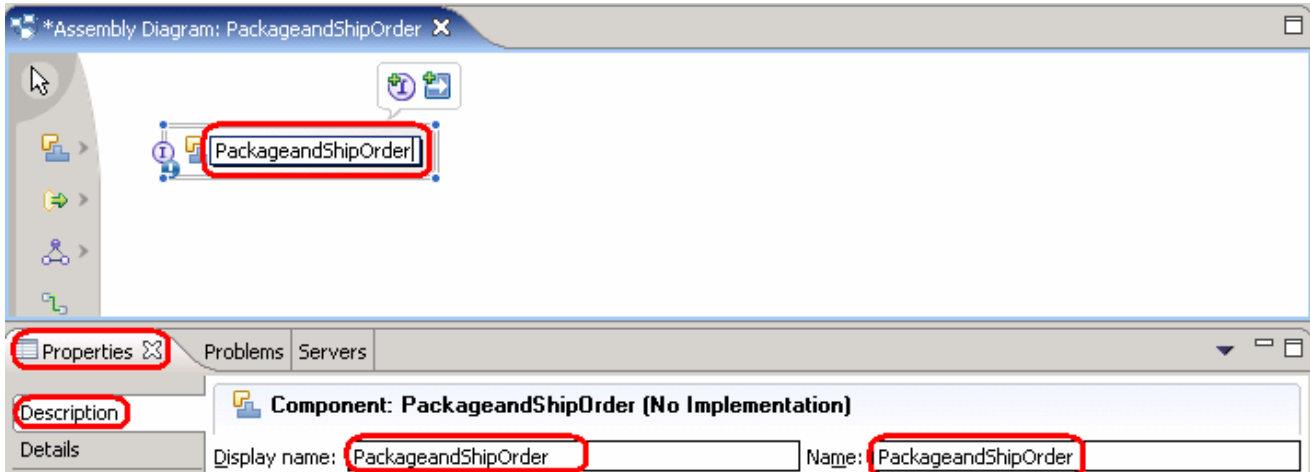


- ___ d. An unimplemented component will be created.

