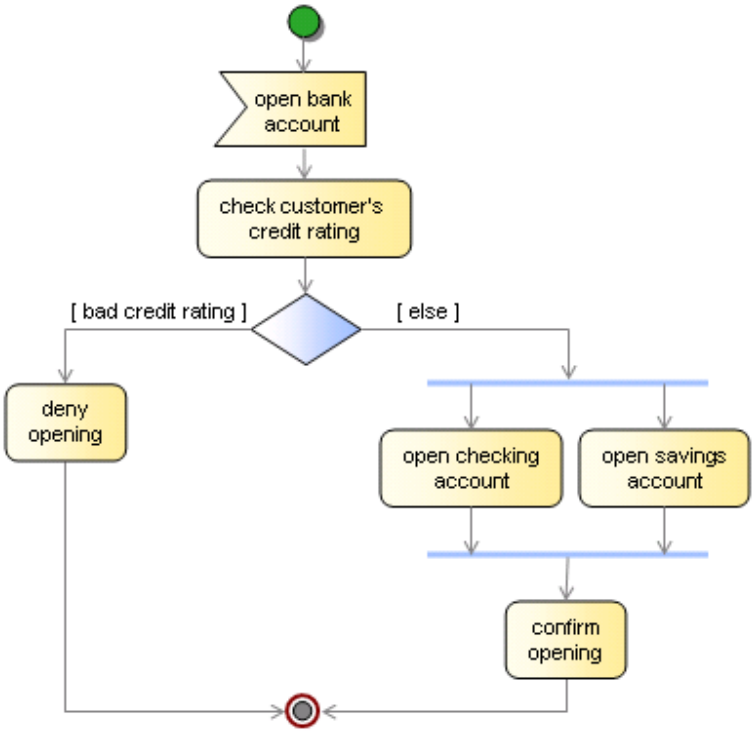
WebSphere Integration Developer – Business Process

Before the lab:

- **Logon to Windows**: pi-eadmin / passw0rd
- **Run the WebSphere Integration Developer** (WID), you can use icon on your Desktop
- In the WebSphere Integration Developer, **start the WebSphere Process Server 6.2**

1. Process Overview



2. Create a Library Project (for Common Artifacts)

To create the ***BankLib*** library project, complete the following steps:

1. Switch to the *Business Integration* perspective.
2. Click the **File > New…** menu item, select Library, name it ***BankLib***.
3. Click **Finish.**

3. Develop the Web Service (bottom up) – *CreditRatingService*

1) Dynamic Web Project

1. Switch to the Java EE perspective. To switch to the Java EE perspective, select **Window > Open Perspective > Other** from the Menu. In the Open Perspective dialog, select **Java EE** and click **OK**.
2. In the **Enterprise Explorer** view, right-click the canvas.
3. From the pop-up menu, select **New > Dynamic Web Project**. The New Dynamic Web Project window opens.
4. In the **Project name** field, enter ***CreditRatingService***.
5. For **Project contents**, keep the default setting.
6. Select ***WebSphere Process Server v6.2*** as the **Target Runtime**.
7. Select ***2.4*** for the **Dynamic Web Module version**.
8. Select ***<custom>*** for the **Configuration**.
9. Make sure **Add project to an EAR** is selected. Leave the default value (***CreditRatingServiceEAR***).
10. Click **Finish**.
11. If the Open Associated Perspective dialog appears, click **Yes**.

2) Create a Java Bean implementation of the Web Service

1. In the **Enterprise Explorer** view, right-click **CreditRatingService**.
2. From the pop-up menu, select **New > Class**.
3. In the **Package** field, enter ***bpc.samples.invoker***.
4. In the **Name** field, enter ***CreditRatingService***.
5. Click **Finish**.
6. Complete the implementation of **CreditRatingService** by adding the following method:

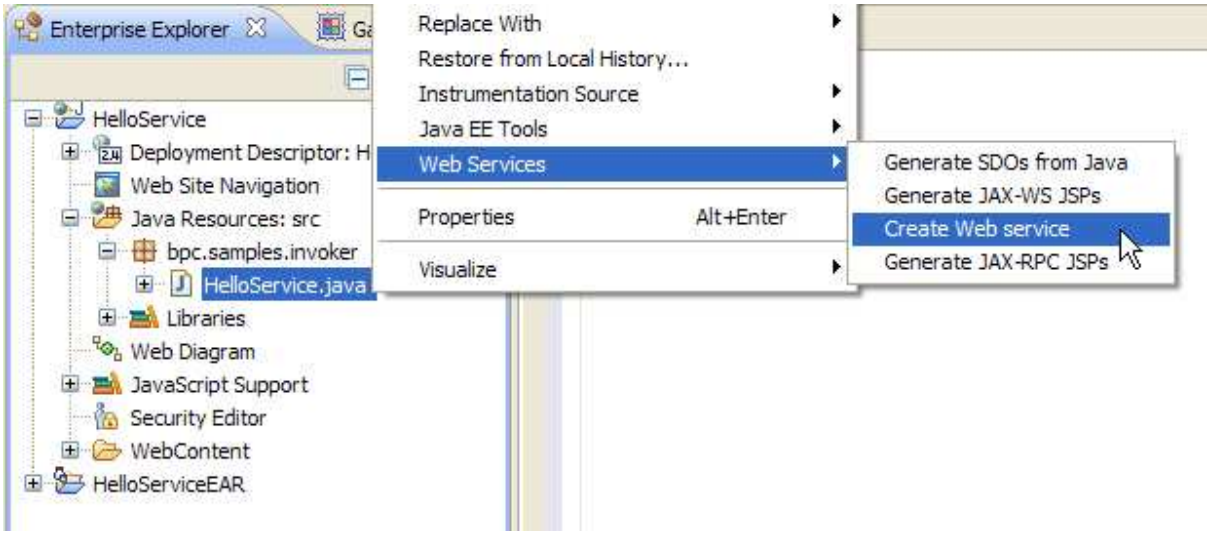
```
public int getCreditRating(String personId, String firstName, String
lastName) {
   System.out.println("GetCreditRating");
   int rating = 0;

   if ("1001".equals(personId)) {
         rating = 200;
   } else if ("1002".equals(personId)) {
         rating = 400;
   } else if ("1003".equals(personId)) {
         rating = 600;
   } else if ("1004".equals(personId)) {
         rating = 800;
   } else if ("1005".equals(personId)) {
         rating = 1000;
   }

   System.out.println("Rating is: " + rating);
```

```
    return rating;
}
```

