© Copyright IBM Corporation 2005 All rights reserved

IBM RATIONAL APPLICATION DEVELOPER 6.0 – LAB EXERCISE

Building and Testing a Struts Portlet Application

What this exercise is about

In this lab, you will build a portlet in Rational Application Developer v6.0 using the Struts Portlet Framework. After you have finished designing and implementing your portlet, you will test it using the WebSphere Portal v5.0 Test Environment in Rational Application Developer.

Lab Requirements

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

- IBM Rational Application Developer v6.0 with the Portal Tools Additional Feature selected
- WebSphere Portal Server v5.0.2.2 Test Environment
- The lab source files LabFiles60.zip must be extracted to the root directory C:\

What you should be able to do

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

- Build a portlet application using the Struts Portlet Framework included by the WebSphere Portal Server
- Test a portlet application in the Portal Test Environment

© Copyright IBM Corporation 2005. All rights reserved

Introduction

In this lab, you will design a portlet that displays basic contact information; you can think of it as an address book. The main page of the View Mode is simply a list of contacts. When the user clicks on the name of a specific contact, the portlet displays detailed information about that contact. The Edit mode works mostly the same way, but clicking on a specific contact allows the user to modify that contact's information.

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
<WAS_HOME>	C:\WebSphere\AppServer	/usr/WebSphere/AppServer /opt/WebSphere60/AppServer
<IRAD_HOME>	C:\Program Files\IBM\Rational\SDP\6.0	
<LAB_FILES>	C:\Labfiles60	/tmp/Labfiles60
<TEMP>	C:\temp	/tmp
<LAB_NAME>	IRAD_Portal_Struts	

Windows users please note: When directory locations are passed as parameters to a Java program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, C:\LabFiles60\ would be replaced by C:/LabFiles60/

Solution instructions are included at the end in case you are unable to complete the lab.

© Copyright IBM Corporation 2005. All rights reserved

Part 1: Creating a New Struts Portlet Project

____ 1. Start Rational Application Developer.

__ a. Select **Start > Programs > IBM Rational > IBM Rational Application Developer V6.0 > Rational Application Developer**.

__ b. A dialog box will be displayed allowing you to select the location where you would like the workspace directory to be stored. Enter **<LAB_FILES>\<LAB_NAME>\workspace** for the location and select **OK**. Application Developer will start with an empty workspace. An empty workspace will leave your existing workspace untouched and help avoid name conflicts between what you may already have in your workspace and what you will be creating in this lab.

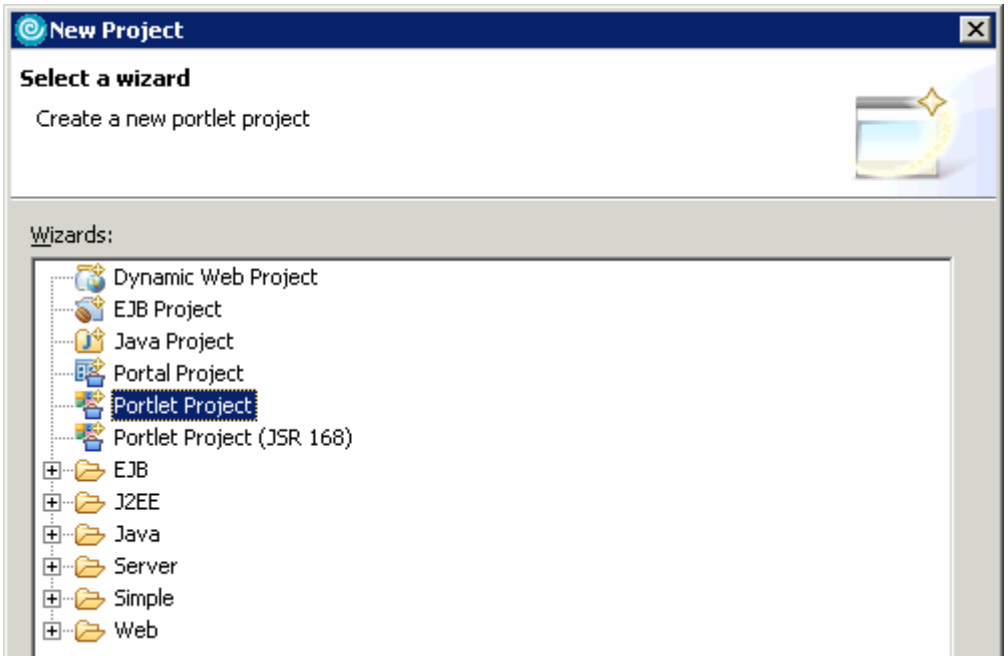
____ 2. When Rational Application Developer v6.0 opens, close the welcome page.

