

IBM RATIONAL APPLICATION DEVELOPER 6.0 – LAB EXERCISE

## Building a JSF Application and Relational Record Lists

What this exercise is about .....	1
Lab Requirements .....	1
What you should be able to do .....	1
Introduction .....	2
Exercise Instructions .....	2
Part 1: Setup Development Environment.....	2
Part 2: Build the Inventory interface for the Supplier Application .....	4
Part 3: Test Supplier Application.....	14
What you did in this exercise .....	17
Solution Instructions .....	18

---

**NOTE:** Education materials and other documentation as applicable including programming manuals, operating guides, physical planning manuals and installation manuals related to the IBM Products may be early versions subject to change. Documents will be furnished solely for the purpose of and for the duration of the Beta Test.

---

---

### What this exercise is about

The objective of this lab is to provide you with an understanding of basic creation of a website using JavaServer Faces (JSF) and the JDBC Mediator (Relational Record and Relational Record List).

### Lab Requirements

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

- IBM Rational Application Developer v6.0 beta with embedded WebSphere Application Server v6.0 Test Environment installed with beta patches installed
- Lab source files (Labfiles60.zip) must be extracted to the local system (preferably C:\).

### What you should be able to do

At the end of this lab you should be able to use IBM Rational Application Developer v6.0 to:

- Create a Faces JSP File
- Define a new Relational Record and Relational Record list (SDO-based technology using the JDBC Mediator)

- Add and configure JSF components to a JSF page
- Define a JSF navigation rule
- Update a relational record using tooling generated code

---

## Introduction

IBM Rational Application Developer v6.0 allows for Rapid Application Development using Java Server Faces. Application developers, generally start with the UI, because they think of the application first in terms of how it looks. Behaviors such as accessing data are then added on top of the UI, and pages are linked together to form the application. IBM Rational Application Developer v6.0 provides an easy, visual way to develop rich Web applications that leverage the J2EE environment. The application that you will be working with is, Supplier, a back-end application for the PlantsByWebSphere sample. The Supplier application will simulate a 3<sup>rd</sup>-party supplier that processes back-orders of plants and seeds. For this lab you will create a UI for the Supplier application that allows for lists of customers, stored in a database, to be viewed and individual customers updated.

## Exercise Instructions

Because these instructions are not operating-system specific, the directory locations will be specified in the lab instructions using symbolic references, as follows:

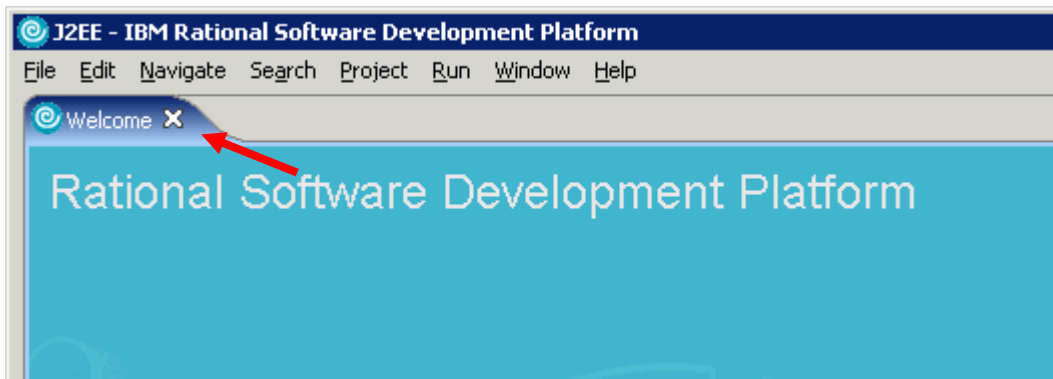
Reference Variable	Windows Location	AIX/UNIX Location
<LAB_FILES>	C:\Labfiles60	/tmp/Labfiles60

The following sections list the instructions for this lab.

## Part 1: Setup Development Environment

In this part of the lab you will run a script that will create the supplier database you will be accessing throughout this exercise. After setting up the database you will open IBM Rational Application Developer with a blank workspace.

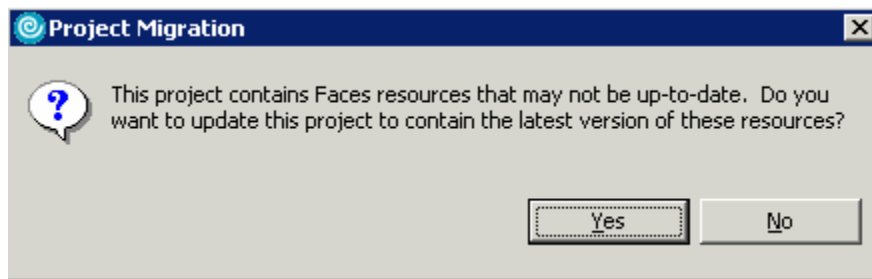
- \_\_\_ 1. Create the supplier database. This database will contain the customers which the JSF pages you create will list and allow for updating.
  - \_\_\_ a. Open a command window and navigate to:  
`<LAB_FILES>\IRAD_JDBCMediator\`
  - \_\_\_ b. Enter the following command:  
`setup\setupSupplierDB.bat`
- \_\_\_ 2. Start IBM Rational Application Developer v6.0.
  - \_\_\_ a. Select **Start > Programs > IBM Rational > IBM Rational Application Developer v6.0 > Rational Application Developer**.
  - \_\_\_ b. When prompted enter `<LAB_FILES>\IRAD_JDBCMediator\workspace` for your workspace.
  - \_\_\_ c. Click **OK**.
- \_\_\_ 3. When IBM Rational Application Developer v6.0 opens, close the welcome page.