7. On the toolbar, click the **Save** button .

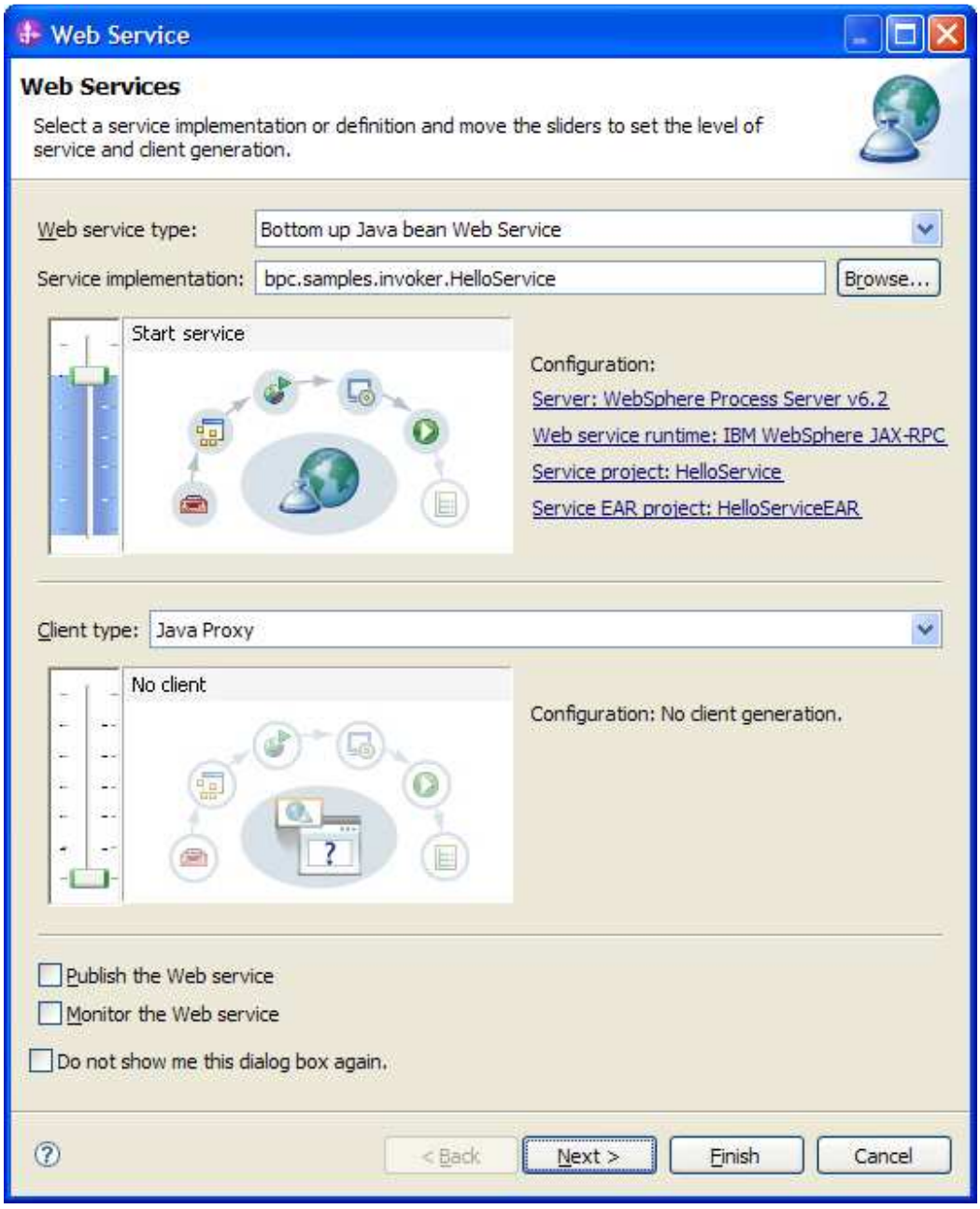
3) Generate a Web Service for a Java class

8. In the **Enterprise Explorer** view, expand **CreditRatingService > Java Resources > bpc.samples.invoker**.
9. Right-click **CreditRatingService.java**.
10. From the pop-up menu, select **Web Services > Create Web service**.



The Web Service window opens.

11. In the **Web service type** section, make sure *Bottom Up Java bean Web Service* is selected.
12. Accept all other values.



13. Click **Next**.
14. On the **Service Endpoint Interface Selection** page, accept the default values.
15. Click **Next**.
16. On the **Web Service Java Bean Identity** page, accept the default values.
17. Click **Next**.
18. Wait for the Web Service publishing to finish.
19. On the **Web Service Publication** page, accept the default values.
20. Click **Finish**.

4) Import the Web Service interface to the BankLib library

To import the ***CreditRatingService*** interface to the BankLib library, complete the following steps:

1. Switch to the Business Integration view.
2. Find out your current workspace directory by clicking **File > Switch Workspace**. You may copy this path for later use into your clipboard, then click **Cancel**.
3. In the navigation tree, expand **BankLib** and right-click **Interfaces**.
4. From the pop-up menu, select **Import**.
5. Expand **Business Integration** and select **WSDL/Interface** as input source.
6. Click **Next**. The **WSDL/Interface Import** page opens.
7. In the **Input from** field, paste the name of your workspace directory.
8. Browse to the **CreditRatingService** web project, choose folder **CreditRatingService/WebContent/WEB-INF/wsdl**. Click **OK**
9. Select **CreditRatingService.wsdl**.
10. Click **Finish**.