____ 3. Create a new project that will contain a portlet that utilizes the Struts-Portlet Framework.

__ a. Click **File > New > Project…**. The New Project wizard opens.

__ b. In the Select panel, select **Portlet Project** and click **Next >**.

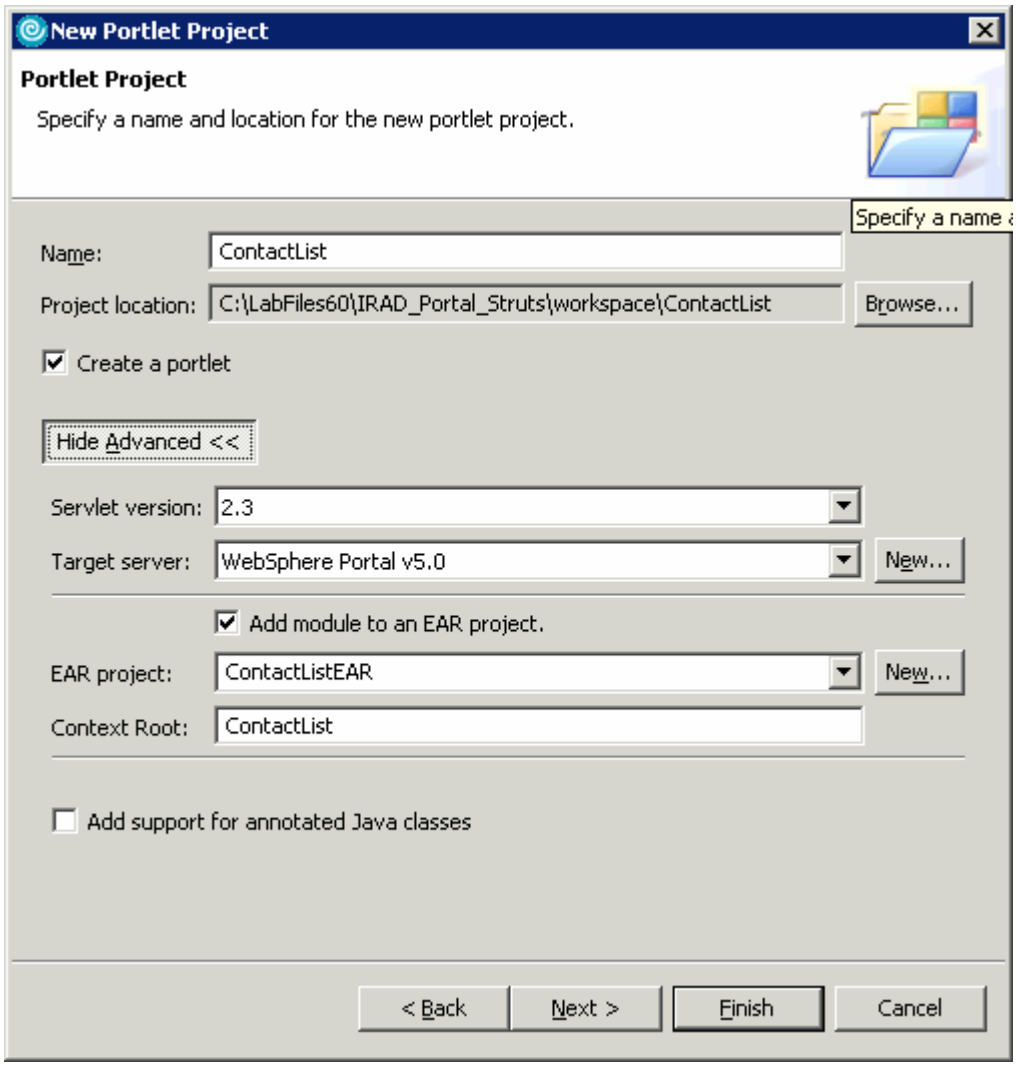
NOTE: The Struts-Portlet Framework only supports creating portlets using the IBM Portlet API.



__ c. In the Confirm Enablement window, read the message and click **OK**. Creating a Portlet project requires that advanced Web Development capabilities be enabled within Application Developer.