- \_\_\_ 4. Import the base Supplier application into Rational Application Developer.
  - \_\_\_ a. Select **File > Import...**
  - \_\_\_ b. Select **EAR file** and click **Next**.
  - \_\_\_ c. Select **Browse...** and navigate to `<LAB_FILES>\IRAD_JDBCMediator\SupplierInitial.ear` and click **Open**.
  - \_\_\_ d. For the Project name, enter **SupplierEAR**.
- \_\_\_ 5. Click **Finish**.
- \_\_\_ 6. Switch to the Web Perspective.
  - \_\_\_ a. From the menu select **Window > Open Perspective > Other...**
  - \_\_\_ b. Select **Web** and click **OK**.

## Part 2: Build the Inventory interface for the Supplier Application

- \_\_\_ 1. Create a new Faces JSP page for the UI.
  - \_\_\_ a. From the Project Explorer view, expand **Dynamic Web Projects > Supplier**.
  - \_\_\_ b. Right click on Web Content and select **New > Faces JSP File** from the context menu.
  - \_\_\_ c. Enter **customers** for the file name, and make sure the checkbox next to the option **Create from page template** is selected. Click **Next**.
  - \_\_\_ d. On the Page Template File Selection page select the **User-defined page template option** and select **template.jptl** from the list of templates.
  - \_\_\_ e. Click **Finish**.
  - \_\_\_ f. Click **Yes** to the message box asking if the project should be updated with the appropriate Faces resources.



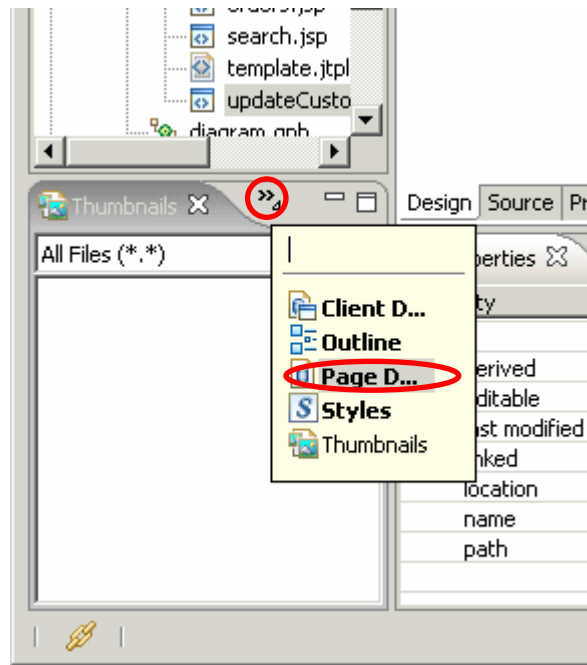
- \_\_\_ g. **Replace** the “**Default content of bodyarea**” with **Customers**.
- \_\_\_ h. From the menu select **Format > Paragraph > Heading 1**.
- \_\_\_ i. From the menu select **Format > Align > Horizontal Center**.

---

NOTE: You can also use items on the toolbar to format and align items in the Page Designer