4. Create a Process

1) Create a new module

1. In the *Business Integration view*, click the link under **Projects** to add a business integration project.
2. The New Business Integration Project window opens. Make sure **Create a module** is selected.
3. Click **Next**.
4. In the **Module Name** field, enter ***OpenBankAccount***.
5. Click **Finish**. The module will be created. Wait until the automatic build process has finished.
6. Double-click on **Dependencies** under the project **OpenBankAccount**.
7. In the **Dependencies editor**, in the **Libraries** section, add the **BankLib** library project.

2) Define a data type (for input)

To create a data type, complete the following steps:

1. In the business integration view, right-click **Data Types.**
2. From the pop-up menu, select **New > Business Object**.
3. In the New Business Object window, enter values for **Name**, **Folder**, and **Namespace**. See the table below for the name of the Business Object and namespace, use **bpc/samples/flow** for the folder.

4. Click **Finish**.

To add an attribute to a business object, complete the following steps:

1. In the business object editor, click the **Add Attribute** button  .
2. To edit the name, click the name in the business object figure. See the table below for the name and type.
3. To edit the type, click the type in the business object figure, then select a type from the list.

In the folder **bpc/samples/flow**, create the business object that is listed in the following table:

Business Object	Namespace	Attribute Name	Attribute Type
AccountRequestData	http://bpc/samples/flow	personId	string
		firstName	string
		lastName	string
		dateRequested	date

To save your changes, click the **Save** button on the toolbar.

3) Define a data type (for output)

Repeat the same steps as in the previous part for the following table:

Business Object	Namespace	Attribute Name	Attribute Type
AccountInfo	http://bpc/samples/flow	accountCreated	boolean
		reason	string
		accountSavingsNo	string
		accountCheckingNo	string

4) Define an interface

To create an interface, complete the following steps:

1. In the business integration view, right-click **Interfaces**.
2. From the pop-up menu, select **New > Interface**.
3. In the Create a new interface panel, use the default mechanism for the **Namespace** and use **bpc/samples/flow** as the **Folder**; as the **Name** of the interface, use ***OpenBankAccount***.
4. Click **Finish**.

To add a request response operation to an interface, complete the following steps:

1. In the interface editor, click the **Add Request Response Operation** button .
2. To edit the name of the operation, click the operation name. See the table below for the operation name.
3. Each new request response operation contains one input message by default. To add an additional input message, select the operation and click the **Add Input** button .
4. To edit the name of an input message, click the message name. See the table below for the message name.
5. To edit the type of an input message, click the message type, and select a type from the list. If you want to assign a custom business object that is not already included in the list as the type, select **Browse**. In the **Data Type Selection** dialog, browse for and select a business object. See the table below for the type.
6. Each new request response operation contains one output message by default. To add an additional output message to an operation, select the operation and click the **Add Output** button .
7. Repeat steps 4 and 5 to edit the name and type of the output message.

In the folder *bpc/samples/flow*, create the interface that is listed in the following table:

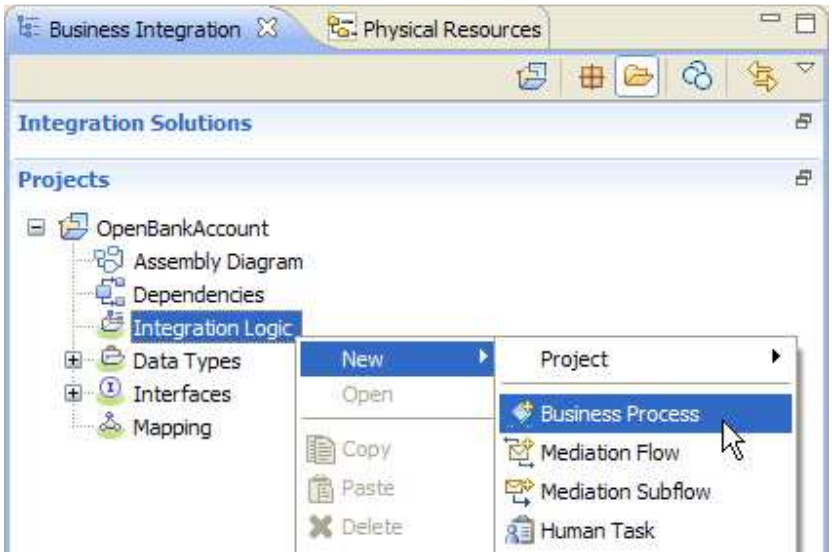
Interface	Namespace	Operation Name	Message	Name	Type
OpenBankAccount	http://bpc/samples/flow/OpenBankAccount	openAccount	Input	accountRequest	AccountRequestData
			Output	bankResponse	AccountInfo

To save your changes, click the **Save** button on the toolbar.

5) Develop the business process

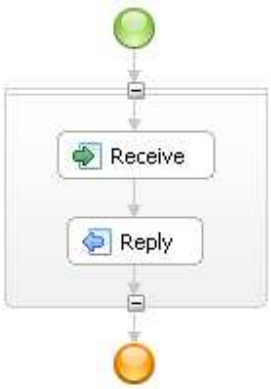
To create a new business process, complete the following steps:

1. Switch to the Business Integration view.
2. In the navigation tree, right-click **OpenBankAccount > Integration Logic**.
3. From the pop-up menu, select **New > Business Process**.