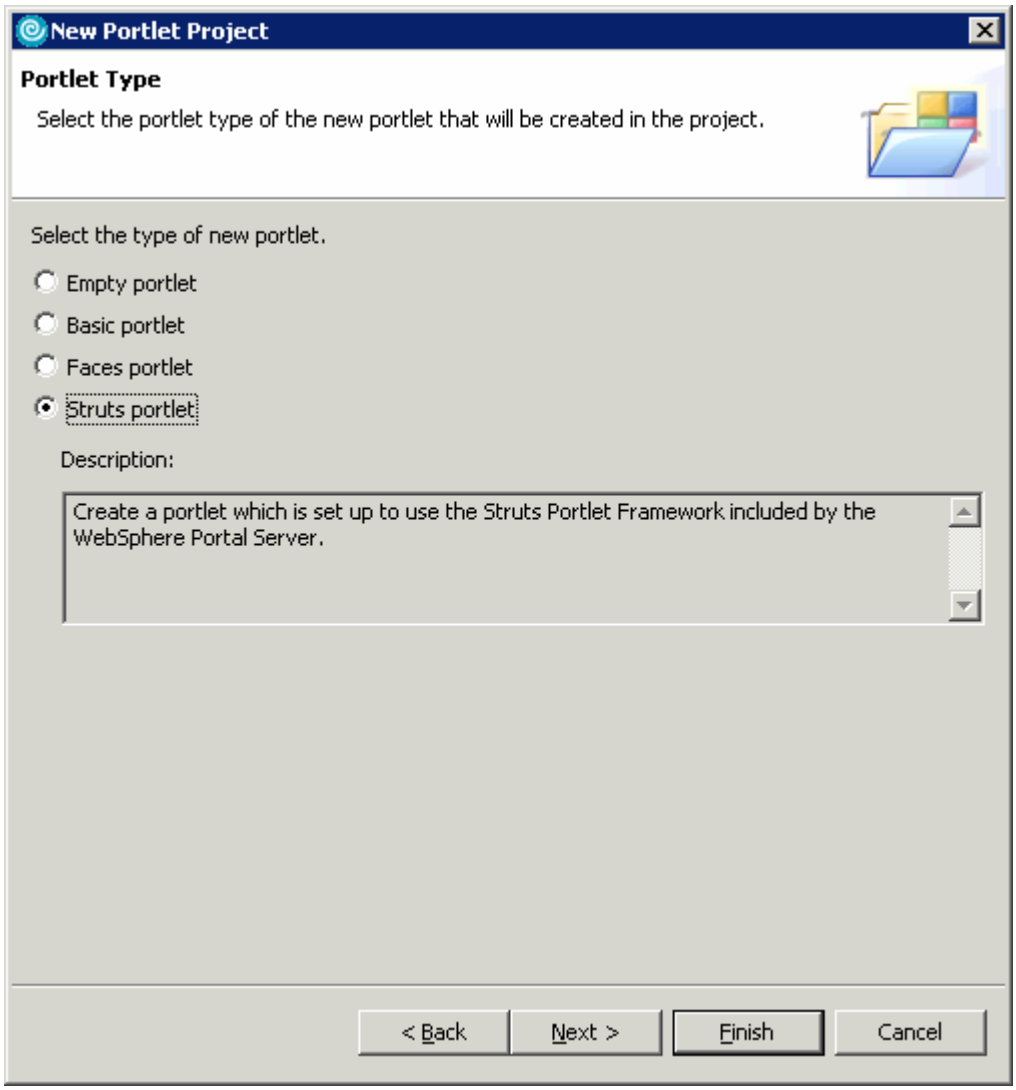
© Copyright IBM Corporation 2005. All rights reserved

__ d. In the Portlet Project panel, enter **ContactList** in the Name field. Click **Show Advanced >>** to show the advanced properties of the portlet project wizard. Verify that the Target server field is set to **WebSphere Portal v5.0**. Click **Next >**.