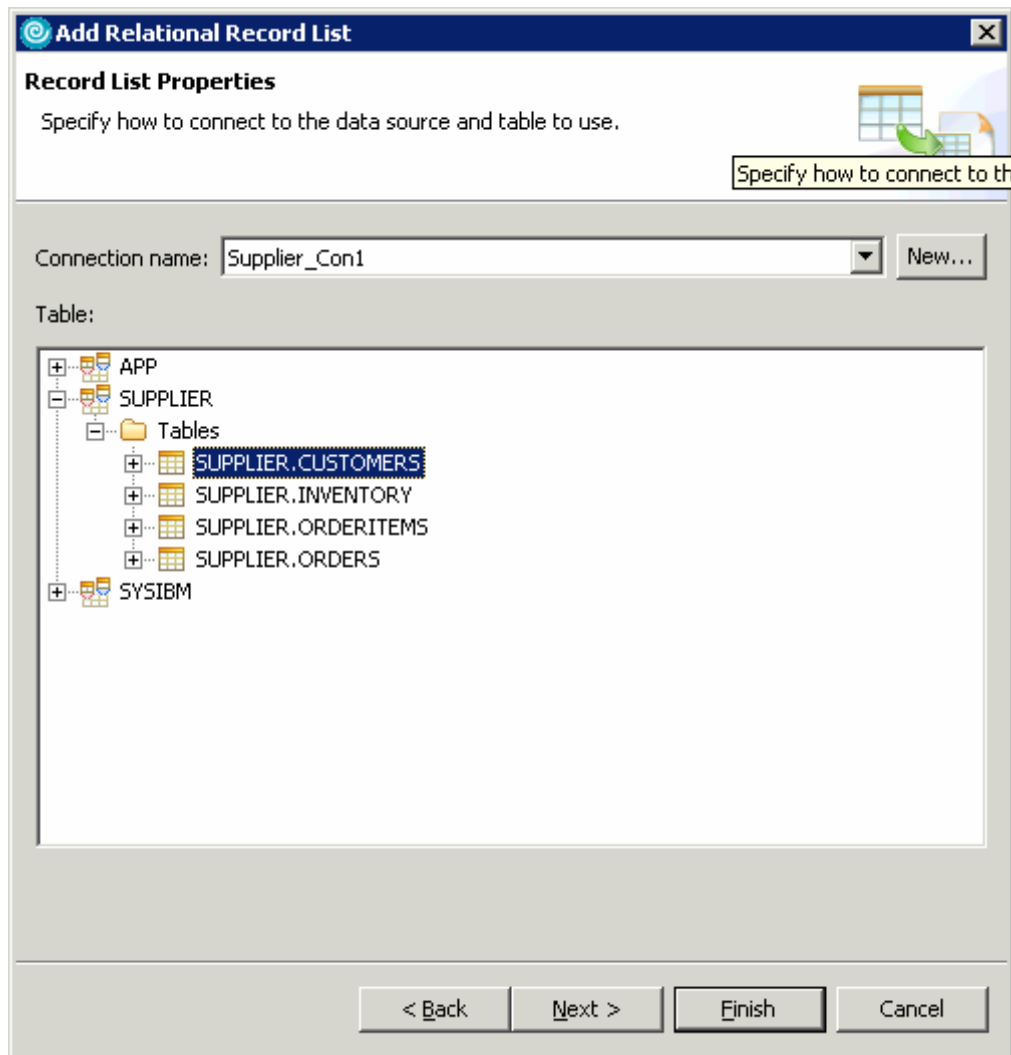
---

- \_\_\_ 2. The UI will display a list of customers stored in the SupplierDB database. The customers can be displayed on the page by using the a JSF component. Create a Relational Record from the Page Data view.
  - \_\_\_ a. Open the customers.jsp file if it is not still open in the editor area.
  - \_\_\_ b. The **Page Data** view by default, is located in the lower left frame in your workspace. If it is not visible, click on the >> icon and select **Page Data** from the list.

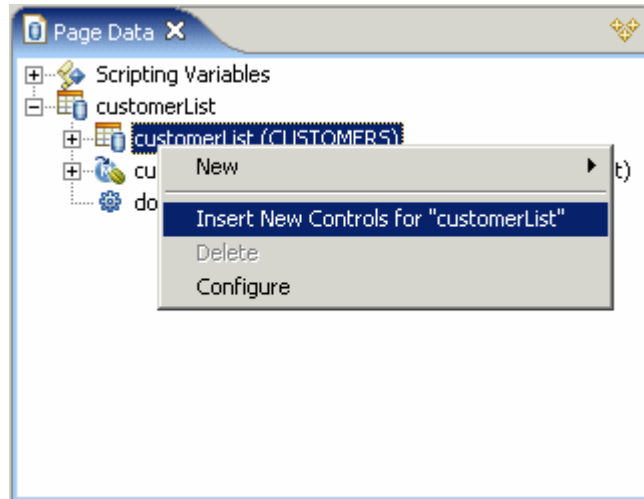


- \_\_\_ c. Right click in the Page Data view and select **New > Relational Record List**.
- \_\_\_ d. Enter **customerList** for the name of the Relational Record List and click **Next**.
- \_\_\_ e. On the Record List Properties page click the **New...** button next to the Connection name field.
- \_\_\_ f. On the Select A Database page, click the **Create New DB Connection** button.
- \_\_\_ g. Accept the default connection name and click **Next**.
- \_\_\_ h. Make sure **Cloudscape v5.1** is selected for the database manager. For the database location enter or navigate to <LAB\_FILES>IRAD\_JDBCMediator\databases\SUPPLIERDB. Click **Finish**.
- \_\_\_ i. Click **Finish** again on the Select a Database page.