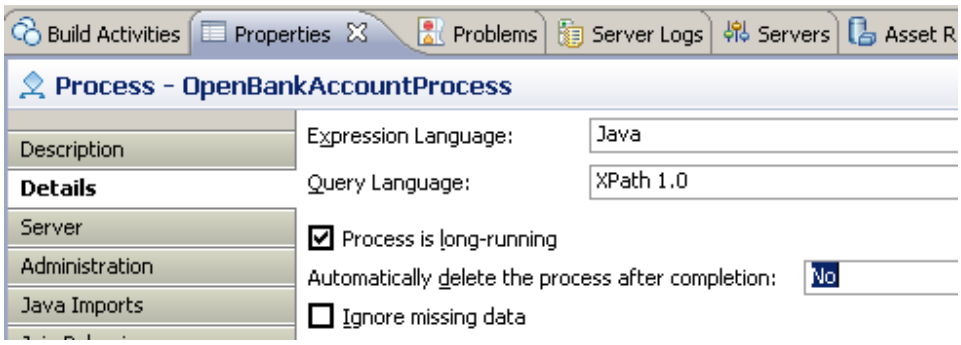


The New Business Process window opens.

4. Accept the default value for the namespace.
5. In the **Folder** field, enter ***bpc/samples/flow***.
6. In the **Name** field, enter ***OpenBankAccountProcess***.
7. Click **Next**.
8. In the Select a Business Process Type panel, select **Long-running process**.
9. Click **Next**.
10. In the Select an Interface panel, select **Select an interface**.
11. Click **Browse**. The Interface Selection window opens.
12. Select **OpenBankAccount**, then click **OK**.
13. Select Operation **openAccount**.
14. Click **Finish**.
15. Add component **Scope** from the palette to the canvas.
16. Move activities **Receive** and **Reply** to the scope, so the diagrams looks like in the picture.



17. Examine the properties of both activities.
18. Click on the empty space of the process to see its properties. In the properties, set **Automatically delete the process after completion** to **No**.

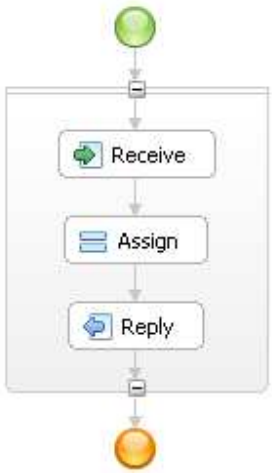


6) Add the created web service (CreditRatingService) to the process

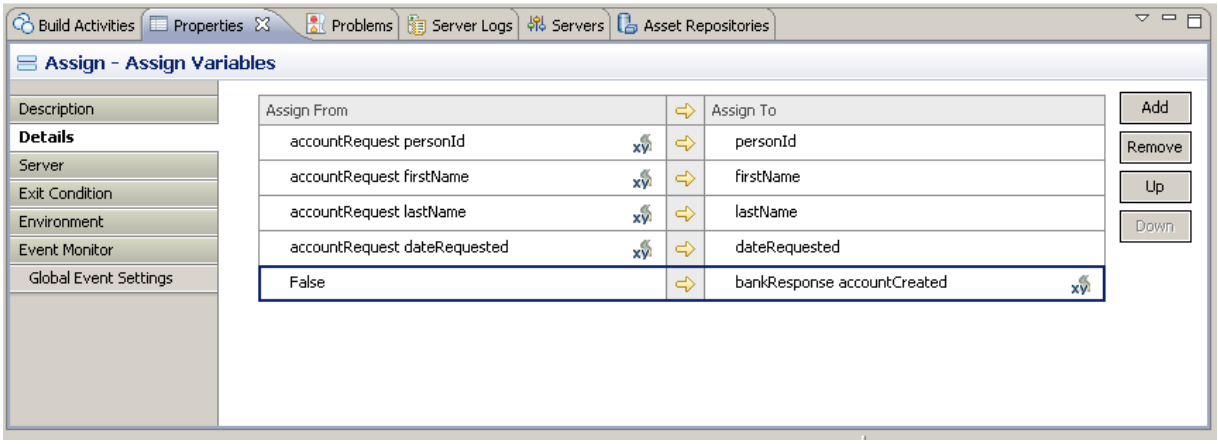
1. Create four variables (on the right side of the screen). Names and data types of these variables are in the table below.

Attribute Name	Attribute Type
personId	string
firstName	string
lastName	string
dateRequested	date
creditRating	int
accountConfirmed	boolean

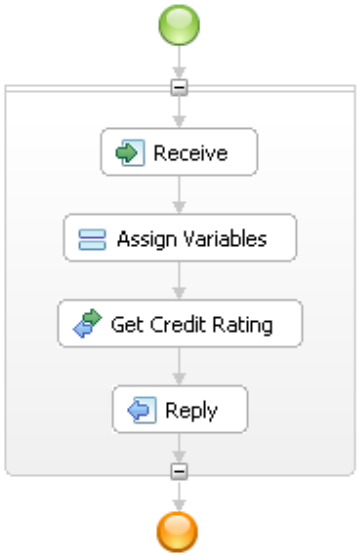
2. Drag the **Assign** activity from the palette to the flow so it looks like it is displayed in the picture.