© Copyright IBM Corporation 2005. All rights reserved

__ e. In the Portlet Type panel, select **Struts portlet** (see the screen capture below) and click **Next >**.



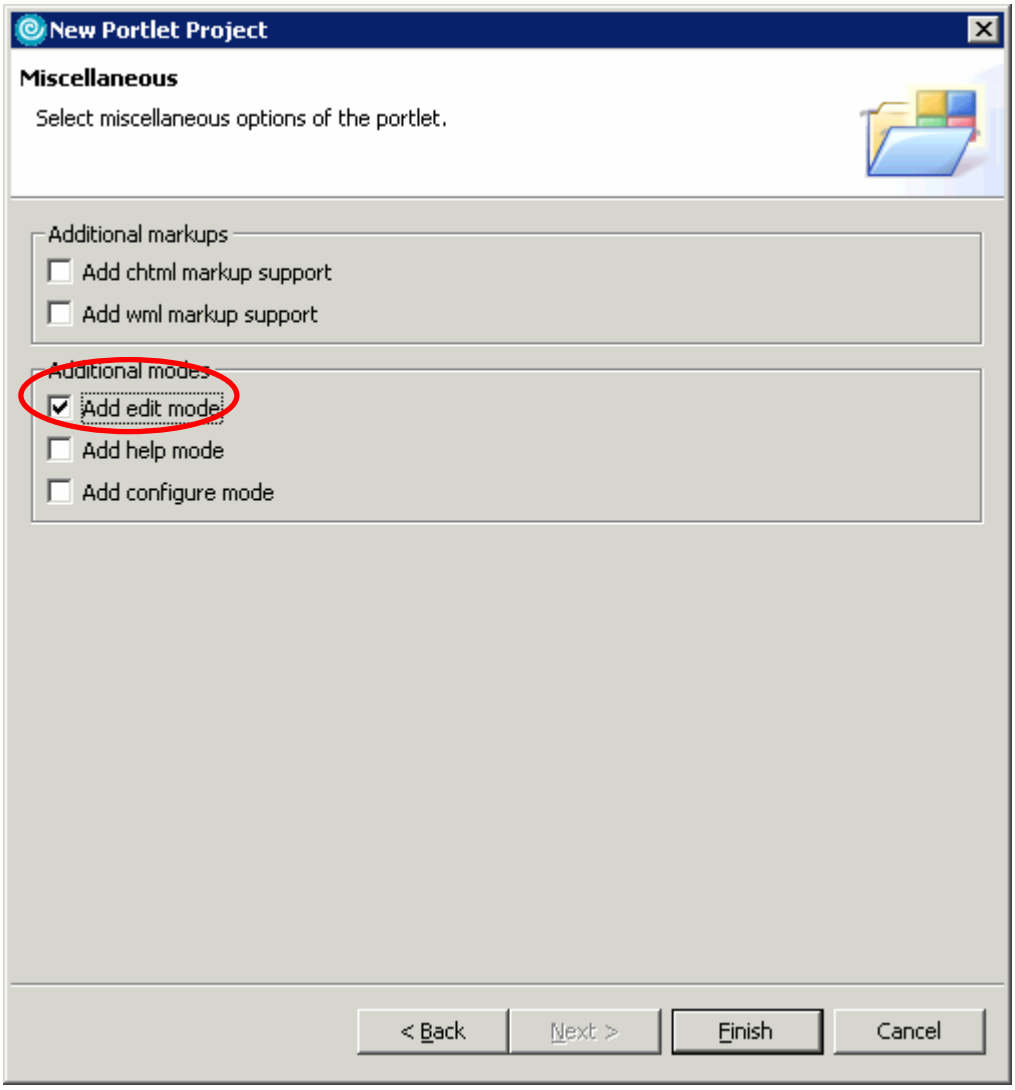
__ f. In the Features Page panel, observe the defaults and click **Next >**.

__ g. In the Portlet Settings page, observe the defaults and click **Next >**.

__ h. In the Struts Portlet Settings page, observe the defaults and click **Next >**.

© Copyright IBM Corporation 2005. All rights reserved

__ i. In the Miscellaneous page, select the box next to **Add edit mode** (circled in the screen capture below) and click **Finish**.



__ j. In the Confirm Perspective Switch window, read the message and click **Yes**. Portlet projects are associated with the Web Perspective within Application Developer.

____ 4. Observe the default settings of the new Portlet project.

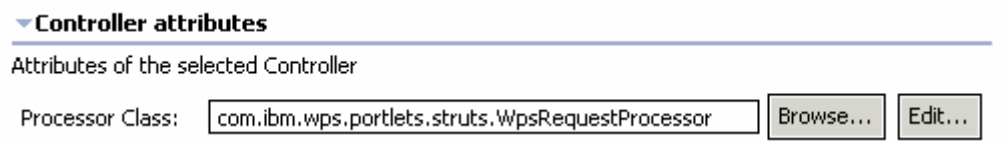
__ a. In the Project Explorer view, expand **Dynamic Web Projects > ContactList > WebContent > WEB-INF**. Note that two Struts configuration files, **struts-config.xml** and **struts-html-edit.xml** have already been created.

The struts-config.xml is the Struts configuration file for the View mode of the new portlet, and the struts-html-edit.xml is the Struts configuration file for the Edit mode of the new portlet.

__ b. Double click on **struts-config.xml** to open it in the Struts Configuration File editor.

© Copyright IBM Corporation 2005. All rights reserved

__ c. Select the **Controller** tab of the editor. Note that the Controller Processor class is com.ibm.wps.portlets.struts.WpsRequestProcessor.

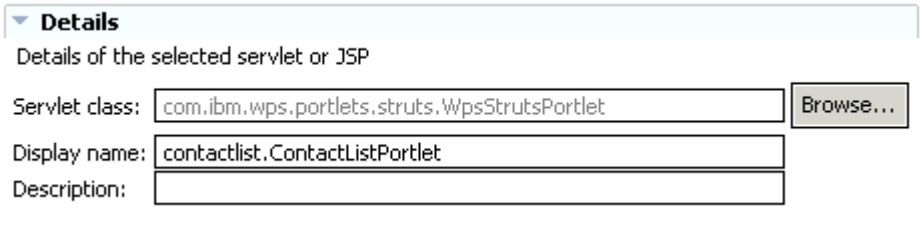


__ d. **Close** the Struts Configuration File editor.

__ e. In the Project Explorer view, double click on **web.xml** to open it in the Web Deployment Descriptor editor.

__ f. Select the **Servlets** tab of the editor.

__ g. Under the **Servlets and JSPs** section, select **contactlist.ContactListPortlet**. Under the Details section, note that the Servlet class is com.ibm.wps.portlets.struts.WpsStrutsPortlet.