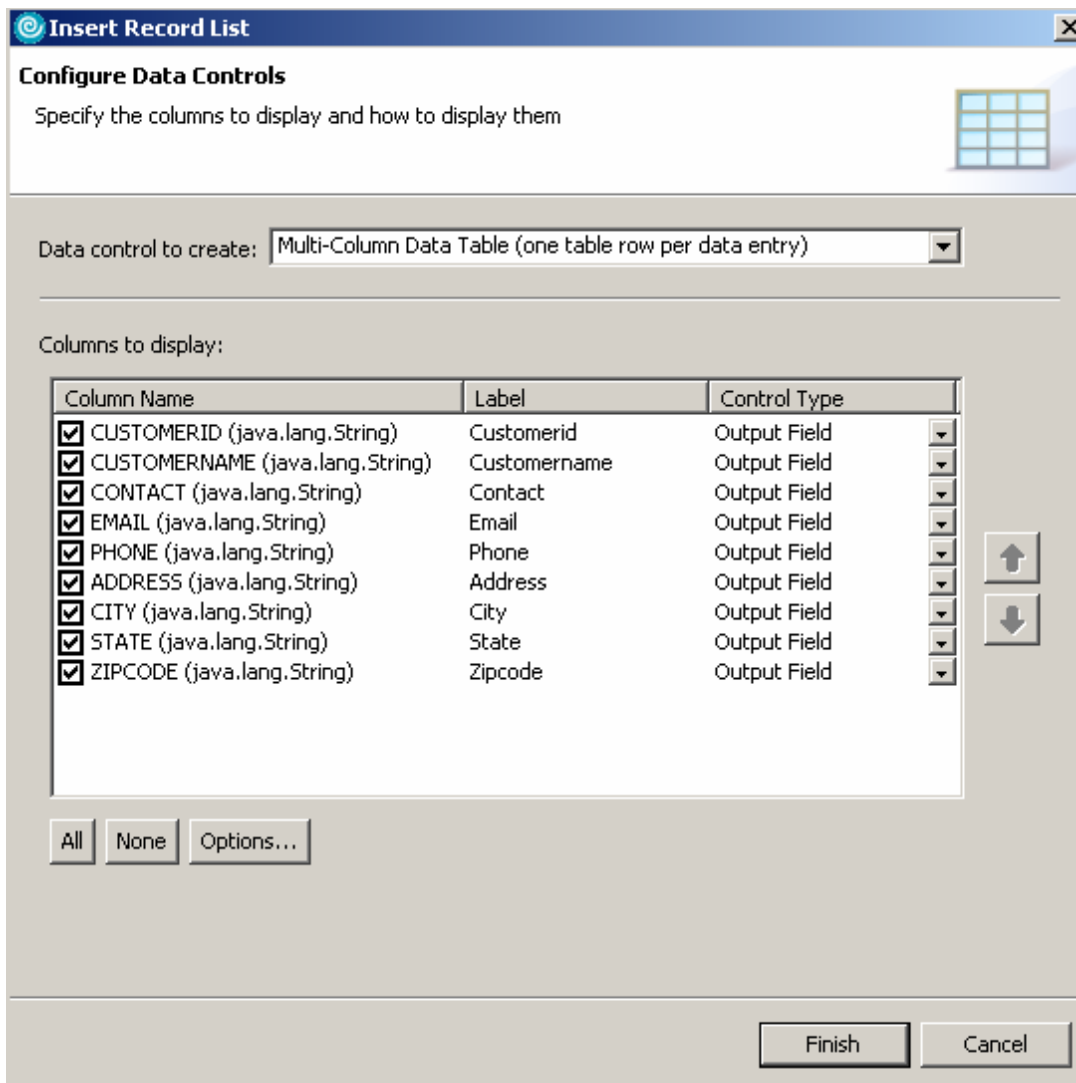
- \_\_\_ j. Select the **SUPPLIER.CUSTOMERS** table from the list of tables available from the SUPPLIERDB and click **Finish**.



3. After creating the customerList Relational Record List in the page data view you should see the following items in this view. Right click on **customerList (CUSTOMERS)** and select **Insert New Controls for "customerList"** from the context menu.




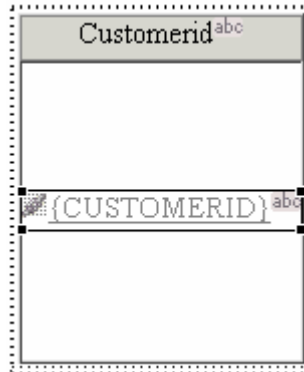
4. Insure **Multi-Column Data Table...** from the Data Control drop down list is selected. Click **Finish**.



5. Configure the data table that displays the inventory items.
- Select a column from the **customerList** table and go to the **Properties** view.
  - Select the **h:dataTable** tab at the left of the **Properties** view.
  - Enter **100** for the width and select % from the drop down list.
  - Check the box next to **Show header**.
  - Click **Display options** underneath **h:dataTable** in the **Properties** view.
  - Enter **5** for **Rows per page** and click the “**Add a web style pager**” button.
6. Add the Faces Link to the Customerid column. This will make a customer displayed in a row to be selected and linked to another page.



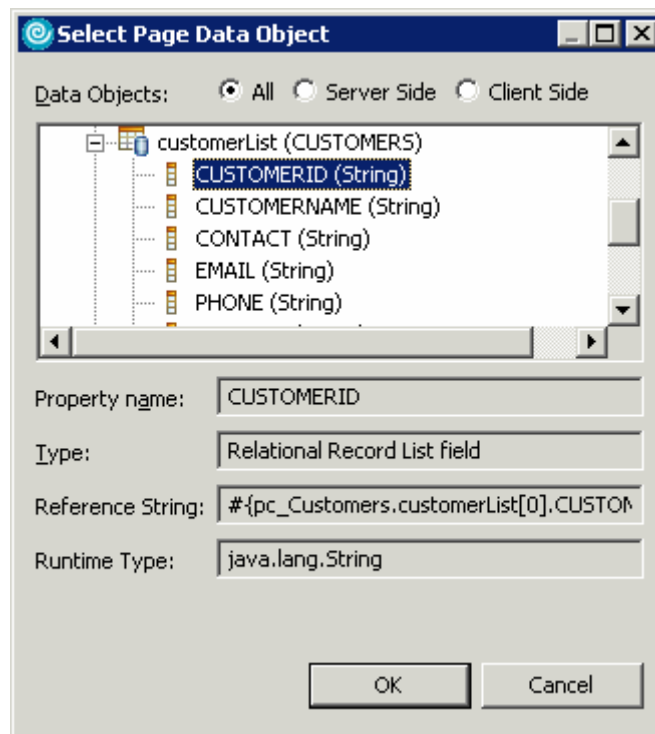
- \_\_\_ a. Drag and drop the **Faces Link** (  ) onto the very beginning of CUSTOMERID label. Your mouse point should be positioned at the very first letter. When dropped correctly, the Customerid column should look like the following:



