

IBM® WebSphere® Application Server V7 – LAB EXERCISE

Developing a basic Java EE 5 application with Rational Application Developer

What this exercise is about	1
Lab requirements	2
What you should be able to do	2
Exercise instructions	2
Part 1: Prepare Rational Application Developer	3
Part 2: Create a Java EE 5 project	6
Part 3: Configure the sample database	10
Part 4: Create entities and session beans in the EJB project.....	12
Part 5: Create a Web interface to the application	17
Part 6: Deploy the application to the server	29
Part 7: Test the application	30
Part 8: Clean up	32
What you did in this exercise	33

What this exercise is about

The objective of this lab is to provide you with an understanding of the new tools in Rational Application Developer V7.5 that enable you to create Java EE 5 applications and deploy them to WebSphere Application Server V7.

You will create a basic Java EE5 application that uses EJB 3.0, JPA, JSP, and JSF components, and deploy that application to WebSphere Application Server V7.

The application is a basic banking application. It supports creating new customers, creating new accounts, and transferring funds between accounts.

This exercise is designed to teach you how to use the new tools in Rational Application Developer, and is not a primer on coding Java EE 5 applications. The majority of the code in this exercise is generated automatically by Rational Application Developer. The session bean logic is provided for you in code snippets, which you can examine to see how they work.

The primary purpose of this exercise is to demonstrate the use of the new Java EE 5 tools in Rational Application Developer V7.5. It is not necessarily intended to demonstrate "best practices" for overall application design.

This lab is provided **AS-IS**, with no formal IBM support.

Lab requirements

List of system and software required for the student to complete the lab.

- WebSphere Application Server V7
- Rational Application Developer for WebSphere Software V7.5
- The “Labfiles70” directory, which can be downloaded from the IBM Education Assistant Web site (<http://www.ibm.com/software/info/education/assistant/>)

What you should be able to do

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

- Configure Rational Application Developer for WebSphere Software to work with an existing installation of WebSphere Application Server.
- Create and configure a Derby database instance using Rational Application Developer’s data tools.
- Use the new JPA tools in Rational Application Developer to create JPA Entities from a set of database tables.
- Use the EJB wizard to create EJB 3.0 session beans.
- Build a Web application to access and modify the database using the Web tools and JSF components.
- Deploy an application to WebSphere Application Server.

Exercise instructions

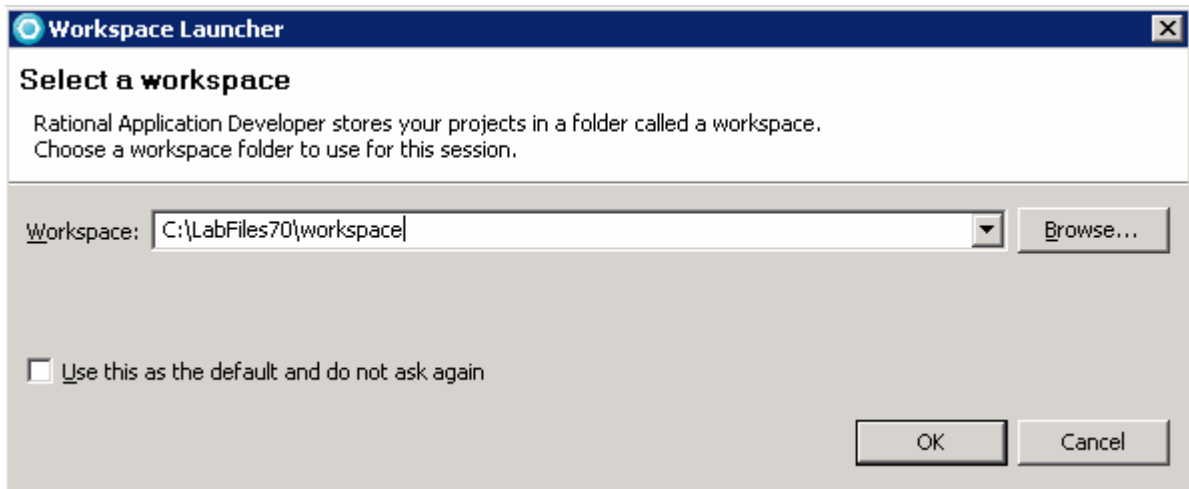
Some instructions in this lab are Windows[®] operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to run the appropriate commands and use appropriate files (.sh or .bat) for your operating system. The directory locations are specified in the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

Reference variable	Windows location	UNIX [®] or Linux [™] locations
<WAS_HOME>	C:\WebSphere\AppServer	/usr/WebSphere/AppServer /opt/WebSphere/AppServer
<RAD_HOME>	C:\Program Files\IBM\SDP	
<LAB_FILES>	C:\Labfiles70	/tmp/Labfiles60
<TEMP>	C:\temp	/tmp

Part 1: Prepare Rational Application Developer

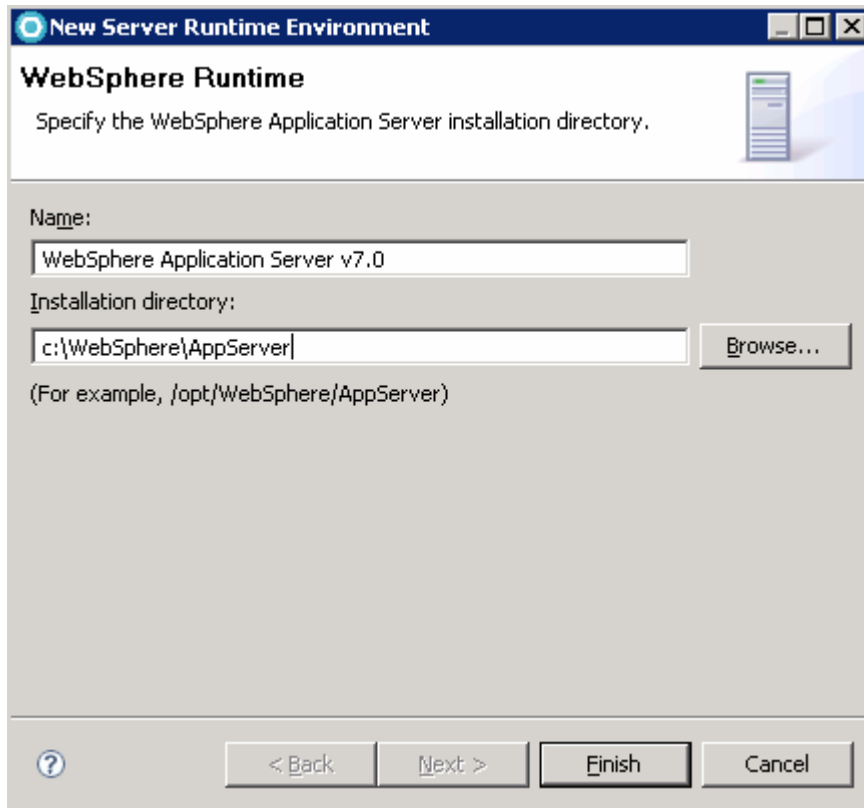
In this section you will create a new Rational Application Developer workspace, and configure Rational Application Developer to work with an existing local installation of WebSphere Application Server V7.

- ___ 1. Launch Rational Application Developer
 - ___ a. Open Rational Application Developer from the “IBM Software Delivery Platform” program group in the Start menu.
- ___ 2. When prompted for a workspace, as shown in the screen capture below, enter **C:\LabFiles70\workspace**, and click **OK**.



- ___ 3. When the workspace has been created, you will be presented with the “Welcome” tab. Note a lot of useful information is available from this tab, including several sample applications. These samples will not be used in this exercise, but you may want to return to them later by selecting “Welcome” from the Help menu. For now, **close** the Welcome tab by clicking the “x” on the tab header.
- ___ 4. Configure the local installation of WebSphere Application Server as a test server in Rational Application Developer.
 - ___ a. Click the **Servers** tab on the bottom panel.
 - ___ b. Right-click in the empty pane, and select **New > Server**.
 - ___ c. Click **Add...** in the “Define a new server” window.

- ___ d. Enter **C:\WebSphere\AppServer** (or the location of your WebSphere Application Server installation, if different) in the “Installation directory” field, as shown below.