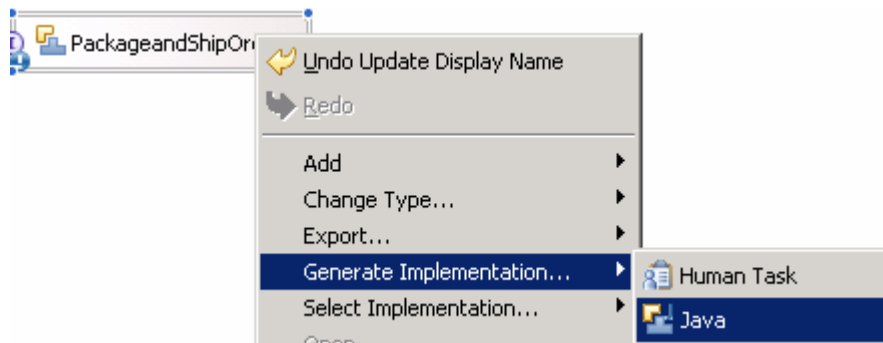


__ e. Click on the Component1 and rename it to **PackageandShipOrder**.

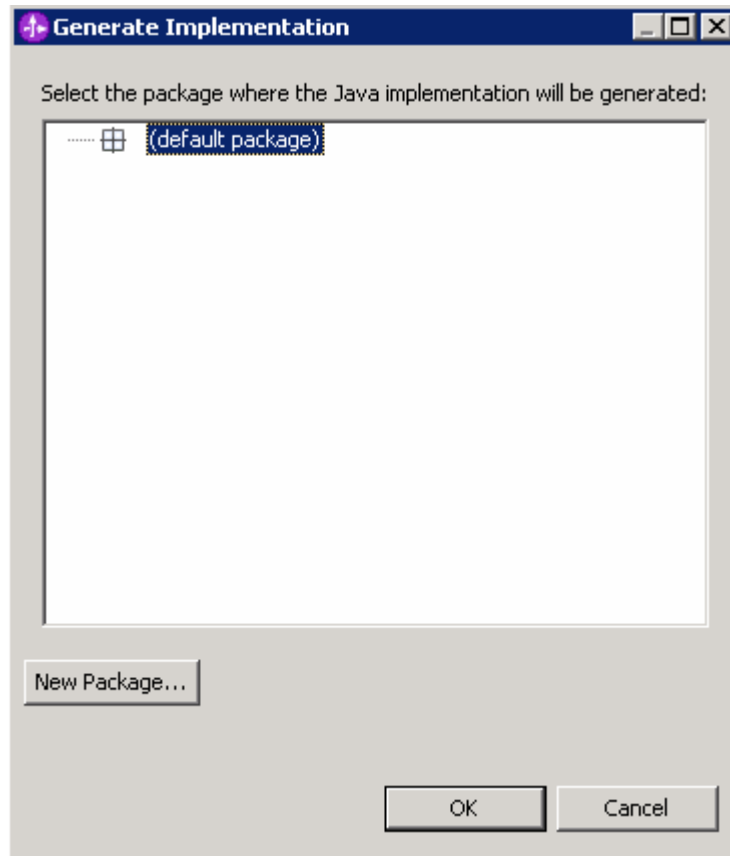
NOTE: You can also change the name by changing the Display name (Name field will automatically be changed filling Display name field) under the Description tab from Properties view of the Component1. If you change the name from the Assembly Diagram you should see the Display name and Name changed from Component1 to PackageandShipOrder in the Description tab of Properties View.



__ f. The component can be implemented any way as long as it uses the PackageandShipOrder interface. You could implement the component as a human task, business state machine, business rule or even a business process. For simplicity, right click on the **PackageandShipOrder** and select **Generate Implementation > Java**.



__ g. Accept the default package from Generate Implementation and click **OK**.



__ h. PackageandShipOrderImpl.java will be opened in Assembly editor. Drop down the arrow (if it [▾] is not already) and modify the java code as below:

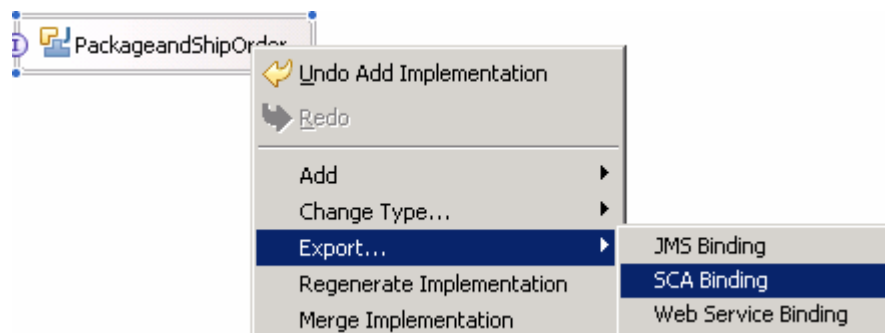
```

public DataObject InputCriterion(DataObject input) {
    return input;
}

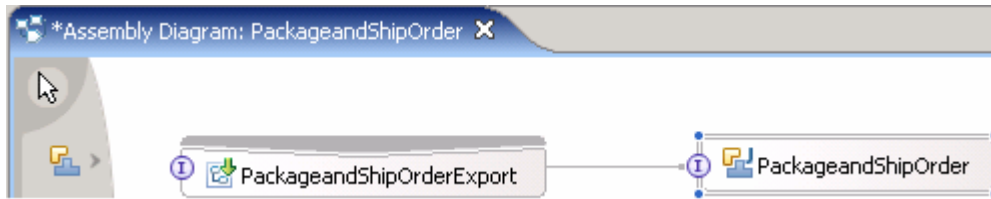
```

__ i. Save your work (**Ctrl+S**). Close the PackageandShipOrderImpl.java file.

__ j. Right click on the **PackageandShipOrder** and select **Export > SCA Binding**. This will make the component available to be called from outside of the module.