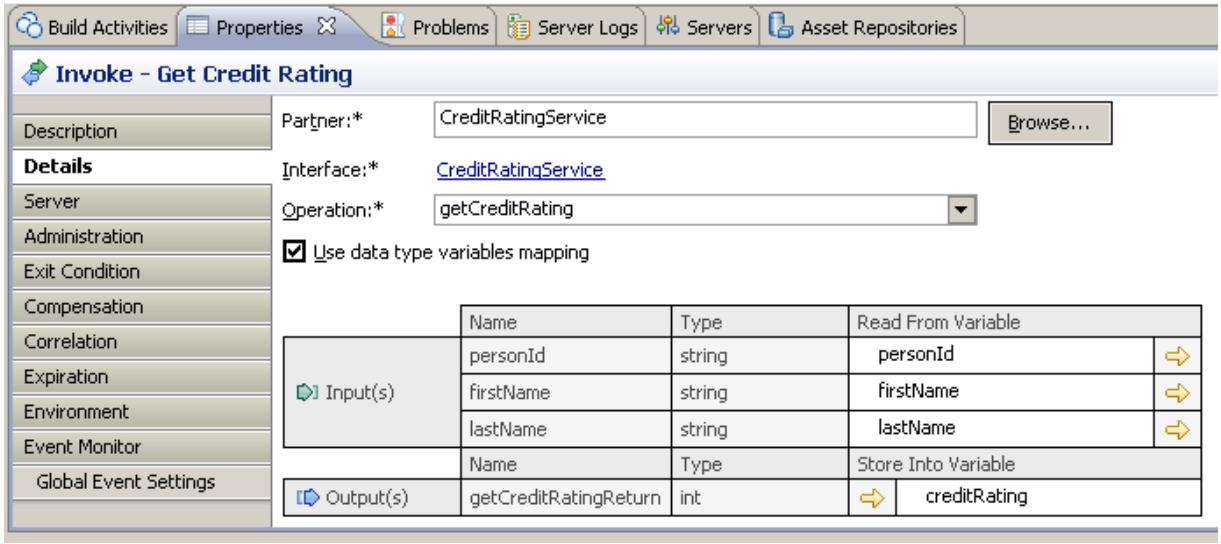
3. Rename the *Assign* activity to **Assign variables**. We are setting the display name.
4. Copy properties of the input object to the new variables in the **Assign variables** activity, as displayed in the picture.



5. Drag the **CreditRatingInterface** from the **BankLib** project to the empty area of the scope activity. The reference partner (right side of the screen) has been added.
6. On the right side, expand **CreditRatingService** reference partner, then **CreditRatingService** web service. Finally expand **getCreditRating** operation and drag it into the **Scope** container.
7. Rename the task from *Invoke* to **Get Credit Rating**. We are setting the display name. The process now looks like this.



8. In the properties of the **Get Credit Rating** activity, set the input and output variables, like displayed in the picture.



7) Add another activities to the process - Snippets

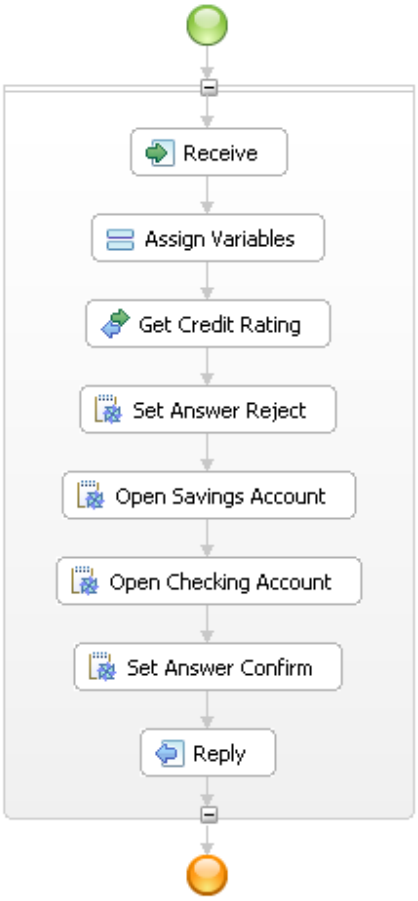
Follow these steps to add snippets to the **Scope** activity:

1. Click the snippet icon on the palette.
2. Click the empty area of the scope activity. The snippet will be added.
3. In the name field of the activity, enter the name. See the table below for the name.

Activity Name	Display Name	Activity Type	Symbol

SetAnswerReject	Set Answer Reject	Snippet	
OpenSavingsAccount	Open Savings Account	Snippet	
SetAnswerConfirm	Set Answer Confirm	Snippet	
OpenCheckingAccount	Open Checking Account	Snippet	

The Scope Activity figure now looks like this:

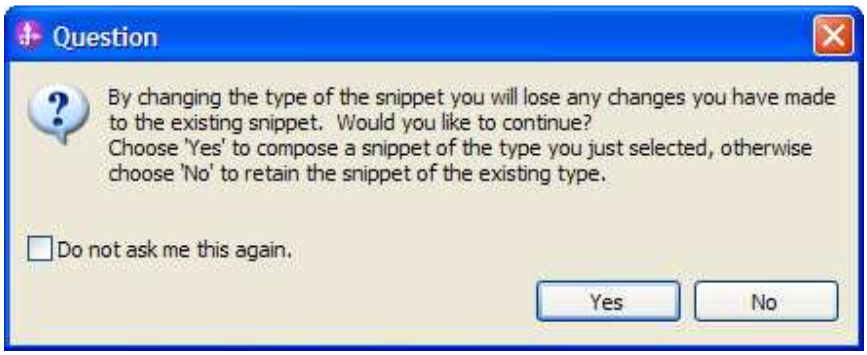


Snippet

To implement a snippet activity, complete the following steps:

1. In the business process editor, click the snippet.
2. In the Properties view, click the **Details** tab.
3. Select **Java**.

4. If the **Question** dialog appears, click **Yes**.



5. In the text area, enter the Java code.
6. On the toolbar, click the **Save** button .

Complete the implementation of the following snippet activities:

SetAnswerReject

```
System.out.println("SetAnswerReject");
bankResponse.setBoolean("accountCreated", false); //rejected
bankResponse.setString("reason", "Your request to open an account was
rejected.");
```

OpenSavingsAccount

```
System.out.println("OpenSavingsAccount");
java.util.Random rand = new java.util.Random();
int num = rand.nextInt(10000);
bankResponse.setString("accountSavingsNo", (new
Integer(num)).toString());
```

OpenCheckingAccount

```
System.out.println("OpenCheckingAccount");
java.util.Random rand = new java.util.Random();
int num = rand.nextInt(10000);
bankResponse.setString("accountCheckingNo", (new
Integer(num)).toString());
```

SetAnswerConfirm

```
System.out.println("SetAnswerConfirm");
bankResponse.setBoolean("accountCreated", true); //confirmed
bankResponse.setString("reason", "Your request has been accepted.");
```

8) Create a human task

Human task will be used by bank officers. We will only involve people in this process when the score is in the middle range (400-700).