- ___ e. Click **Finish**.
- ___ f. The window will close, returning you to the “Define a new server” window. Click **Next**.
- ___ g. Uncheck the **Run server with resources in the workspace** check box, so that all application resources will run on the defined server.

__ h. Enter **wsdemo** in the “Password” field. The window should be configured as shown below.

New Server

WebSphere Server Settings

Input settings for connecting to an existing WebSphere Application Server.

WebSphere profile name: [Configure profiles...](#)

Server connection types and administrative ports

Automatically determine connection settings
 Manually provide connection settings

Connection Type	Port	Default port	Description
<input checked="" type="checkbox"/> IPC	9633	9633	Recommended for local servers
<input checked="" type="checkbox"/> RMI	2809	2809	Designed to improve communicat
<input checked="" type="checkbox"/> SOAP	8880	8880	Designed to be more firewall com

Run server with resources within the workspace
 Security is enabled on this server

Current active authentication settings:

User ID:

Password:

WebSphere server name:

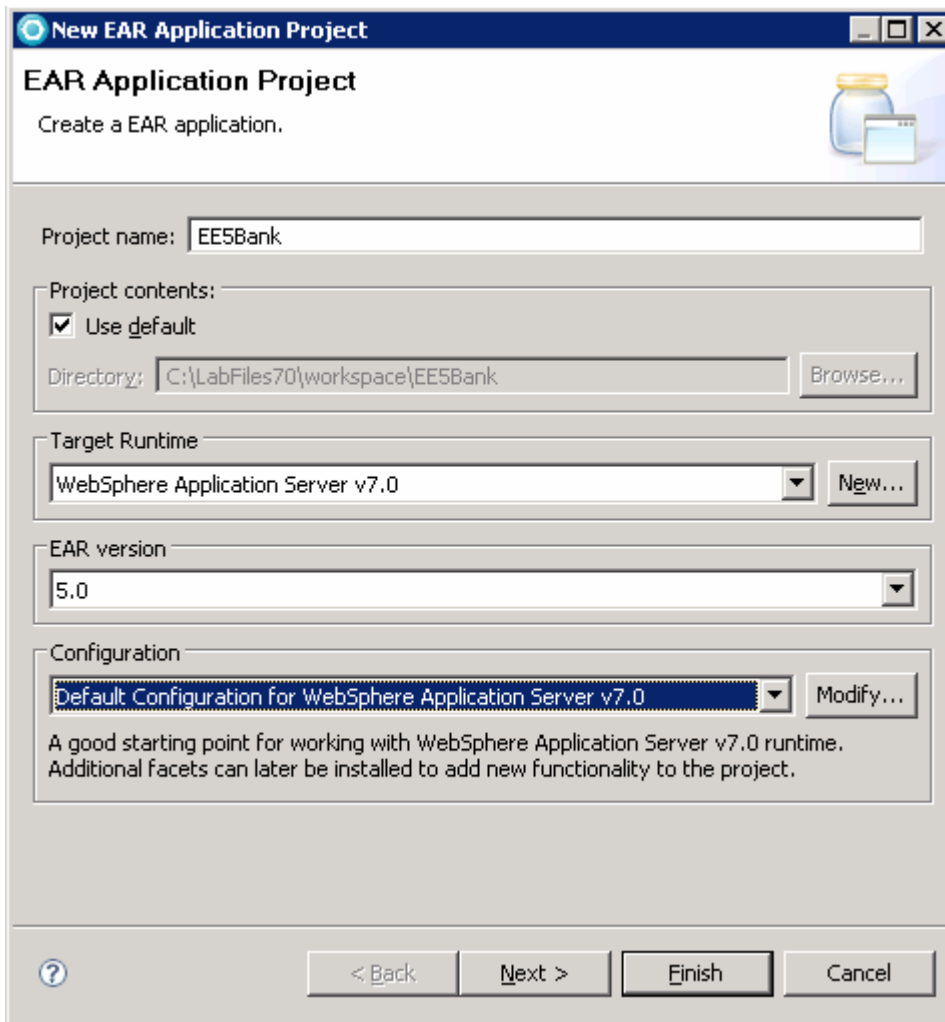
[Test Connection](#)

__ i. Click **Finish**.

Part 2: Create a Java EE 5 project

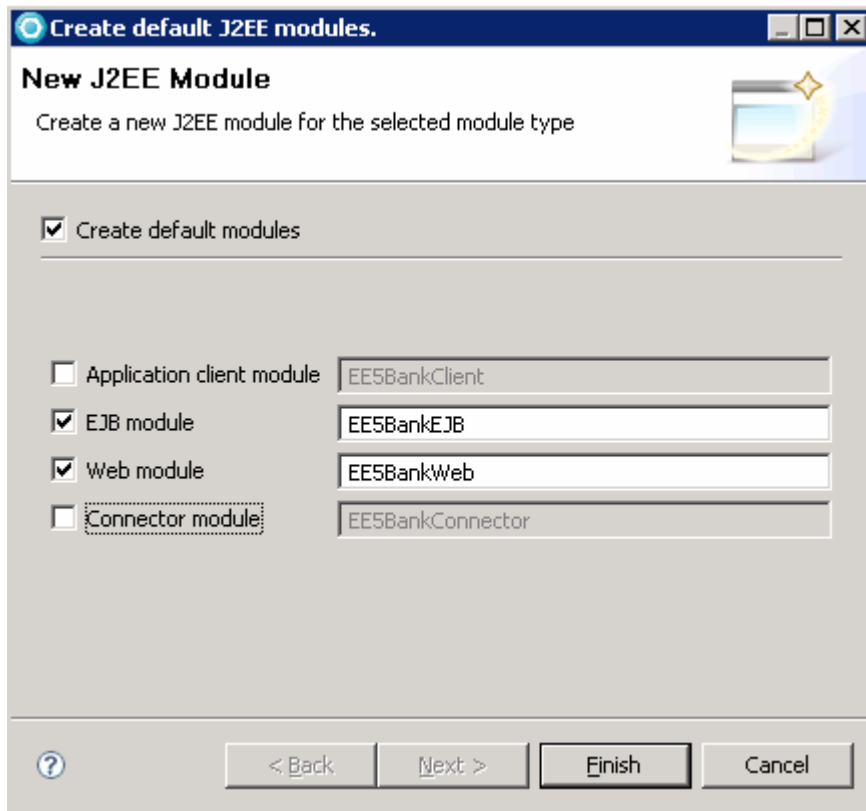
In this section you will create a Java EE 5 project, which will contain Web and EJB modules. Your project will contain neither an application deployment descriptor nor an EJB deployment descriptor, as they are not required parts of a Java EE 5 application. You will also see how to add JSF and JPA support to a project.

5. Rational Application Server starts new workspaces in the Java EE perspective. In this perspective, the rightmost panel is titled “Project Explorer”. Use this perspective to create a new Java EE 5 project.
 - a. **Right-click** in the Project Explorer panel and select **New > Enterprise Application Project**.
 - b. In the “New application EAR project” window, enter the project name **EE5Bank**.
 - c. Ensure the “Target Runtime” is **WebSphere Application Server v7.0**.
 - d. Ensure that the “EAR version” field is set to **5.0**. The window should resemble the image below.