__ k. Your PackageandShipOrder Assembly diagram will now look like this:

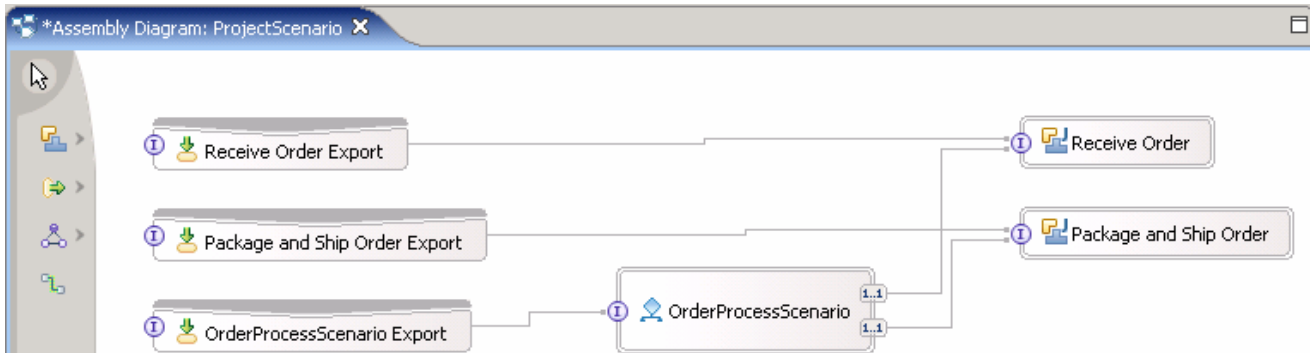


__ l. Save your work (**Ctrl+S**) and close the PackageandShipOrder Assembly diagram.

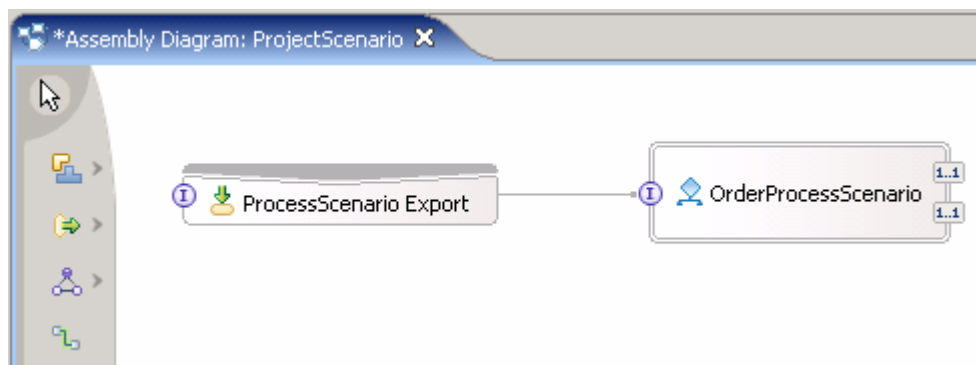
__ m. Repeat the above steps for **ReceiveOrder** module.

4. Rewire process to components.

__ a. Open the Assembly editor for ProjectScenario again by double-clicking on **ProjectScenario > ProjectScenario** in the Business Integration view.



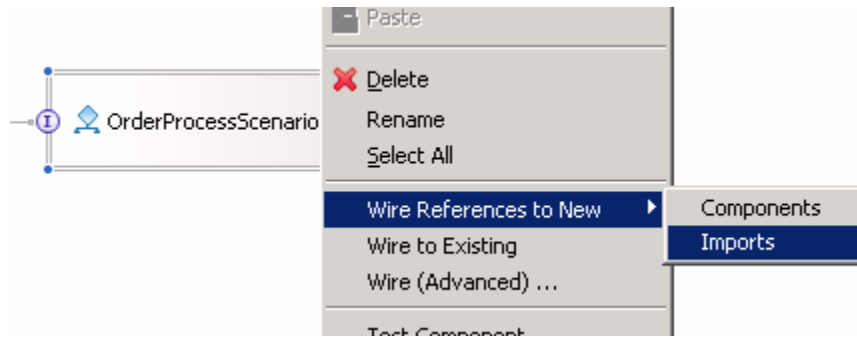
__ b. Delete Receive Order, Receive Order Export, Package and Ship Order, Package and Ship Order Export components.