__ h. **Close** the Web Deployment Descriptor editor.

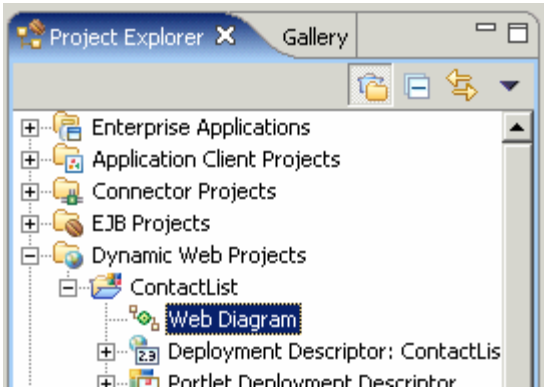
© Copyright IBM Corporation 2005. All rights reserved

Part 2: Define Web Diagrams for the Contact List Portlet

When you design a portlet using the Struts-Portlet Framework, each mode in the portlet can have its own Struts Application Diagram file. This makes it easy to modularize the execution flow of the individual modes in your portlet. In this lab, you will design a portlet that displays basic contact information; you can think of it as an address book. This portlet will have a View Mode and an Edit Mode.

____ 1. Design the Web diagram for the View Mode of your portlet. A web diagram was created automatically for you as part of the Struts portlet project.

__ a. Open the view mode's Web diagram. In the Project Explorer view, expand **Dynamic Web Projects** and double click on **Web Diagram**.

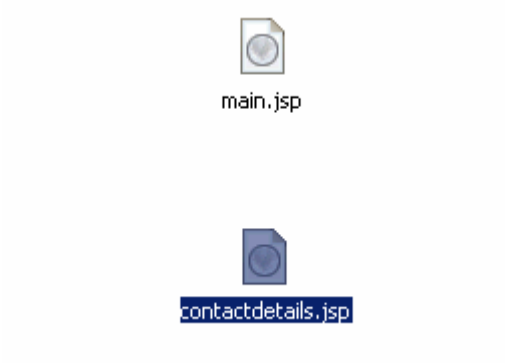


__ b. Add a JSP file named main.jsp" to the diagram. The main.jsp page will display a list of contact names. Click the **Web Page button** in the Palette view and then click anywhere on the Web diagram editor (see the screen capture below for an illustration). Type "**main.jsp**" and press Enter.

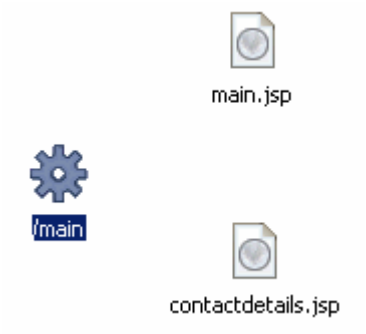
NOTE: If you do not specify an extension when you type the name, the ".jsp" extension will be added to the name automatically. You can rename a file later by right-clicking on a JSP file and then selecting Rename.

© Copyright IBM Corporation 2005. All rights reserved

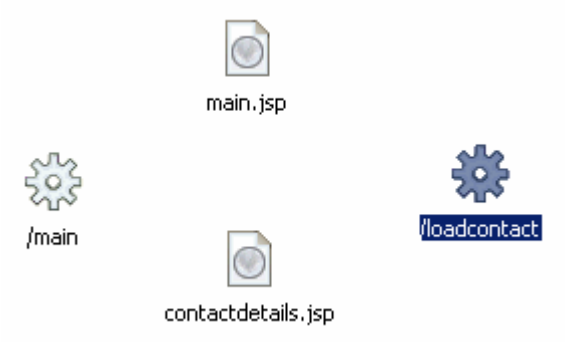
__ c. Add a JSP file named “contactdetails.jsp to the diagram. The contactdetails.jsp page will display more detailed information about an individual contact. Add a Web page named “**contactdetails.jsp**” below the main.jsp page on the diagram (see the screen capture below).



__ d. Add the “main” action mapping to the diagram. The “main” action will prepare the list of contacts required by main.jsp. Click the **Action Mapping button** in the Palette view and then click a spot on the diagram to the left of the main.jsp and contactdetails.jsp pages (see the screen capture below). Type “**main**” and press Enter.