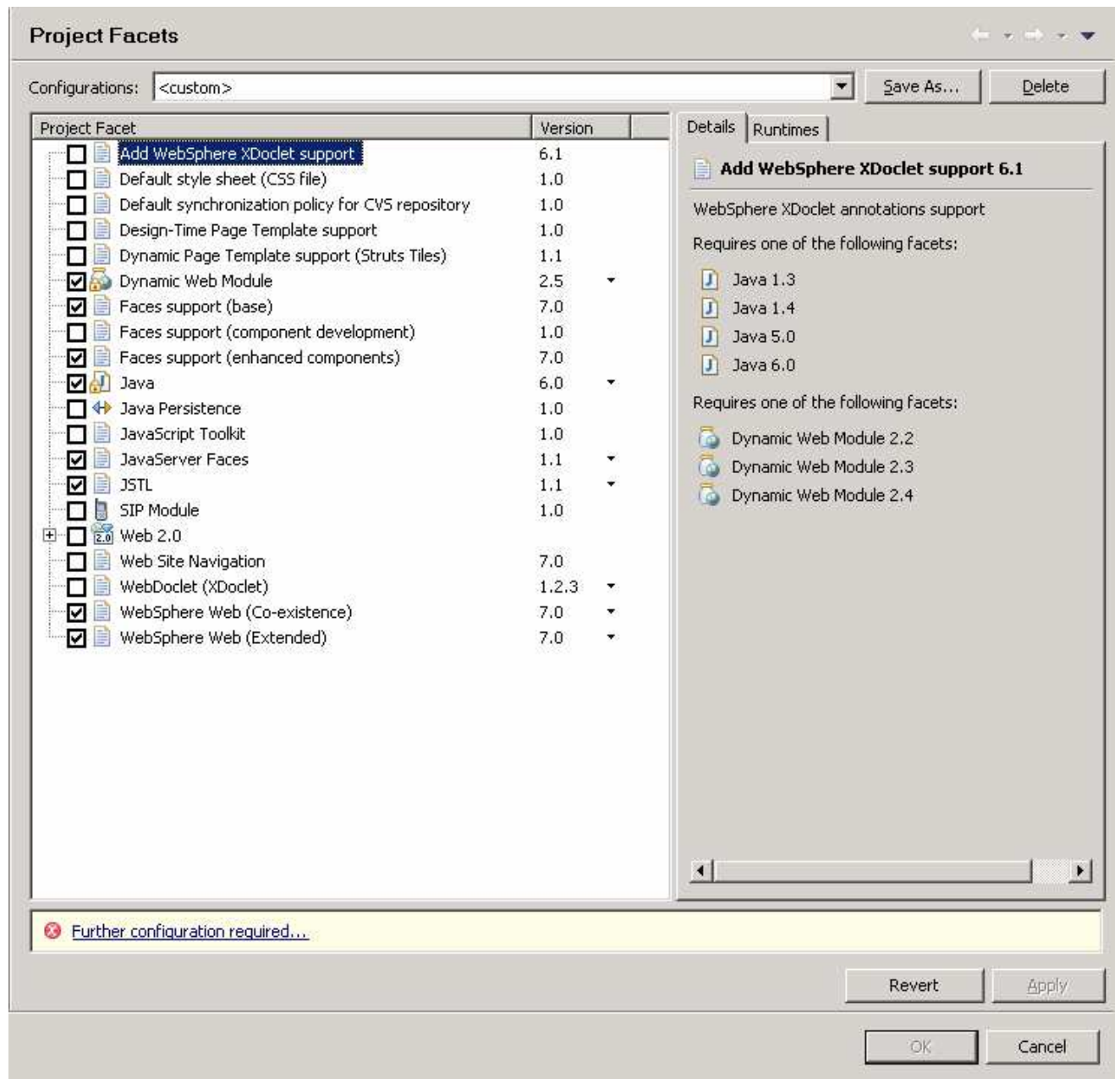
- e. Click **Next**.

- ___ f. Note that on the next panel, the “Generate Deployment Descriptor” check box is unchecked by default. This is because application deployment descriptors are optional in Java EE5. Leave the box unchecked, as you will not be using the deployment descriptor in this exercise.
- ___ g. Click **New Module...**
- ___ h. Uncheck the boxes next to **Application client module** and **Connector module**.
- ___ i. Keep the default names for the modules as shown below, and click **Finish**.



- ___ j. Click **Finish** again to complete the wizard.
- ___ 6. Add Java Persistence API support to the EJB project.
 - ___ a. Right-click **EE5BankEJB** in the Project Explorer, and select **Properties** from the contextual menu.
 - ___ b. Click **Project facets** in the left panel.
 - ___ c. Check the box next to **Java Persistence**.
 - ___ d. Click **OK** to close the properties window.
- ___ 7. Add JavaServer Faces support to the Web project.
 - ___ a. Right-click **EE5BankWeb** in the Project Explorer, and select **Properties** from the contextual menu.
 - ___ b. Click **Project facets** in the left panel.

- ___ c. Click **Modify Project**.
- ___ d. Check the boxes next to **Faces support (base)**, **Faces support (enhanced components)**, **JavaServer Faces**, and **JSTL** as shown below.




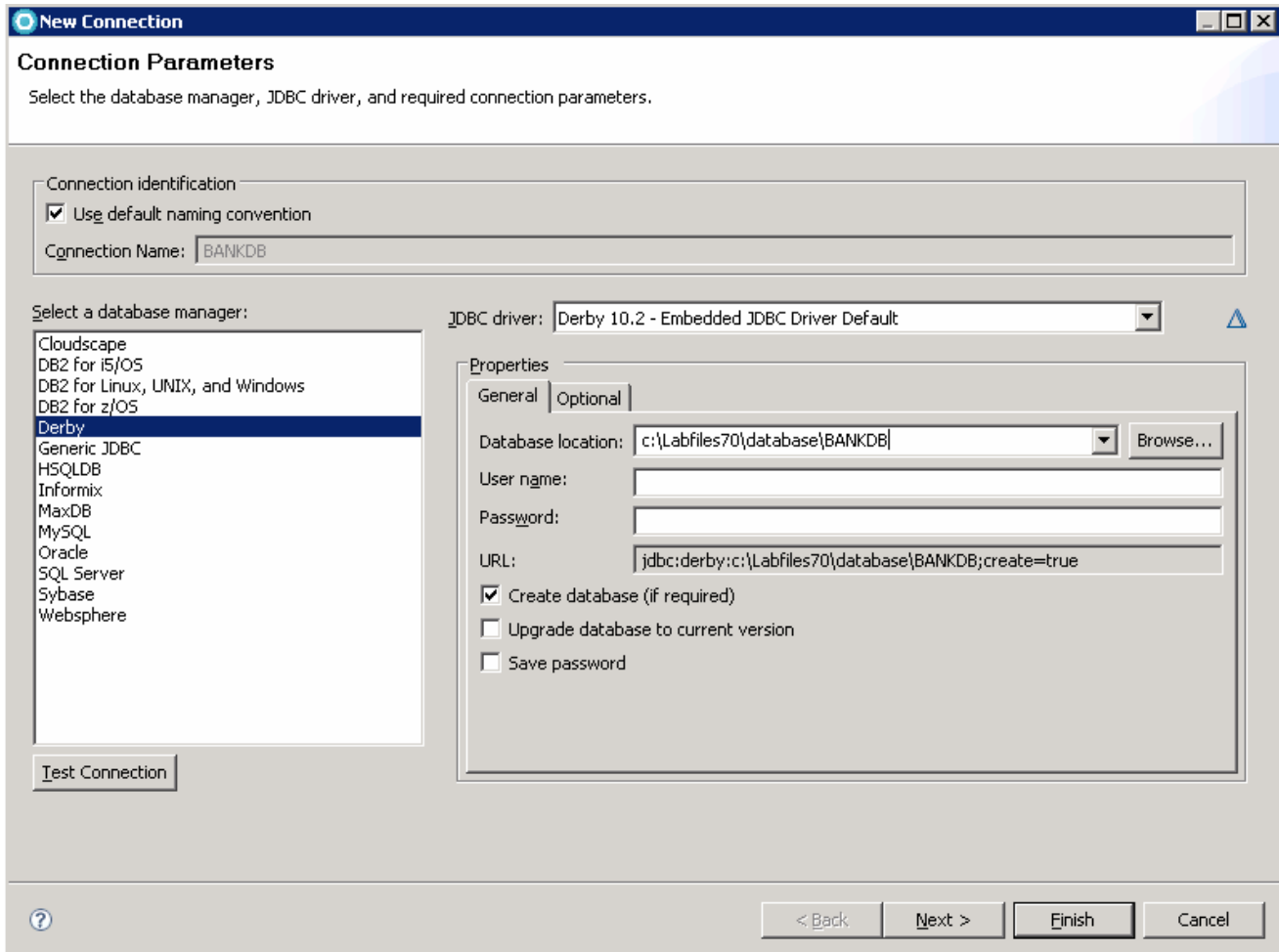
- ___ e. Click the **Further configuration required...** link.
- ___ f. Accept the default values in the “JSF configuration” window, and click **OK**.