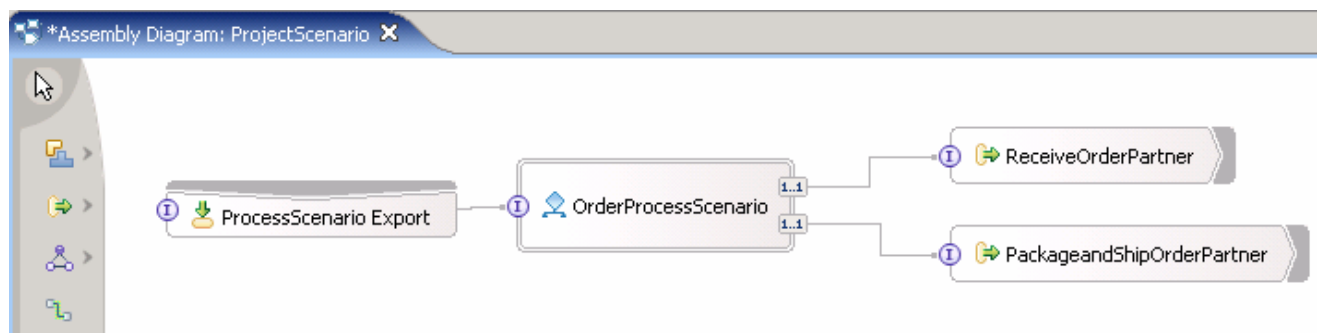
__ c. Save (**Ctrl+S**) your work.

Note: You should save your assembly diagram after deleting the above elements. Otherwise, you will see these exports while selecting SCA exports in step j and m.

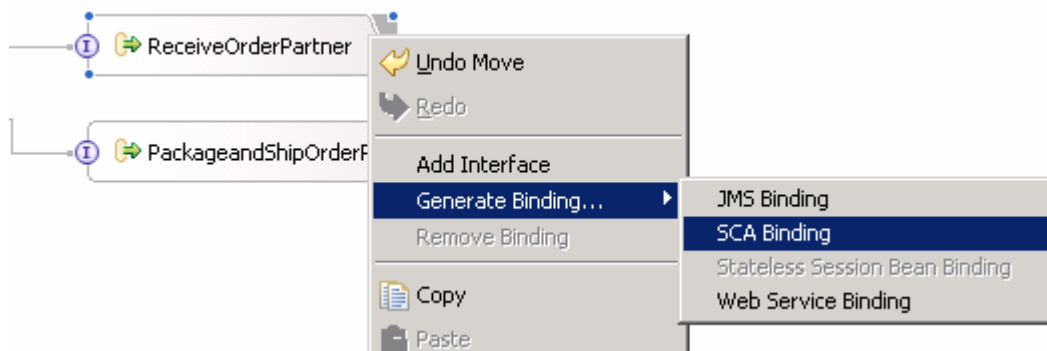
__ d. Right click on **OrderProcessScenario** and select **Wire References to New > Imports**.



__ e. Imports matching the ReceiveOrder and PackageandShipOrder interfaces will be created.

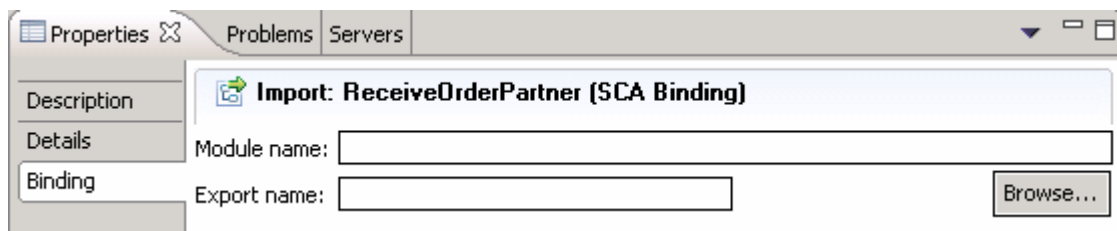


__ f. Right click on **ReceiveOrderPartner** and select **Generate Binding > SCA Binding**.



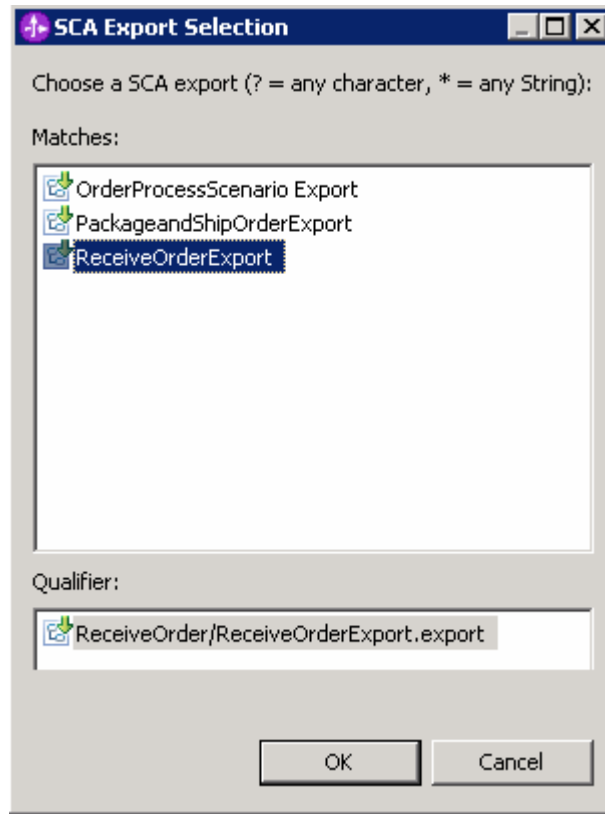
__ g. Repeat the above step for **PackageandShipOrderPartner** to generate SCA Binding.