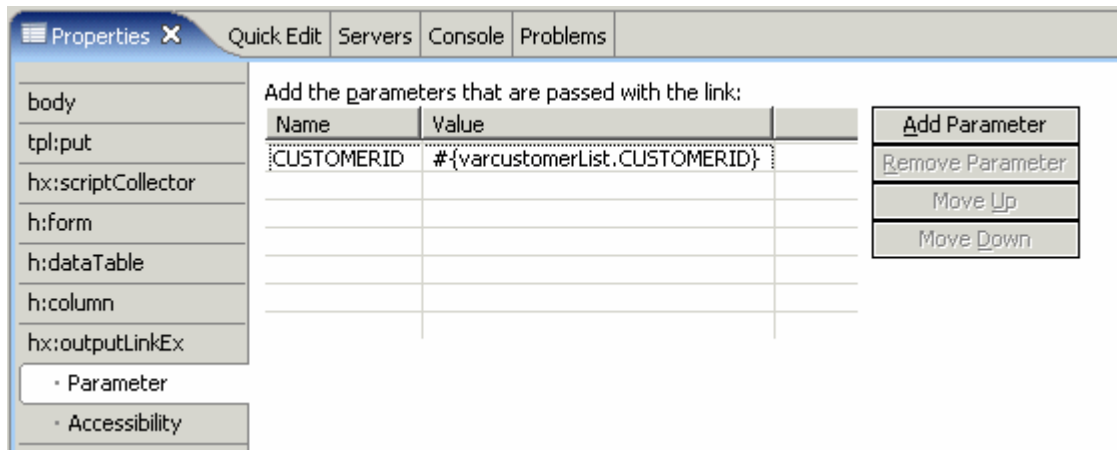
- \_\_\_ b. For the **URL** enter **/Supplier/faces/updateCustomer.jsp** and click **OK**.

NOTE: If you receive an error message indicating the Mapping JSF Tags to Java Methods, click Close. This is a known problem and will be fixed when the product is released.

- \_\_\_ c. Go to the **properties view > hx:outputLinkEx > Parameter** and click on the **Add Parameter** button.
- \_\_\_ d. Type **CUSTOMERID** for the Name.
- \_\_\_ e. To set the value field, click on the browse icon next to default value (  ). Select **CUSTOMERID** under customerList. Click **OK**.



\_\_\_ f. Verify the parameter value has been specified correctly.



\_\_\_ g. **Save** and **Close** the file.

---

NOTE: If you receive an error message which states a failure to Update JSP Index, click Close. This is a known problem and will be fixed when the problem is released.

---

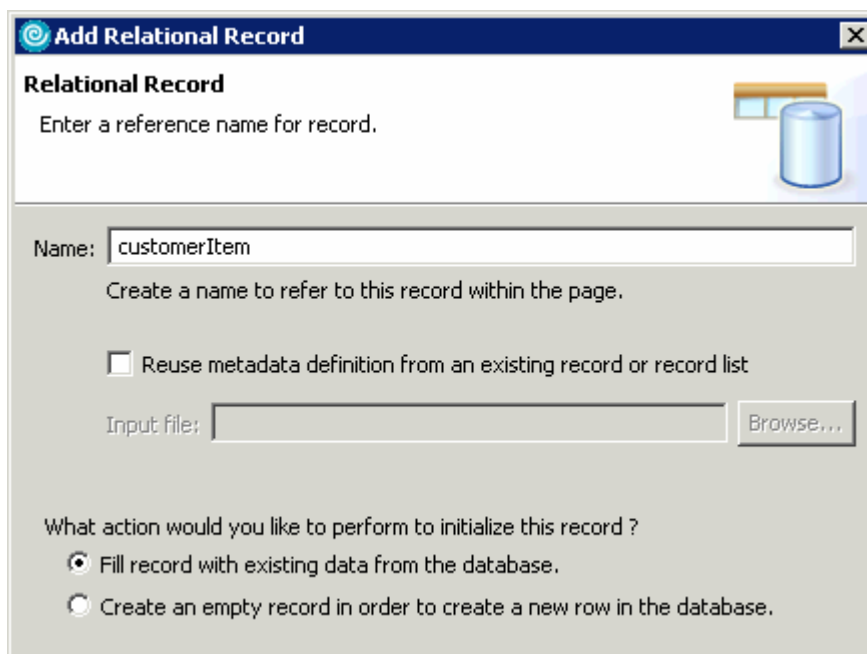
\_\_\_ 7. Create a new faces JSP file called **updateCustomer**.

\_\_\_ a. Follow the steps outlined in Step 1 above when you created the customer.jsp page.