- ___ 8. Add the EJB project as a reference to the Web project
 - ___ a. With the properties window still open, click **Project References**.
 - ___ b. Check the box next to **EE5BankEJB**.
 - ___ c. Click **OK** to apply the change and close the properties window

Part 3: Configure the sample database

In this section you will use Rational Application Server's data tools to create a Derby database, and run a DDL file to create tables in that database.

- ___ 9. Switch to the Data perspective
 - ___ a. Select **Window > Open perspective > Other...**
 - ___ b. Select **Data** and click **OK**.
- ___ 10. Create a new derby database connection.
 - ___ a. Click the **New Connection Profile**  button in the Data Source Explorer panel.
 - ___ b. Select **Derby** from the "Select a database manager" list.
 - ___ c. Select **Derby 10.2 – Embedded JDBC Driver Default** from the "JDBC Driver" menu.
 - ___ d. Enter **C:\LabFiles70\database\BANKDB** in the "Database location" field. The configuration should look the same as the picture below.



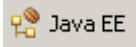
___ e. Click the **Test connection** button.

___ f. Wait for the message "Connection succeeded" to appear, and click **OK**. If the connection fails, ensure that your settings match the above picture and try again.

___ g. Click **Finish**.

___ 11. Create the tables defined in Bank.ddl

___ a. Import Bank.ddl into your EJB project.

1) Switch back to the Java EE perspective by clicking the  button at the top right.

2) Right-click **EE5BankEJB** and click **Import > Import...**

3) Expand **General** and select **Filesystem**.

4) Click **Next**.

5) Click the **Browse** button and navigate to **C:\LabFiles70\database** and click **OK**.

6) Check the box next to **Bank.ddl** and click **Finish**.

___ b. Run Bank.ddl to create database tables.

1) Double-click **Bank.ddl** in the Project Explorer to open the file.

2) From the **Run** menu, select **Run SQL**.

3) Select **BANKDB** from the "Connections" list.

4) Click **Finish**.

5) A blue check mark and the word "Succeeded" will appear in the bottom panel to confirm that the tables were created.

___ c. **Close** Bank.ddl. If prompted to save your changes, click **Save**.

Part 4: Create entities and session beans in the EJB project

In this section, you will be using the Java persistence tools to auto-generate JPA entities based on the tables that you previously created in the Derby database. You will also create session beans that manipulate the entities using the “New EJB” wizard, and paste code snippets into the session beans to add business logic. Finally, you will use a wizard to create JPA manager beans. These will be used later when you are building a Web interface to your application – JPA manager beans utility classes for accessing JPA entities, and are required if you want to use Rational Application Developer’s built in JSF components for accessing JPA data.

- ___ 12. Configure the EE5BankEJB project to use the “BANKDB” connection that you created.
 - ___ a. Right-click **EE5BankEJB** and select **Properties**.
 - ___ b. Click **JPA**.
 - ___ c. Select **BANKDB** from the “Connection” menu.
 - ___ d. Check the box next to **Override default schema from connection**.
 - ___ e. Select **BANKSCHEMA** from the “Schema” menu.
 - ___ f. Click **OK**.

- ___ 13. Use the Java persistence tools to create entities that represent the Customer and Account database tables.
 - ___ a. Right click **EE5BankEJB** in the Project Explorer and select **JPA Tools > Generate Entities...**
 - ___ b. Select **BANKDB** from the “Connection” menu.
 - ___ c. Select **BANKSCHEMA** from the “Schema” menu.
 - ___ d. Click **Next**.
 - ___ e. Enter **com.ibm.websphere.beta.ee5bank** in the “Package” field.
 - ___ f. Check the boxes next to **ACCOUNT** and **CUSTOMER**.
 - ___ g. Click **Finish**.

- ___ 14. Create a stateless session bean called Create that will be used to create customers and accounts.
 - ___ a. Right-click **EE5BankEJB** in the Project Explorer and select **New > Session bean**.
 - ___ b. Enter **com.ibm.websphere.beta.ee5bank** in the “Package” field.
 - ___ c. Enter **Create** in the “Class name” field.
 - ___ d. Check the box next to **Remote**, so that both local and remote interfaces are created.

__ e. The window should reflect the settings shown in the screen capture below.

__ f. Click **Finish**.

__ g. If a “Confirm enablement” screen appears, click **Cancel**.

__ h. Open :\\LabFiles70\\RADLab\\snippets\\snippet1_create.txt with Notepad. This file contains the business logic for the “Create” EJB.

Note: The EJB contains two methods, createAccount and createCustomer. Each method creates an object, sets some properties, and then persists it to the database using the JPA Entity Manager.

__ i. Select **Edit > Select All**, then select **Edit > Copy**

__ j. Return to Rational Application Developer and place the cursor on line 11 of Create.java (below the line that reads “Public class Create implements...”)

__ k. Select **Edit > Paste** to paste the code into the editor.

__ l. Some errors will appear in the code. To correct these errors, right-click in the source code editor, and select **Source > Organize Imports**.

__ m. A “choose type to import” window will appear. Select **javax.persistence.EntityManager** and click **Finish**.

__ n. Promote the business methods to the local interface

- 1) Right-click **Create** in the Outline panel (far right) and select **Refactor > Pull up**.
- 2) Select **com.ibm.websphere.beta.ee5bank.CreateLocal** from the "Target business interface" menu.
- 3) Check the boxes next to **createCustomer** and **createAccount**.
- 4) Click **Next**.
- 5) Click **Finish** to accept the modifications to your code.

o. Promote the business methods to the remote interface.

- 1) Right-click **Create** in the Outline panel (far right) and select **Refactor > Pull up**.
- 2) Select **com.ibm.websphere.beta.ee5bank.CreateRemote** from the "Target business interface" menu.
- 3) Check the boxes next to **createCustomer** and **createAccount**.
- 4) Click **Finish**.

p. **Save** and **close** Create.java

q. **Save** and **Close** Classdiagram.dnx

_____ 15. Create a stateless session bean called Transfer that will be used to transfer funds between accounts.

a. Right-click **EE5BankEJB** in the Project Explorer and select **New > Session bean**.

b. Enter **com.ibm.websphere.beta.ee5bank** in the "Package" field.

c. Enter **Transfer** in the "Class name" field.

d. Check the box next to **Remote**, so that both local and remote interfaces are created.

__ e. The window should reflect the settings shown in the screen capture below.

Create EJB 3.0 Session Bean

Specify class file destination.

EJB project: EE5BankEJB

Source folder: \EE5BankEJB\ejbModule

Java package: com.ibm.websphere.beta.ee5bank

Class name: Transfer

Superclass:

State type: Stateless

Create business interface

Remote com.ibm.websphere.beta.ee5bank.TransferRemote

Local com.ibm.websphere.beta.ee5bank.TransferLocal

__ f. Click **Finish**.

__ g. Open :\\LabFiles70\\RADLab\\snippets\\snippet2_transfer.txt with Notepad. This file contains the business logic for the “Create” EJB.

Note: The EJB contains one method, transferFunds. This method uses the JPA Entity Manager to get the balances from two accounts, performs the necessary arithmetic, and then sets the new values for the balance of each account.

__ h. Select **Edit > Select All**, then select **Edit > Copy**

__ i. Return to Rational Application Developer and place the cursor on line 11 of Transfer.java (below the line that reads “Public class Transfer implements...”)

__ j. Select **Edit > Paste** to paste the code into the editor.

__ k. Some errors will appear in the code. To correct these errors, right-click in the source code editor, and select **Source > Organize Imports**.

__ l. A “choose type to import” window will appear. Select **javax.persistence.EntityManager** and click **Finish**.

__ m. Promote the business methods to the local interface

- 1) Right-click **Create** in the Outline panel (far right) and select **Refactor > Pull up**.
- 2) Select **com.ibm.websphere.beta.ee5bank.TransferLocal** from the “Target business interface” menu.
- 3) Check the box next to **transferFunds**.
- 4) Click **Next**.
- 5) Click **Finish** to accept the modifications to your code.

___ n. Promote the business methods to the remote interface.

- 1) Right-click **Transfer** in the Outline panel (far right) and select **Refactor > Pull up**.
- 2) Select **com.ibm.websphere.beta.ee5bank.TransferRemote** from the “Target business interface” menu.
- 3) Check the boxes next to **transferFunds**.
- 4) Click **Finish**.