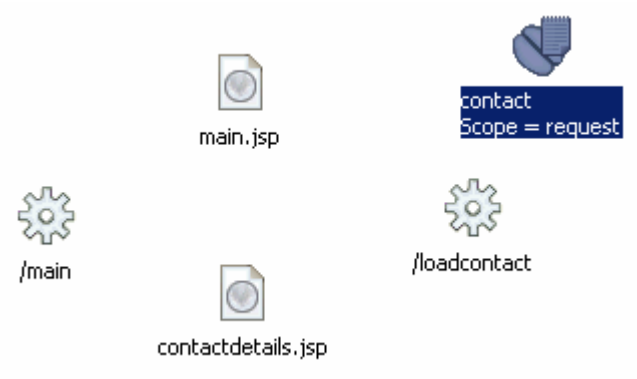


__ e. Add the “loadcontact” action mapping to the diagram. The “loadcontact” action will prepare the data needed by contactdetails.jsp. Add an action mapping named “**loadcontact**” to the right of the main.jsp and contactdetails.jsp pages on the diagram (see the screen capture below).

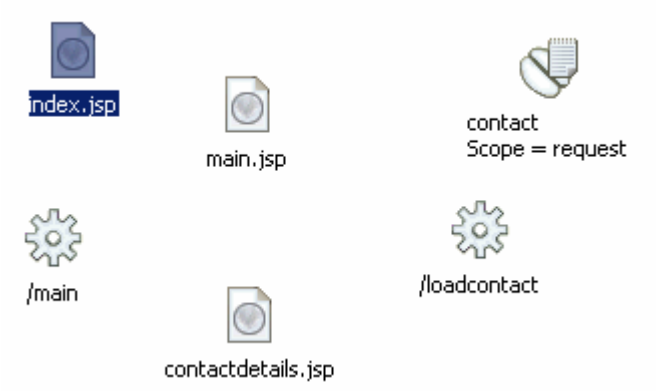


© Copyright IBM Corporation 2005. All rights reserved

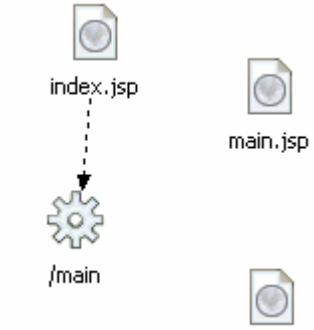
__ f. Add the “contact” Form Bean to the diagram. The data for an individual contact will be stored in a single “contact” Form Bean. Click the **Form Bean button** in the Palette view and then click a spot on the diagram above the loadcontact action mapping (see the screen capture below). In the Form Bean Attributes window, for the Form Bean Name field fill in **contact** and click **OK**.



__ g. The main.jsp page requires a contact list object that is initialized by the main action mapping. Since an action mapping cannot be specified as the default Welcome Page, a redirect page will need to be used. Add a Web page named “**index.jsp**” to the left of the main.jsp page on the diagram (see the screen capture below).

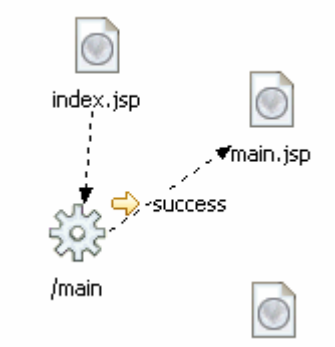


__ h. Add navigation flow to the diagram by connecting nodes in the graph. The index.jsp page is the first page that’s loaded when the View Mode opens. It forwards execution to the “main” action mapping, which initializes the contact list object utilized by main.jsp. To connect nodes in the diagram, click the **Connection button** in the Palette view. In the diagram, click “**index.jsp**” first and then click the “/**main**”. See the screen capture below to verify your actions.

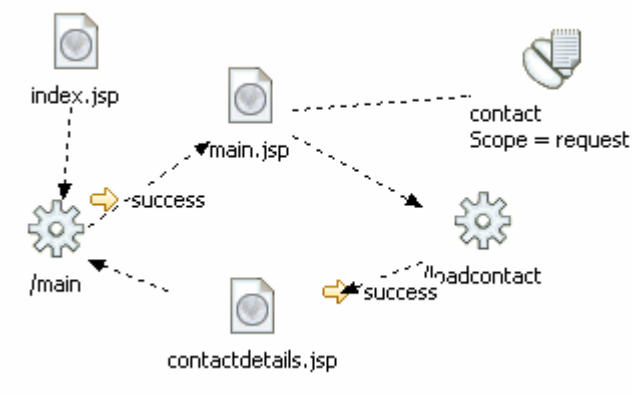


© Copyright IBM Corporation 2005. All rights reserved

__ i. Add a connection between “/main” and “main.jsp”. Click the **Connection button**. In the diagram, click “**/main**” and then click “**main.jsp**”. In the Choose a Connection window, expand **Local Forward** and double-click **<new>**. Type “**success**” and press Enter. See the screen capture below to verify your actions.