1. As discussed in 4.4, create an interface ***ConfirmBankAccount*** in the project **OpenBankAccount** regarding the next table. This interface will serve as interface of the new Human Task.

Interface	Namespace	Operation Name	Message	Name	Type
ConfirmBankAccount	http://bpc/samples/flow/ConfirmBankAccount	confirmAccount	Input	accountRequest	AccountRequestData
			Input	creditRating	int
			Output	confirmed	boolean

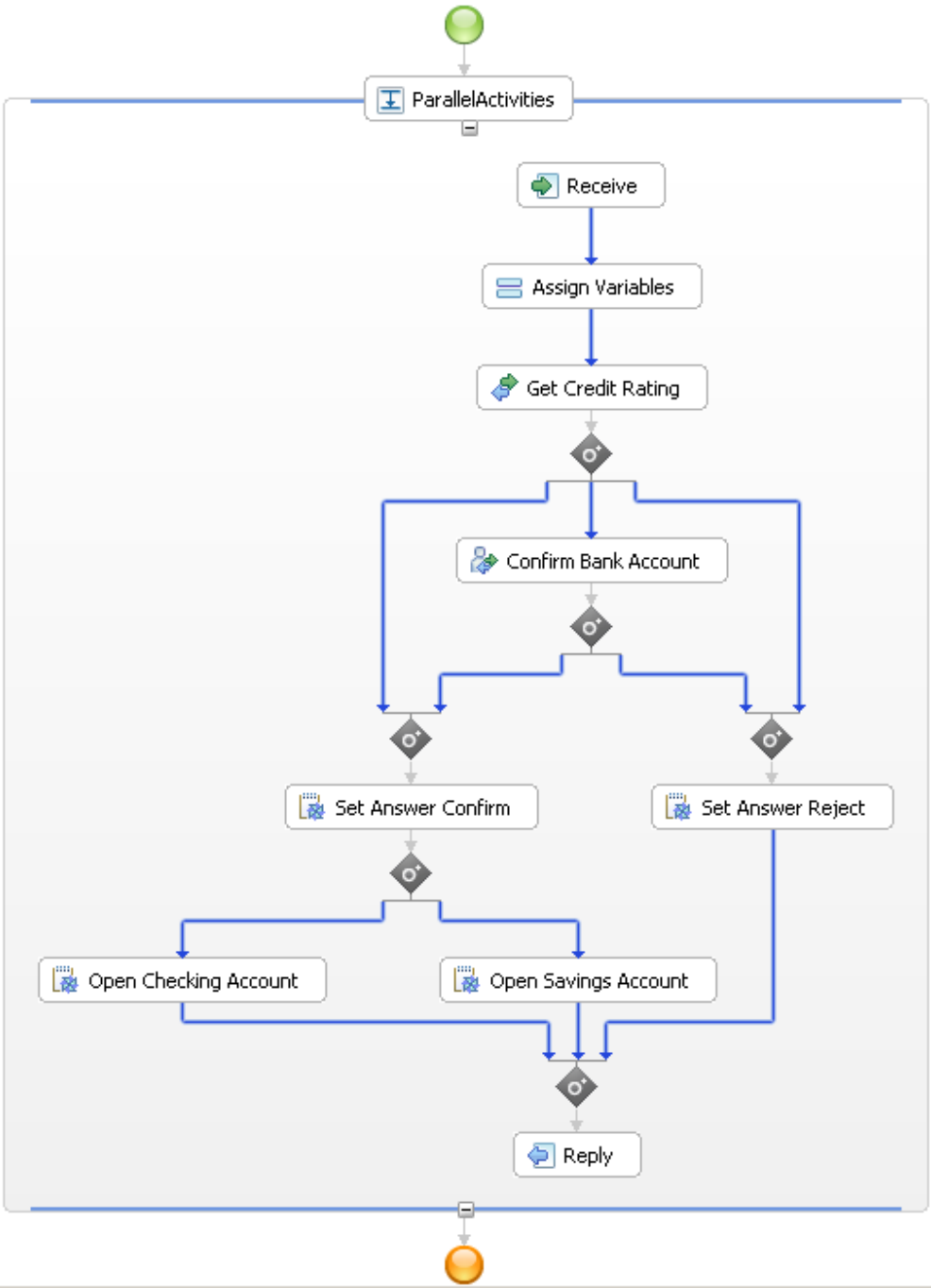
2. Drag **Human task** activity to the **Scope** task.
3. In the dialog, choose **ConfirmBankAccount** as an interface.
4. Rename the task from **OpenBankAccountProcessTask1** to ***Confirm Bank Account***.

9) Add control links between activities

Because we now want multiple branches in the process, we substitute the **Scope** task with the **Parallel Activities** task and connect the activities in appropriate way.

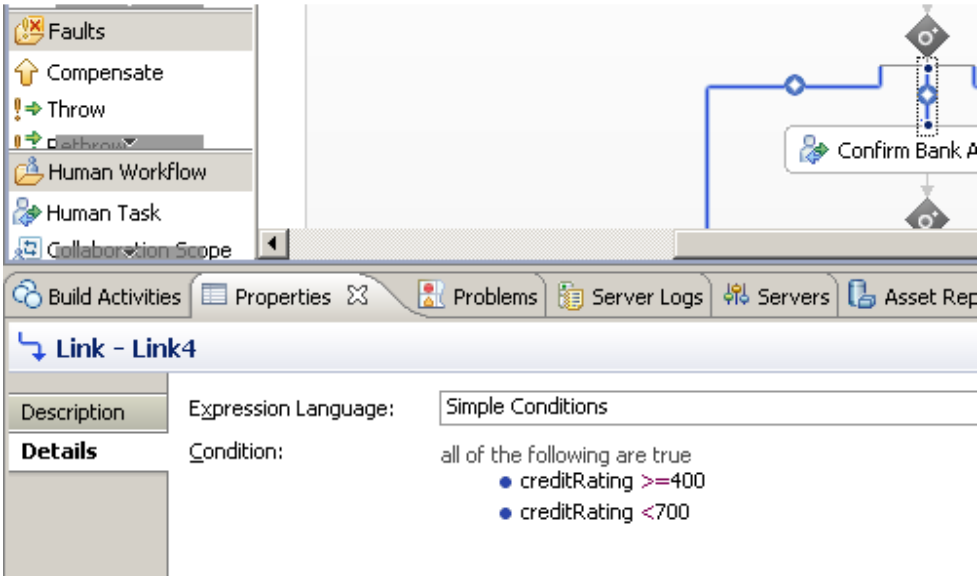
1. Move **Parallel Activities** to the process
2. Move activities from Scope to **Parallel Activities**