___ o. **Save** and **close** Transfer.java

___ p. **Save** and **Close** Classdiagram.dnx

_____ 16. Create JPA manager beans for your Entities. A JPA manager is a façade class that makes it easy for the Web tools to access the JPA Entities in your project.

- ___ a. Right-click **EE5BankEJB** and select **JPA Tools > Add JPA manager beans...**
- ___ b. Check the boxes next to both **Account** and **Customer**.
- ___ c. Click **Finish**.

Part 5: Create a Web interface to the application

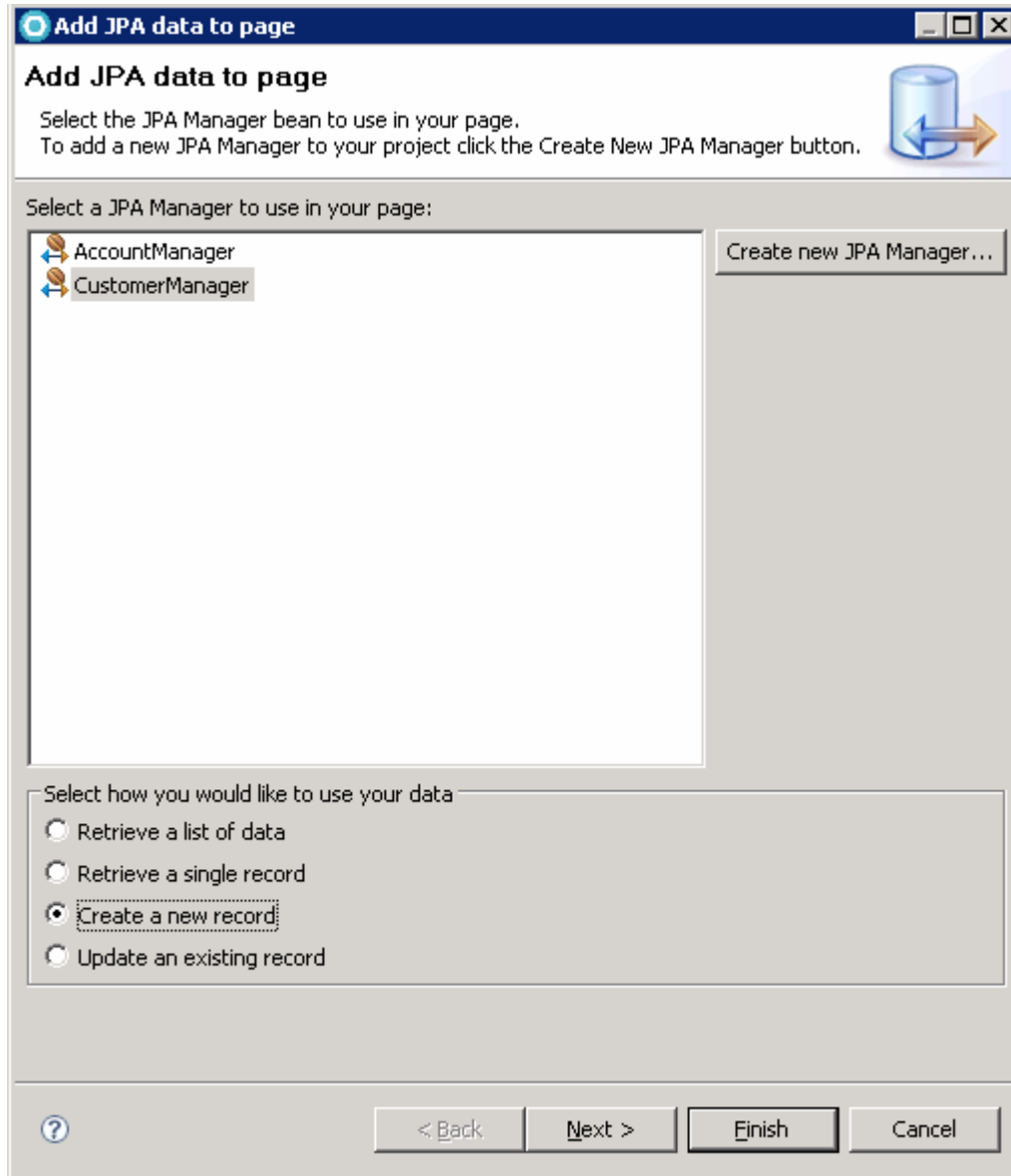
In this section, you will create several Web pages using Rational Application Developer's Web tools. These pages will be built using JavaServer Faces technology, and will use the built-in JSF components to work with session beans and JPA Entities.

- ___ 17. Open the Web perspective
 - ___ a. Navigate to **Window > Open perspective > Web**.
- ___ 18. Ensure that the classes in the **EE5BankEJB** project are available to the Web project.
 - ___ a. Right-click **EE5BankWeb** and select **Properties**.
 - ___ b. Click **Java Build Path**.
 - ___ c. Click **Add...**
 - ___ d. Check the box next to **EE5BankEJB** and click **OK**.
 - ___ e. Click **Java EE Module Dependencies**, in the left menu of the Properties window.
 - ___ f. Click the **Web Libraries** tab.
 - ___ g. Check the box next to **EE5BankEJB** and click **OK**. The Properties window will close.
- ___ 19. Create new pages by updating the Web Site Navigation.
 - ___ a. Expand **EE5BankWeb** in the Project Explorer.
 - ___ b. Double-click **Web Site Navigation** to open it in the editor.
 - ___ c. Click the **Palette** tab at the far right to activate the Web palette. (Or select **Window > Show view > Palette**)
 - ___ d. Create home.jsp
 - 1) Click-and-drag the **New page** component from the palette into the editor.
 - 2) Type **home.jsp** in the component's text field to name it.
 - 3) Double-click **home.jsp** to edit it.
 - 4) Ensure that the "File name" field reads **home.jsp**.
 - 5) Expand **Sample Templates > Family A** and select **A-01_blue.html**.
 - 6) Click **Finish**.
 - 7) A Web page editor will open. Type "**Welcome to EE5Bank**" and type **enter** three times to add carriage returns, where you see "place your page content here".
 - 8) **Save** and **Close** home.jsp
 - ___ e. Create newCustomer.jsp
 - 1) Click-and-drag the **New page** component from the palette into the editor, beneath home.jsp (you will see a black horizontal line appear below home.jsp)

- 2) Type **newCustomer.jsp** in the component's text field to name it.
- 3) Double-click **newCustomer.jsp** to edit it.
- 4) Ensure that the "File name" field reads **newCustomer.jsp**.
- 5) Ensure that **A-01_blue.html** is selected as the template.
- 6) Click **Finish**.
- 7) A Web page editor will open. Type "**Add a new customer**", and type **enter** three times to add carriage returns, where you see "place your page content here".