__ j. Connect the remaining nodes in the diagram based on the screen capture below.



__ k. Save the diagram. Type **Ctrl-S**.

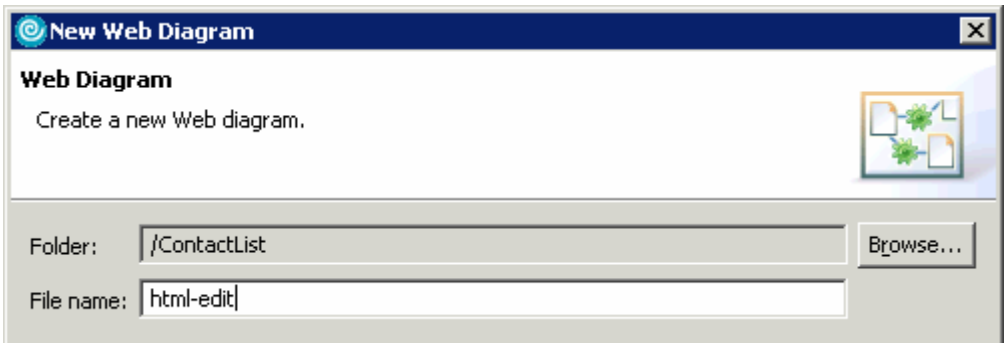
__ l. Close the Web diagram.

____ 2. Create a Web diagram for the Edit mode of the Contact List portlet.

__ a. In the Project Explorer view, expand **Dynamic Web Projects**. Right-click on **ContactList** and select **New > Other**. The New project wizard opens up.

__ b. On the Select a wizard panel, expand **Web** and select **Web Diagram**. Click **Next >**.

__ c. On the Web Diagram panel, enter **html-edit** for the File Name field. Click **Finish**.

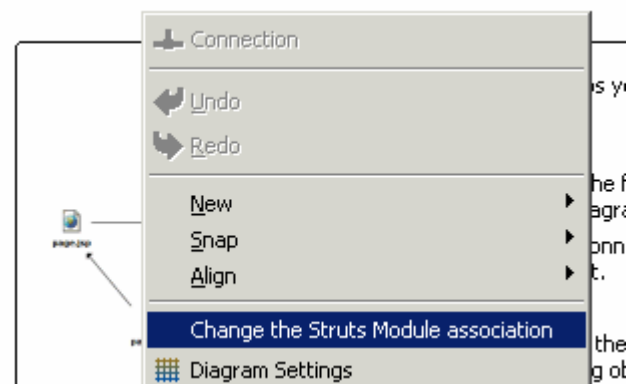


© Copyright IBM Corporation 2005. All rights reserved

__ d. The html-edit Web diagram is created and opened.

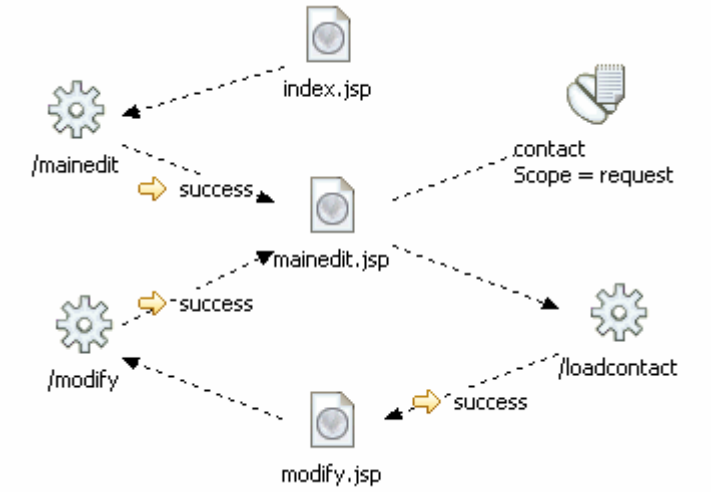
____ 3. Design the Web Diagram for the Edit Mode of the Contact List portlet.

__ a. Modify the Web diagram's Struts module association. Right-click anywhere in the Web diagram and select **Change the Struts Module association**.



__ b. In the Change Module Association window, select /**html/edit** for the Struts Module Name field. Click **OK**.

__ c. Construct the html-edit Web diagram using the screen capture below. An explanation of the how the Edit mode will work follows in the next step.



__ d. The Welcome page is index.jsp; it calls the "mainedit" action through a global forward. The "mainedit" action populates a list object used by the mainedit.jsp page to display a list of contacts. When the user clicks on the name of a specific contact in the list, the "loadcontact" action is called, which populates a form in the modify.jsp page where the user can modify a contact's information. When the user completes the form in modify.jsp, the "modify" action updates a contact's information and refreshes the list object to be displayed in the mainedit.jsp page. Note that the same Form-Bean will be used in the Edit Mode as was used in the View Mode.