3. Put cursor over an activity, you will see orange circle. Click on the circle and drag the connection to another task. Repeat this task, so the overall process looks like similar as the picture below.
4. Use **Context menu** > **Arrange Parallel Activities Contents** to organize paths between activities.
5. Delete empty **Scope** activity



6. Add **conditions** to some of the paths (**together 5 conditions**), so that:
    a. The accounts are automatically created when the score is 700 or more.
    b. Human task is used when the score is >= 400 and < 700.
    c. The request is automatically rejected for scores 0 to 399.
    d. After the human task, the accounts are created only if it is approved by the officer (based on the value of **accountConfirmed** variable).

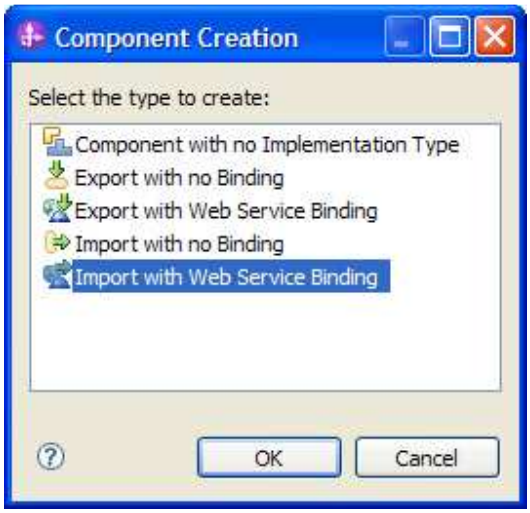
Hint: To set the conditions, click on the connection between tasks and set the condition. Here is an example picture:



10) Module assembly

To finish the implementation of the process, you have to create the module assembly. Complete the following steps:

1. In the Business Integration view, double-click **OpenBankAccount > Assembly Diagram**. The assembly editor opens.
2. In the Business Integration view, expand **OpenBankAccount > Integration Logic > Processes > bpc/samples/flow**.
3. Drag and drop the **OpenBankAccountProcess** process to the assembly editor.
4. In the Business Integration view, expand **BankLib > Interfaces**.
5. Drag the **CreditReportService** interface to the assembly editor. The Component Creation window opens.
6. Select **Import with Web Service Binding**.



7. Click **OK**.
8. A new window **CreditRatingServiceImport1 Web Service Import...** opens.
9. Select **Use an existing web service port**. Click **Browse**.
10. From the popup, select **CreditRatingService** and click **OK**.
11. When the **Transport Selection for CreditRatingServiceImport1** window appears click **OK**.
12. Click **OK**.
13. The import is added to the editor.
14. In the assembly editor, click **CreditRatingServiceImport1** to select the new import element.
15. Click again on the import element and overtype the default name **CreditRatingServiceImport1** by CreditRatingService.
16. Wire the **1..1 section** of **OpenBankAccountProcess** component and **CreditRatingService** component.
17. On the toolbar, click the **Save** button .

5. Test the business process in the integrated test environment

1) Install the OpenBankAccountProcess process to the test environment

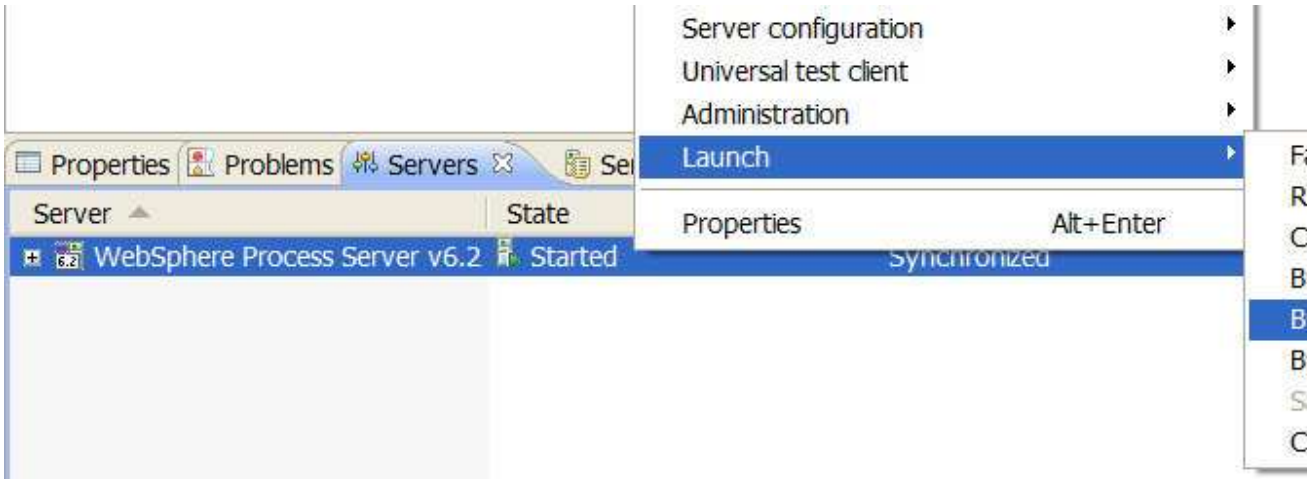
1. In the Servers view, click the **Servers** tab.
2. In the table, right-click **WebSphere Process Server v6.2**.
3. From the pop-up menu, select **Add and Remove Projects**. The **Add and Remove Projects** window opens.
4. Click **Add all**.
5. Click **Finish**.