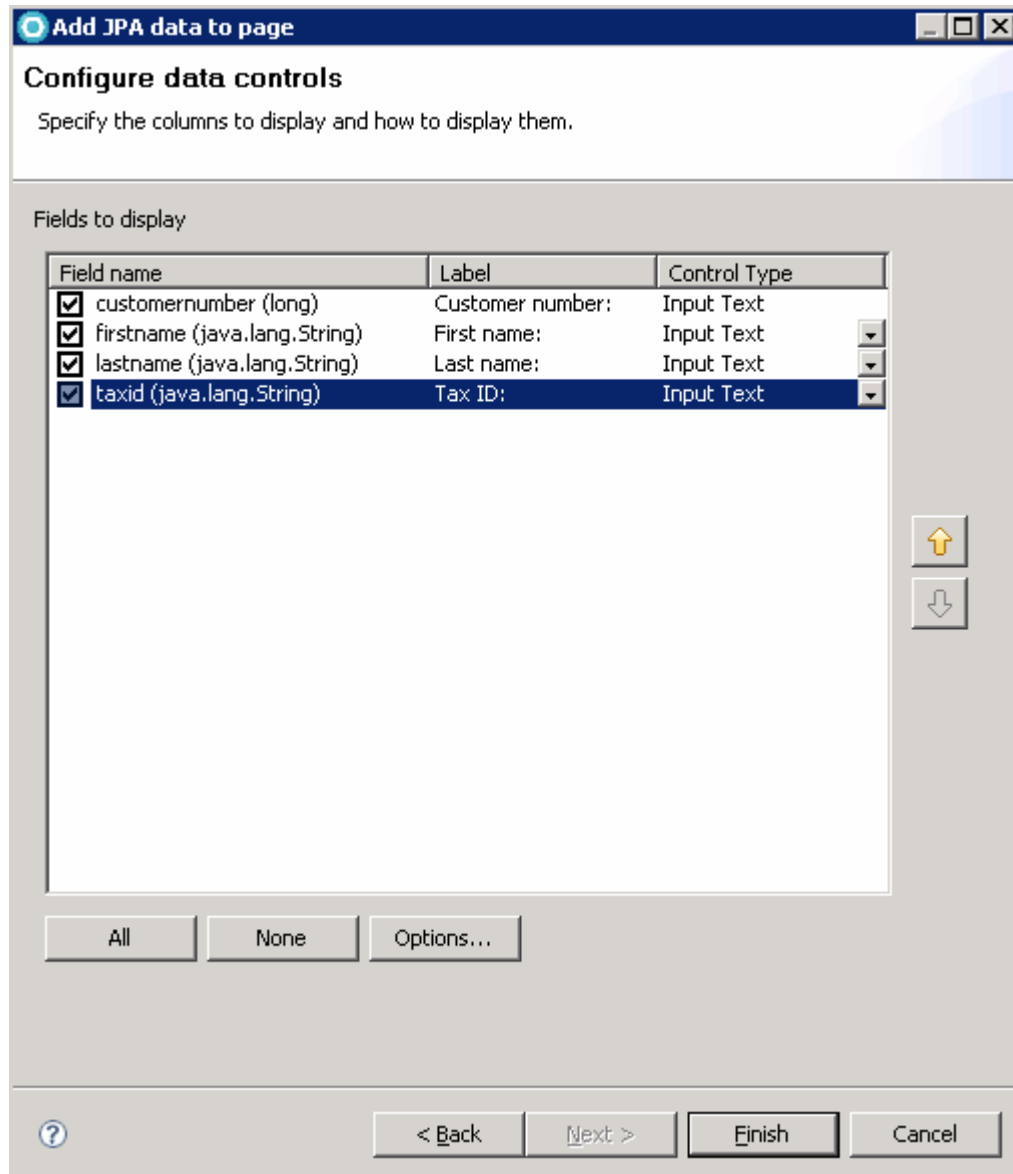
___ f. Add a JPA Data element to the page to create customers in the database.

- 1) Click **Data and Services** in the palette.
- 2) Drag the **JPA Data** element onto the page, below the text you entered.
- 3) Highlight the **CustomerManager** bean, and select **Create a new record** as shown below.



- 4) Click **Next**.
- 5) Highlight **firstname** and click the “up” arrow to move it above lastname. This order determines how the form will appear on your Web page.

- 6) Edit the "Label" fields to give each item a suitable name for display on the Web page. When finished, the window should resemble the example below.



- 7) Click **Finish**. The form that has been added to the Web page will create a new JPA Entity based on the information provided. You will test this later when the application is complete.

- 8) **Save** and **Close** newCustomer.jsp.

___ g. Create a similar page, newAccount.jsp

- 1) Click-and-drag the **New page** component from the palette into the editor, next to newCustomer.jsp (you will see a black vertical line appear next to newCustomer.jsp)
- 2) Type **newAccount.jsp** in the component's text field to name it.
- 3) Double-click **newAccount.jsp** to edit it.

- 4) Ensure that the “File name” field reads **newAccount.jsp**.
 - 5) Ensure that **A-01_blue.htpl** is selected as the template.
 - 6) Click **Finish**.
 - 7) A Web page editor will open. Type “**Add a new account**”, and type **enter** three times to add carriage returns, where you see “place your page content here”.
- ___ h. Add a session bean element to the page. This element will use a JSF form and the “Create” bean to add an account to the database. This is an alternative to using JPA manager beans as was shown in the previous createCustomer example.
- 1) Click **Data and Services** in the palette.
 - 2) Drag the **EJB Session bean** element onto the page, below the text you entered.
 - 3) Click **Add...** in the “Add session bean” window.
 - 4) Expand **EE5Bank** and **EE5Bank.jar**, and select **Create**.
 - 5) Click **Finish**.

- 6) Ensure that createAccount(long,int,int) is selected in the “Select a method” menu as shown below, and click **Next**.

Add Session Bean

Session Bean
Select a Session Bean and a method

Select a Service:

Name	Class
ejb/Create	com.ibm.websphere.beta.ee5bank.CreateLocal

Add...

Select a method:

createAccount(long,int,int)

Create input form and results display (configure details on following pages)

< Back Next > Finish Cancel

7) Edit the values in the "Label" column to give the fields acceptable display names, as shown below.

Add Session Bean

Input Form

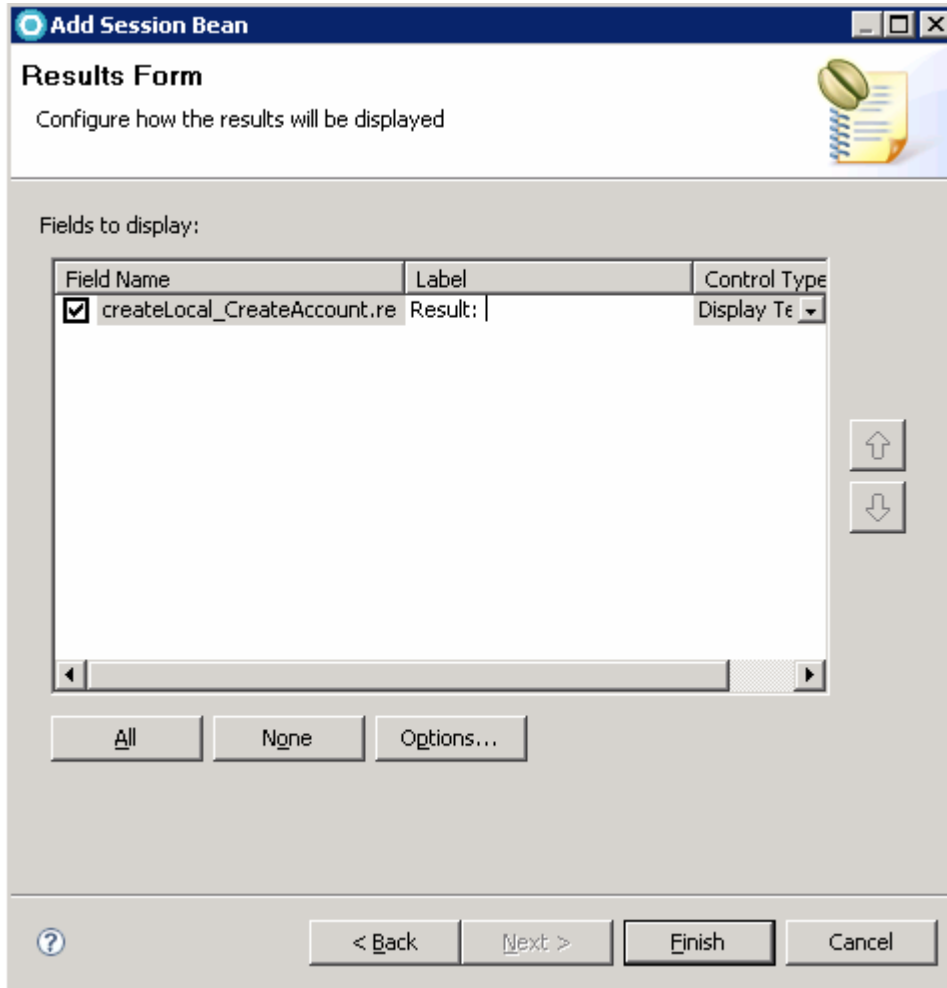
Configure the data controls for the input form

Fields to display:

Field Name	Label	Control Type
<input checked="" type="checkbox"/> customernumber (long)	Customer number:	Input Text
<input checked="" type="checkbox"/> accountnumber (int)	Account number:	Input Text
<input checked="" type="checkbox"/> balance (int)	Balance:	Input Text

8) Click **Next**.