__ h. Select **ReceiveOrderPartner** and then from the **Properties** view at the bottom, select **Binding**.



__ i. Select **Browse....** SCA Export Selection window will be opened.

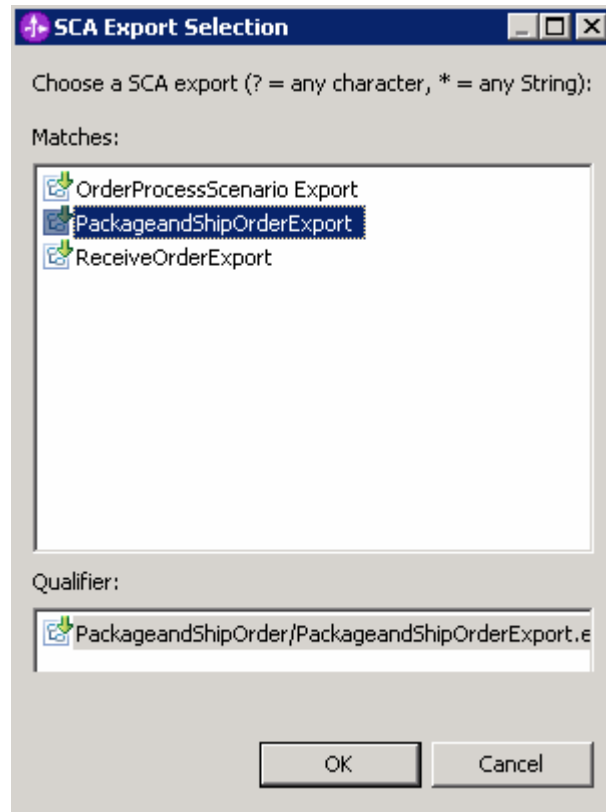
__ j. Select **ReceiveOrderExport** under **Matches** and click **OK**.



__ k. Select **PackageandShipOrderPartner** and select **Binding** from the Properties view.

__ l. Select **Browse....**

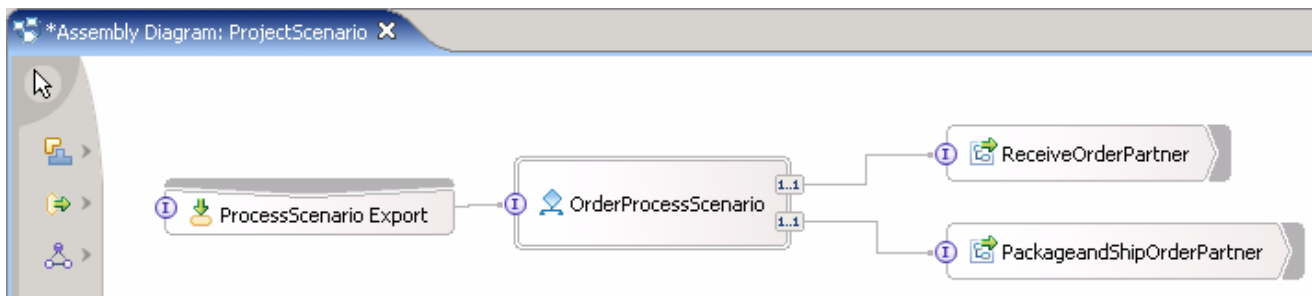
___ m. Select **Package and Ship Order Export** under **Matches**.