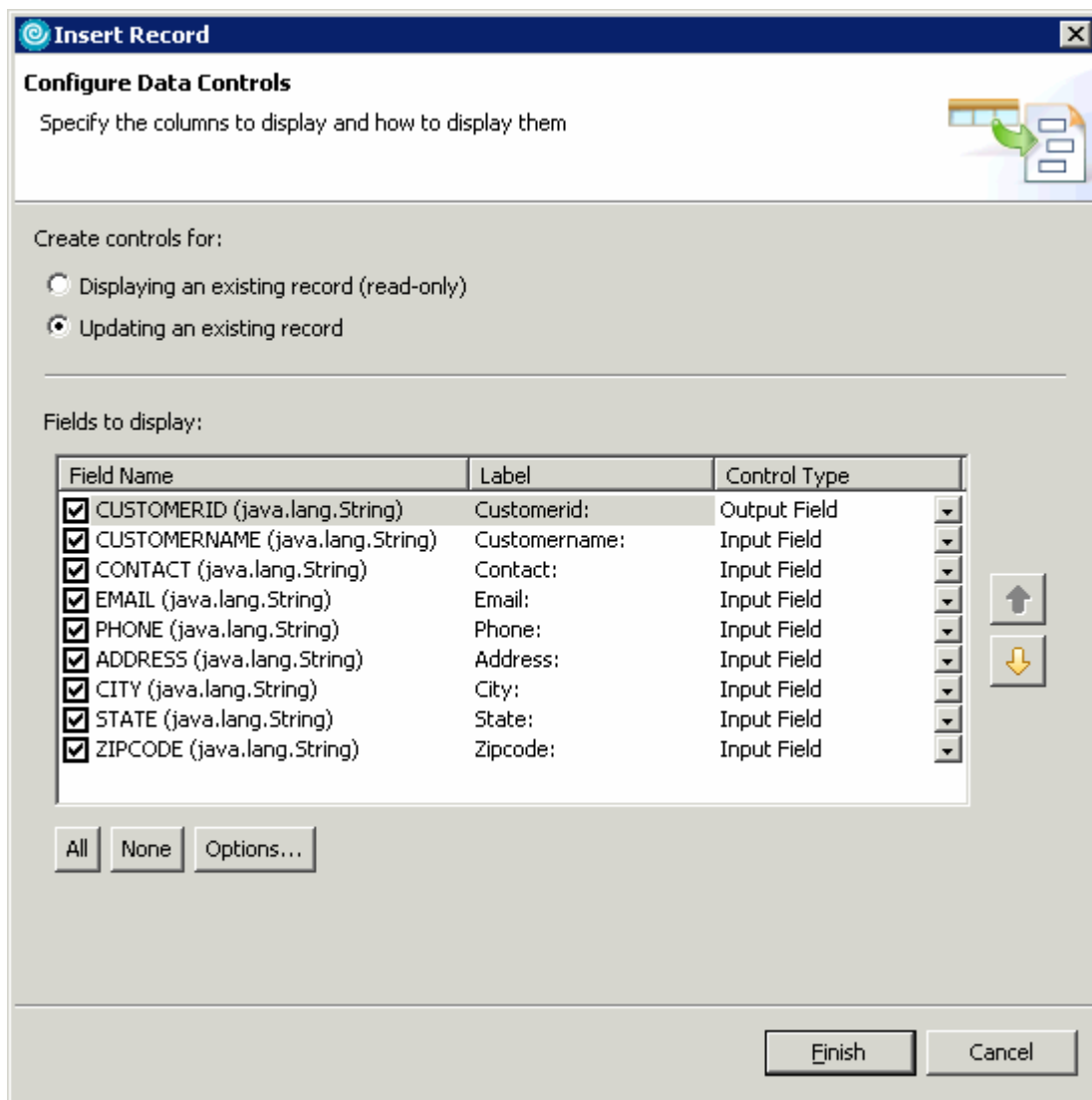
\_\_\_ 8. Add a relational record to the page.

\_\_\_ a. Right click in the Page Data view and select **New > Relational Record**.

\_\_\_ b. Enter **customerItem** as the relational record name. Insure the **Fill record...** radio button is selected. Click **Next**.