2) Run the sample using BPC Explorer

To run this sample, use the Business Process Choreographer Explorer.

To start BPC Explorer in the integrated test environment of your WebSphere Integration Developer, follow these steps:

1. In WebSphere Integration Developer, switch to the Servers view.
2. In the table, right-click **WebSphere Process Server v6.2**.
3. From the pop-up menu, select **Launch > Business Process Choreographer Explorer**.



OR

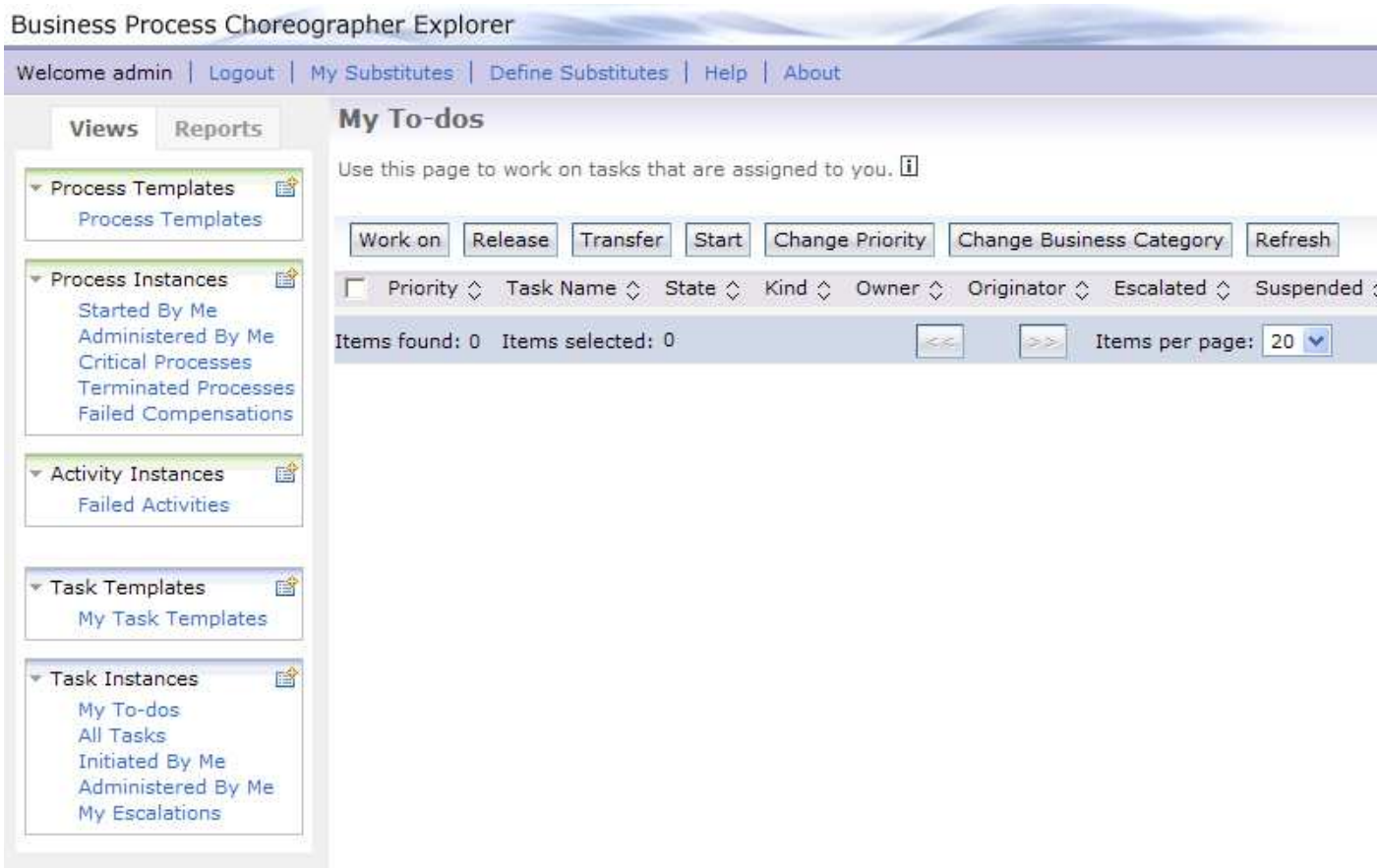
In a web browser, open URL http://*yourhost*:*yourport*/bpc.

> Note: *yourhost* has to be replaced with your server's internet address, *yourport* has to be replaced with the port of your server's HTTP transport, for example http://localhost:9080/bpc.

If security is enabled, the BPC Explorer opens with a login screen:

1. Enter a valid User ID and Password. The default User ID is **admin**, password is **admin**.
2. Click **Login**.

The BPC Explorer opens:



3) Start the sample business process

To start the invoker process, follow these steps:

1. Switch to the BPC Explorer.
2. In the Process Templates Section, click **Process Templates**.

   The Process Templates view is displayed:

3. Select **OpenBankAccountProcess**, then click **Start Instance**.
4. Fill in the **name of the process**, so you can identify it later.
5. Fill in the **input**. You can leave the **dateRequested** field empty.
6. Click **Submit**.

Hint: Based on CreditRatingService implementation, use 1001, 1002, 1003, 1004, or 1005 values as personIds. First name and last name can be whatever you like, the date does not need to be filled. For clients with personIds 1002 and 1003, the human task is invoked.

4) Output messages

Inspect the SystemOut.log of your WebSphere Process Server or the Console within WebSphere Integration Developer.

6. Final steps

- **Remove all applications** from WebSphere Process Server
- **Stop WebSphere Process Server**