___ n. Click **OK**.

___ o. Save your work (**Ctrl+S**).

___ p. The process is complete and ready for deployment.



___ 5. Select **File > Exit** to exit from the WebSphere Integration Developer.

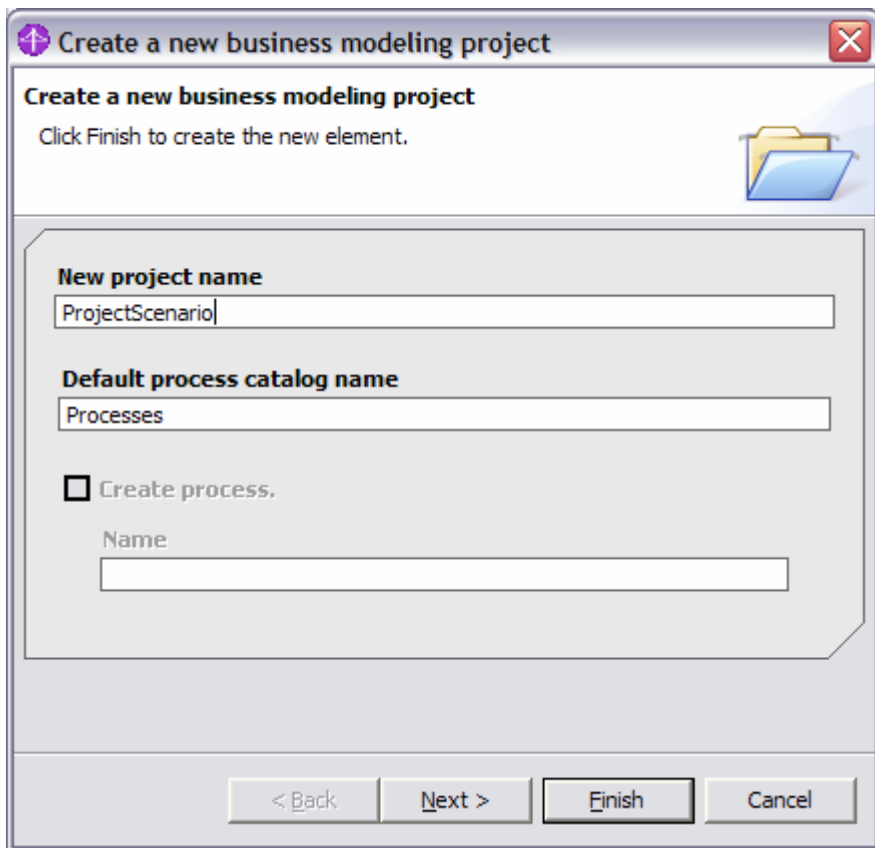
What you did in this exercise

In this lab, you imported a project ProjectScenario into the WebSphere Business Modeler V6. In the WebSphere Process Server modeling mode, you set implementation type to Java for Receive Order and Package and Ship Order tasks. You also converted these tasks to global tasks, and then you added customization information to the process and exported the project.

You then imported the project into WebSphere Integration Developer V6.0.1 and created Receive Order and Package and Ship Order modules for the different components. You established dependencies to the ProjectScenario library before creating components from the common interfaces. You then created SCA bindings for the components using Exports. Finally you rewired the process to use the Imports.