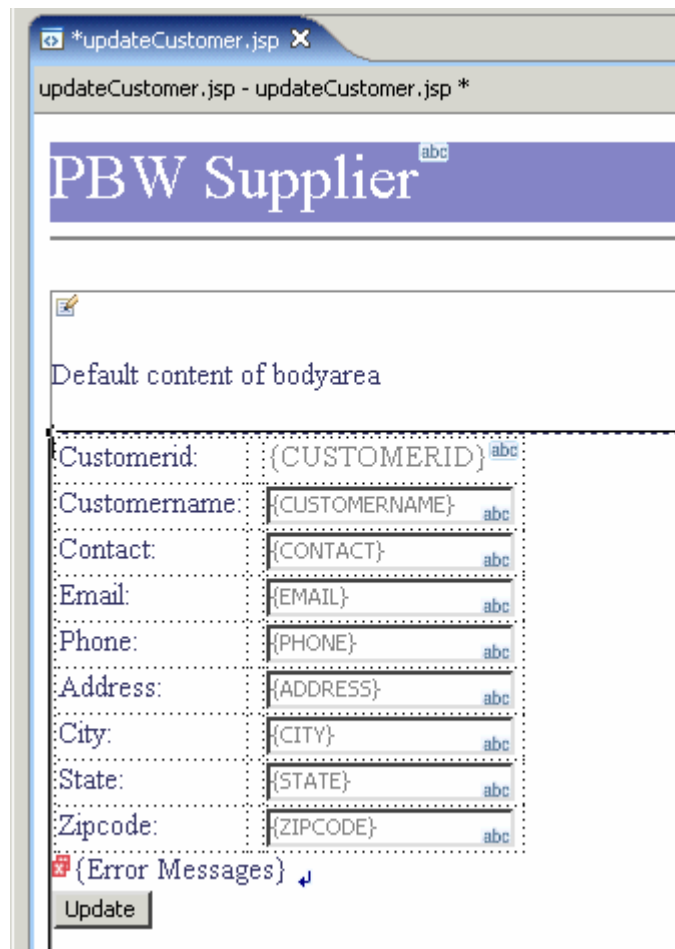


- \_\_\_ c. Select the **CUSTOMERS** table (SUPPLIER.CUSTOMERS). You will reuse the same connection you selected when you added the Relational Record list to the customers.jsp page in step 2.
  - \_\_\_ d. Click **Next**.
  - \_\_\_ e. Click on the **Filter Results** link and notice that the record that is going to be selected is based on the CUSTOMERID that is passed in as a parameter. Click **Close**.
  - \_\_\_ f. Click **Finish**.
- \_\_\_ 9. Once the record has been added to the page data view, you must add it to the Faces JSP page.
- \_\_\_ a. Drag and drop **customerItem (CUSTOMERS)** onto the center of the updateCustomer.jsp page.
  - \_\_\_ b. Select the radio button **Updating an existing record**. Change the Control type for the CUSTOMERID field to Output Field as shown below.



NOTE: Before dropping the relational record onto the page you may want to add one or more new lines after "Default content of bodyarea". To do this, place your cursor at the end of the text "Default content of bodyarea" and press the Enter key one or more times.

- \_\_\_ c. Click the **Options...** button and de-select the option to add a Delete button. Also, change the label of the Submit button to **Update**. Click **OK**.
- \_\_\_ d. Click **Finish**.
- \_\_\_ e. The Faces JSP page should display the interface for updating a customer.



- \_\_\_ 10. Once the controls have been added, select the **Update** button and go to the **Quick Edit** view.

- \_\_\_ a. In the quick edit view modify the Command code to return the String **“customers”**. After a customer is updated, the JSF framework will display the customer Faces JSP page. This is an explicit navigation directive. The Quick Edit view should look like this:

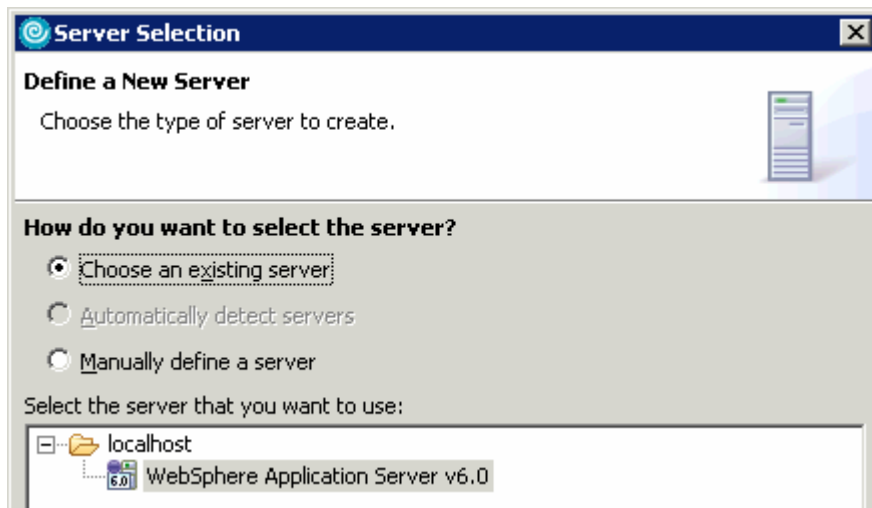
```
try {
    getCustomerItemMediator().applyChanges(
        getCustomerItem().getDataGraph().getRootOb
} catch (Throwable e) {
    logException(e);
}

return "customers";
```

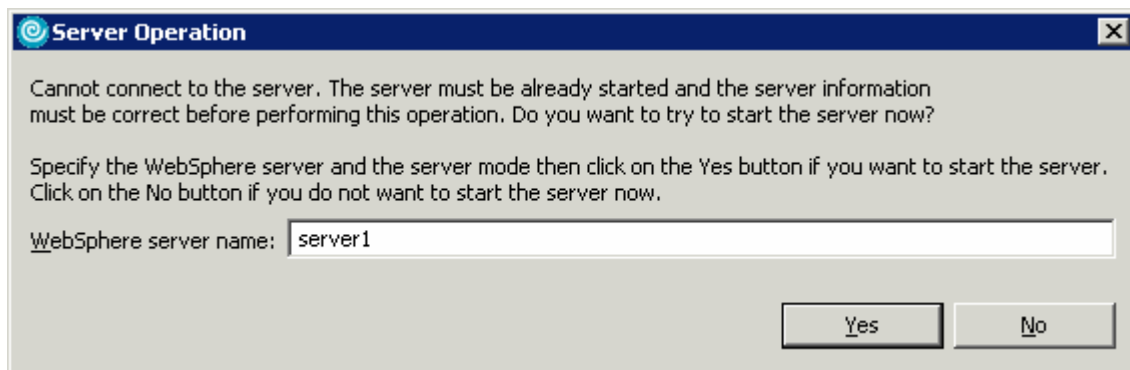
- \_\_\_ 11. **Save** and **Close** the file.