9) Change the “Label” field to **Result:** as shown below.



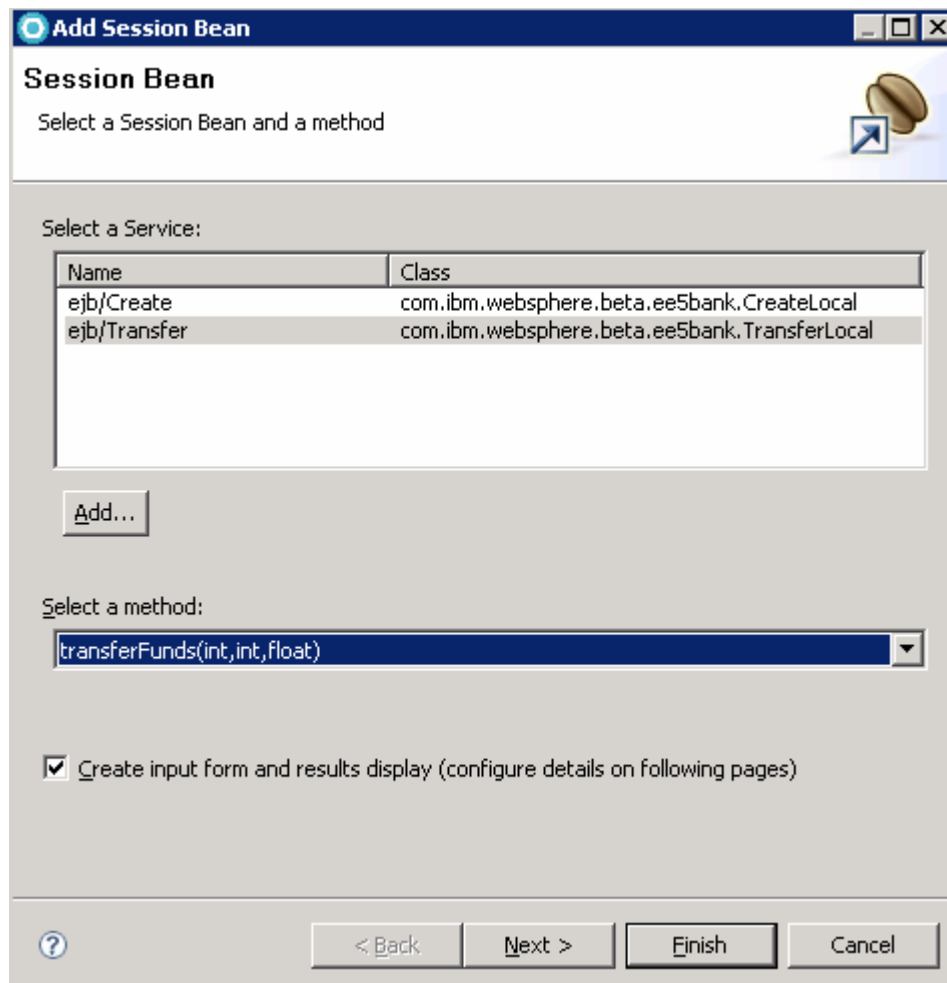
10) Click **Finish**. The form that you just defined in the wizard will be added to the Web page. Note that the field names are the same as the “Label” column from the wizard.

11) **Save** and **Close** newAccount.jsp.

___ i. Create a page called transferFunds.jsp. This page will call the Transfer EJB to transfer money between two accounts.

- 1) Click-and-drag the **New page** component from the palette into the editor, next to newAccount.jsp (you will see a black vertical line appear next to newAccount.jsp)
- 2) Type **transferFunds.jsp** in the component’s text field to name it.
- 3) Double-click **transferFunds.jsp** to edit it.
- 4) Ensure that the “File name” field reads **transferFunds.jsp**.
- 5) Ensure that **A-01_blue.html** is selected as the template.
- 6) Click **Finish**.

- 7) A Web page editor will open. Type “**Transfer funds between two accounts**”, and type **enter** three times to add carriage returns, where you see “place your page content here”.
- ___ j. Add a session bean element to the page. This element will use a JSF form and the “Transfer” bean to transfer money from one account to another.
 - 1) Click **Data and Services** in the palette.
 - 2) Drag the **EJB Session bean** element onto the page, below the text you entered.
 - 3) Click **Add...** in the “Add session bean” window.
 - 4) Expand **EE5Bank** and **EE5Bank.jar**, and select **Transfer**.
 - 5) Click **Finish**.
 - 6) Ensure that `transferFunds(int,int,float)` is selected in the “Select a method” menu as shown below, and click **Next**.



7) Edit the values in the "Label" column to give the fields acceptable display names, as shown below.

Add Session Bean

Input Form

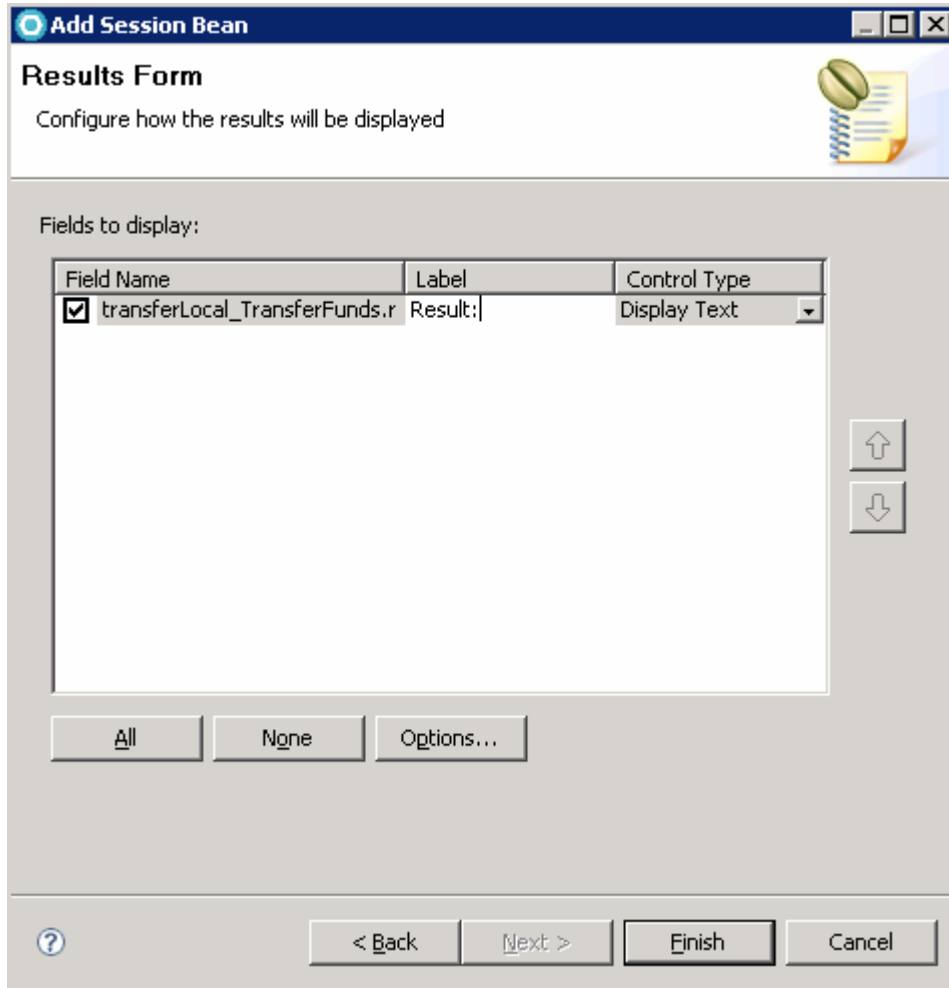
Configure the data controls for the input form

Fields to display:

Field Name	Label	Control Type
<input checked="" type="checkbox"/> acct1 (int)	From account:	Input Text
<input checked="" type="checkbox"/> acct2 (int)	To account:	Input Text
<input checked="" type="checkbox"/> amount (float)	Amount:	Input Text

8) Click **Next**.