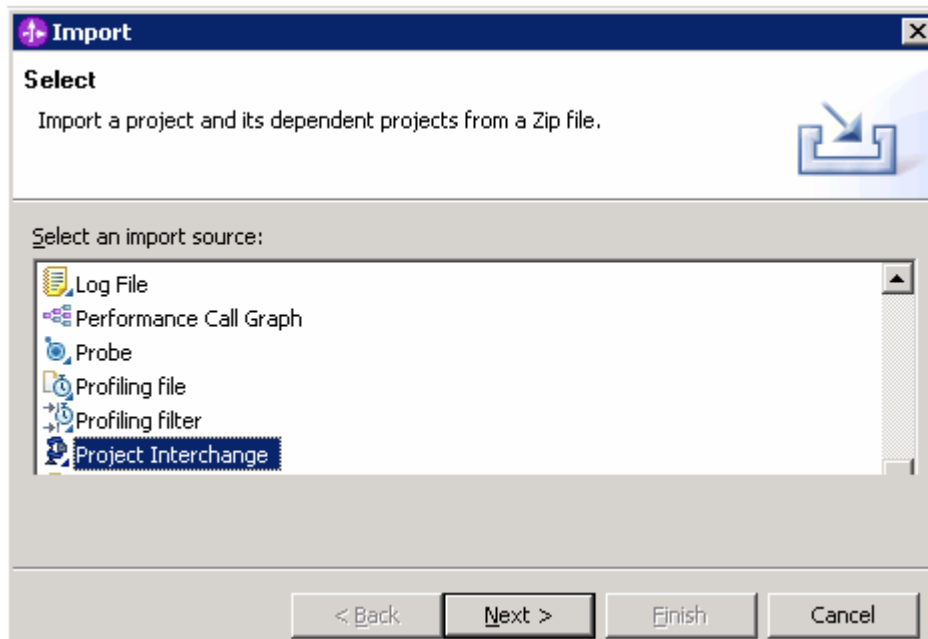
Solution Instructions

- ___ 1. Import the ProjectScenario project into WebSphere Business Modeler for viewing.
 - ___ a. Select **File > Import...**
 - ___ b. Select **WebSphere Business Modeler Import** and click **Next**.
 - ___ c. Select **WebSphere Business Modeler project (.zip)** and click **Next**.
 - ___ d. For the Source directory, click **Browse...** and select the folder **<LAB_FILES>\PrepareProcessForWID\Solution**.
 - ___ e. Select **ProjectScenarioModeler.zip**.
 - ___ f. To the right of the Target project field, click **New**.
 - ___ g. Change the New project name to **ProjectScenario**.



- ___ h. Click **Finish**. The Target project field should be filled in with the ProjectScenario value.
 - ___ a. Click **Finish** to complete the import.
- ___ 2. Import the Model to IBM WebSphere Integration Developer.
 - ___ a. Select **File > Import** from the IBM WebSphere Integration Developer.

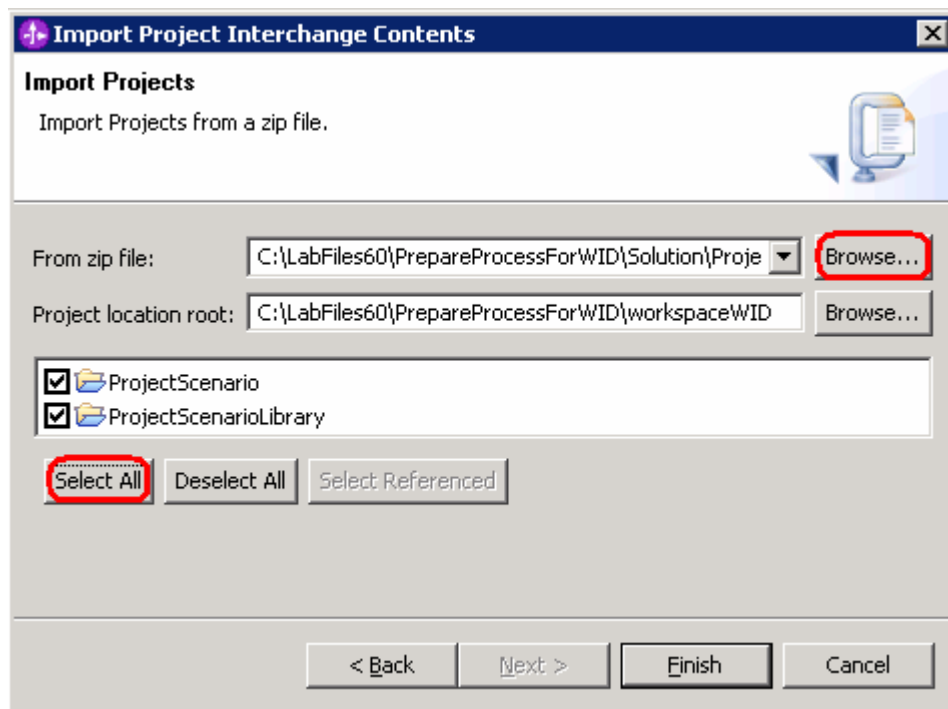
___ b. From the **Import** window select **Project Interchange** and click **Next**.



Import Project Interchange Contents window will be opened.

___ c. Click Browse and select
<LabFiles60>\PrepareProcessForWID\Solution\ProcessScenarioWID.zip as source zip file.

___ d. Click **Select All** and click **Finish**.



This page is left intentionally blank.