## Part 3: Test Supplier Application

- \_\_\_ 1. The application is complete and ready to test.
  - \_\_\_ a. In the Project Explorer view, expand **Dynamic Web Projects > Supplier > WebContent**. Right click on **customer.jsp** and click **Run > Run on Server....**
  - \_\_\_ b. Select the server you just created as the target server. Click **Finish**.



- \_\_\_ c. IBM Rational Application Developer will check to see if the server is started. If it isn't it will attempt to start the server. Click **Yes** to start the test server.



---

NOTE: There is currently 2 known issues to be aware of when using the Run on Server function with a JSF resource. First, there is a problem where the tooling opens the browser and requests the resource before the application has been published to the server. As a result, you will need to wait until the Supplier application has been published in order to enter the URL into the browser and make the request. Second, the URL that is passed to the browser is missing the faces path. Therefore, you will need to enter the following URL into the browser: <http://localhost:9080/Supplier/faces/customer.jsp>.

---

\_\_\_ 2. When the page opens in the browser, you should see something like the following:

PBW Supplier		Home	Customers	Orders	Search
Customerid	Customername	Contact	Email	Phone	Address
<a href="#">PBW001</a>	PlantsByWebSphere	Joe Smith	jsmith@pbw.com	555-555-7171	311 Oak Street
<a href="#">TGC001</a>	The Garden Center	Sue Jones	sjones@gardencenter.com	123-456-7890	123 Maple Street
<a href="#">SL001</a>	Superior Landscaping	John Mills	jmills@landscaping.com	098-765-4321	100 Rose Avenue

\_\_\_ 3. Click on the **PBW001** link.

\_\_\_ 4. The details for customer PBW001 will be displayed.

PBW Supplier		Home
Default content of bodyarea		
Customerid:	PBW001	
Customername:	<input type="text" value="PlantsByWebSphere"/>	
Contact:	<input type="text" value="Joe Smith"/>	
Email:	<input type="text" value="jsmith@pbw.com"/>	
Phone:	<input type="text" value="555-555-7171"/>	
Address:	<input type="text" value="311 Oak Street"/>	
City:	<input type="text" value="Garden City"/>	

\_\_\_ 5. Change the contact name to **Joan Smith** and click **Update**.

\_\_\_ 6. You should return to the customer list and note the update has been made.

Customerid	Customername	Contact	Email	Phone	Address
<a href="#">PBW001</a>	PlantsByWebSphere	Joan Smith	jsmith@pbw.com	555-555-7171	311 Oak Street
<a href="#">TGC001</a>	The Garden Center	Sue Jones	sjones@gardencenter.com	123-456-7890	123 Maple Street
<a href="#">SL001</a>	Superior Landscaping	John Mills	jmills@landscaping.com	098-765-4321	100 Rose Avenue



## What you did in this exercise

In this exercise, you created two new JSF pages and added these to an existing Dynamic Web Project using IBM Rational Application Developer v6. This exercise also highlighted using the Relational Record List and Relational Record page data items. By examining the page code file associated with each JSF page in this lab, the student should be more familiar with the SDO technology and JDBC Mediator APIs that are the basis of the Relational Record and Record List.

## Solution Instructions

If you are unable to get this lab to work, and would like to see the results, you can follow these steps to import the solution to this lab. You will then be able to browse the solution for examples of how the data components look and function in the final application.

- \_\_\_ 7. Start with a clean workspace and follow the steps to create the SupplierDB database in Part 1.
- \_\_\_ 8. Import the **JSF application** into Rational Application Developer for testing.
  - \_\_\_ a. Select **File > Import...**
  - \_\_\_ b. Select **Project Interchange** and select **Next**.
  - \_\_\_ c. Select **Browse...** and navigate to **<LAB\_FILES>\IRAD\_JDBCMediator\solution\suppliersolution.zip** and select **Open**.
  - \_\_\_ d. Click **Select All**.
  - \_\_\_ e. Click **Finish**.
- \_\_\_ 9. Now you can test the solution following the steps outlined in Part 3.

## Trademarks and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	iSeries	OS/400	Informix	WebSphere
IBM(logo)	pSeries	AIX	Cloudscape	MQSeries
e(logo)business	xSeries	CICS	DB2 Universal Database	DB2
Tivoli	zSeries	OS/390	IMS	Lotus

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. Microsoft, Windows, Windows NT, and

the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both. Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are

trademarks of Intel Corporation in the United States, other countries, or both. UNIX is a registered trademark of The Open Group in the United States and other countries. Linux is a registered trademark of Linus Torvalds. Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

This page intentionally left blank