9) Change the “Label” field to **Result:** as shown below.



10) Click **Finish**. The form that you just defined in the wizard will be added to the Web page. Note that the field names are the same as the “Label” column from the wizard.

11) **Save** and **Close** transferFunds.jsp.

___ 20. The navigation links in the template currently show the file names for each Web page you have created. Edit the navigation labels to give the pages more suitable display names.

___ a. In the “Web site navigation” tab, click **home.jsp**.

___ b. The “Properties” panel will be displayed below. Enter **Home** in the “Navigation label” field.

___ c. Click **newCustomer.jsp**.

___ d. The “Properties” panel will be displayed below. Enter **Add customer** in the “Navigation label” field.

___ e. Click **newAccount.jsp**.

___ f. The “Properties” panel will be displayed below. Enter **Add account** in the “Navigation label” field.

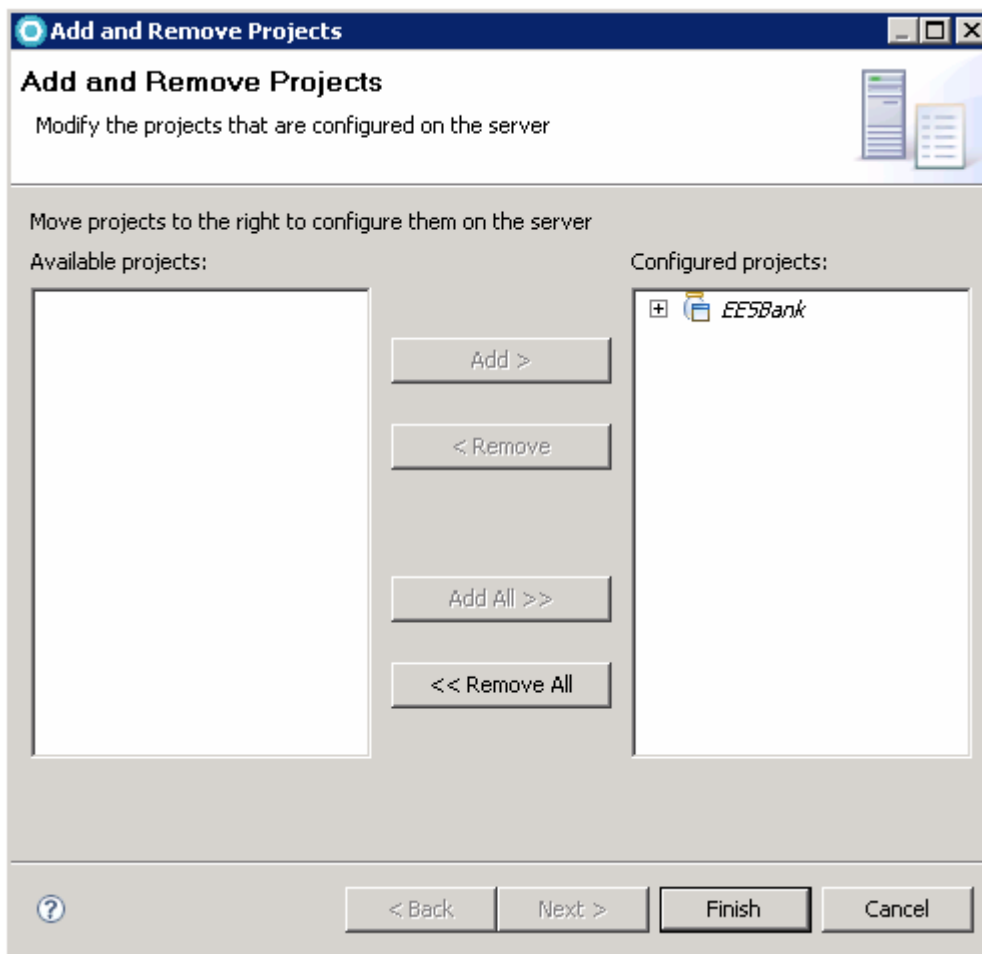
___ g. Click **transferFunds.jsp**.

- ___ h. The “Properties” panel will be displayed below. Enter **Transfer funds** in the “Navigation label” field.
- ___ i. Select **File > Save** to save the current Web site navigation.
- ___ j. Double-click **transferFunds.jsp** to view the new labels applied to the Web site template.

Part 6: Deploy the application to the server

In this section you will associate your enterprise application project with the local installation of WebSphere Application Server, and deploy the application for testing. This process will be familiar to you if you have used previous versions of Rational Application Developer or the Application Server Toolkit.

- ___ 21. Associate the EE5Bank application with the server you defined in Part 1.
 - ___ a. Click **Servers** in the lower panel to display the Servers tab.
 - ___ b. Right click on **WebSphere Application Server v7.0 at localhost** and select **Add and remove projects...**
 - ___ c. Select **EE5Bank** and click **Add >** to move it to the right-side panel, as shown below.



- ___ d. Click **Finish**.
- ___ e. Wait for the server's "Status" column to change from "Publishing" to "Synchronized". (Log data will be displayed in the Console tab between these two states)

Part 7: Test the application

In this section you will use the application that you have developed in the previous parts of this exercise. You will create a customer, and then create two accounts that belong to that customer. After the accounts have been created, you will transfer money from one account to the other.

- ___ 22. Launch the application in a Web browser.
- ___ a. The “Web site navigation” tab should still be displayed. Right-click **home.jsp** and select **Run on server**.
 - ___ b. In the window that is displayed, ensure that **WebSphere Application Server v7.0 at localhost** is highlighted, and click **Finish**.
 - ___ c. If security alerts are displayed by the browser, click **Yes** to proceed.
- ___ 23. Create a customer.
- ___ a. Click **Add customer** in the navigation area of the Web page.
 - ___ b. Enter the following values in the form:

Customer number:	1
First name:	John
Last name:	Doe
Tax ID:	5678
 - ___ c. Click **Submit**.
 - ___ d. Note that when the form has been submitted, the console panel will show JPA loading, and the page will refresh. The page will appear the same. The customer has been created, though there is no displayed status to indicate this.
- ___ 24. Create an account.
- ___ a. Click **Add account** in the navigation area of the Web page.
 - ___ b. Enter the following values in the form:

Customer number:	1
Account number:	1
Balance:	300
 - ___ c. Click **Submit**.
 - ___ d. Note that when the form has been submitted, the “Result” field will update, displaying “Account 1 created successfully”. This is because the session bean JSF component allows you to display a string as the result of an operation. This is one advantage of using a session bean façade to manipulate JPA Entities, rather than working with them directly from the Web container. Examine the Java source code for Transfer.java if you want to see how this is achieved.
- ___ 25. Create another account.

___ a. Enter the following values in the form:

Customer number: **1**
Account number: **2**
Balance: **100**

___ b. Note that the "Result" field now reads "Account 2 created successfully"

___ 26. Transfer \$50.00 from account 1 to account 2.

___ a. Click **Transfer funds** in the navigation area of the Web page.

___ b. Enter the following values in the form:

From account: **1**
To account: **2**
Amount: **50.00**

___ c. Click **Submit**.

___ d. Note that the "Result" field now shows that 50.00 dollars have been transferred, and displays the current balances of both accounts.

Part 8: Clean up

In this section you will stop the WebSphere Application Server process and then exit Rational Application Developer.

- ___ 27. Stop the running instance of WebSphere Application Server.
 - __ a. Click the **Servers** tab in the bottom panel.
 - __ b. Right-click the running server and select **Stop**.
 - __ c. Wait for the server's "State" field to change to "Stopped".
- ___ 28. Close Rational Application Developer for WebSphere Software.
 - __ a. Select **File > Exit**.

What you did in this exercise

In this exercise you developed, deployed, and tested a simple Java EE 5 application using Rational Application developer. In doing so you gained exposure to the new Java EE 5 development tools in Rational Application Developer, and updated versions of existing tools. You used the server tools to connect to WebSphere Application Server, the data tools to create and configure a Derby database, EJB tools to create session beans using a wizard, JPA tools to create Entities based on database tables, and Web tools to create JSF-based Web pages that communicated with the session beans and JPA Entities.

This page is left intentionally blank.