__ e. Save the diagram. Type **Ctrl-S**.

__ f. Close the diagram.

© Copyright IBM Corporation 2005. All rights reserved

Part 3: Import Partially Created Struts Portlet Application

In this part, you will import parts of the Contact List application that have already been created. Portions of the application have been created to save development time.

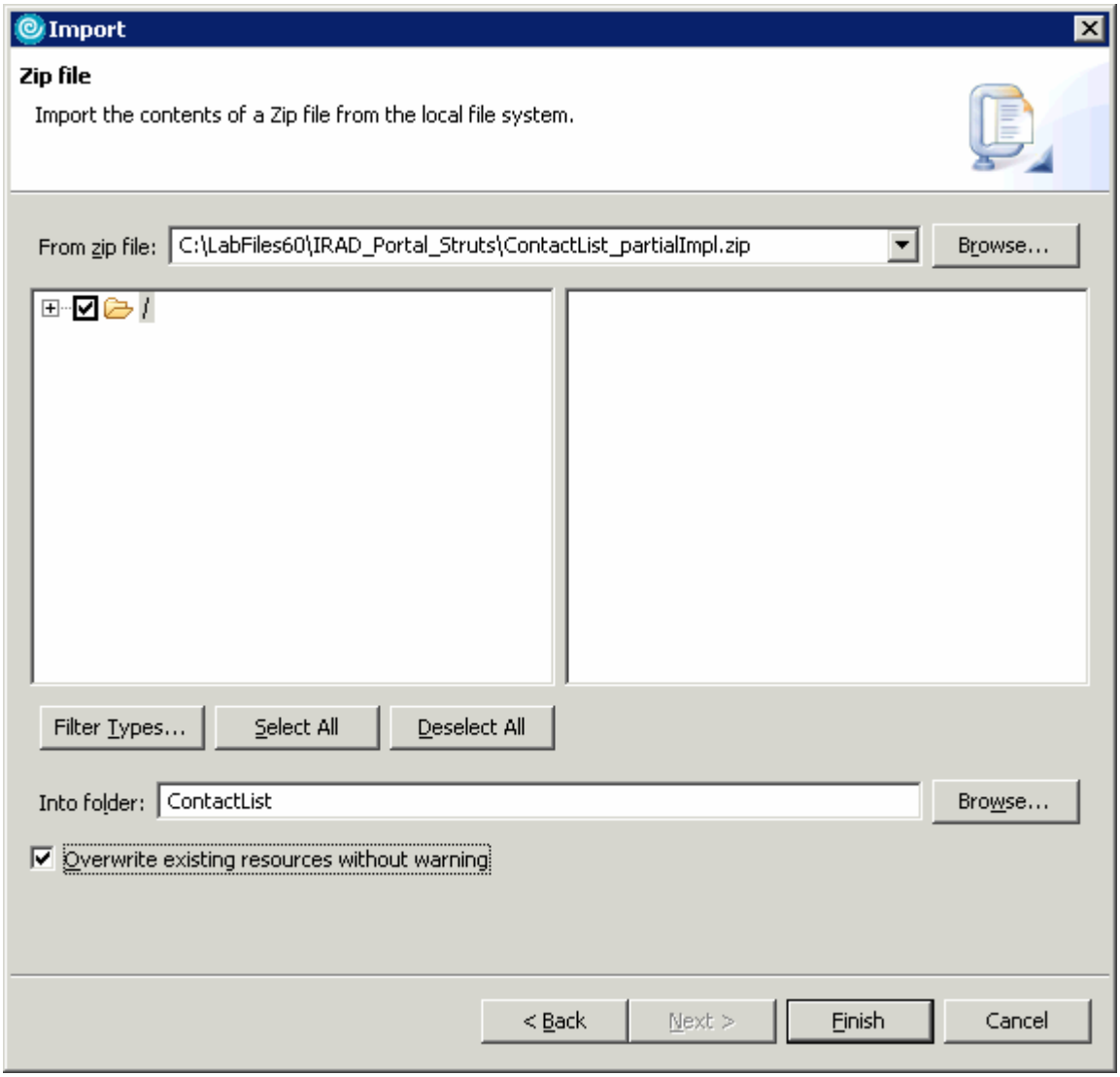
____ 1. Import the partially created Struts portlet application.

__ a. In the Project Explorer view, expand Dynamic Web Projects. Right-click on **Contact List** and select **Import… > Import…**. The Import wizard opens up.

__ b. In the Select panel, select **Zip File** and click **Next >**.

__ c. In the Zip file panel, fill in the following values (see the screen capture below for more details) : the From zip file with **<LAB_FILES>\<LAB_NAME>\ContactList_partialImpl.zip** and the Into folder with **ContactList.**

__ d. Place a check mark in the box next to Overwrite existing resources without warning.



© Copyright IBM Corporation 2005. All rights reserved

__ e. Click **Finish**.

__ f. Some warnings will appear in the Problems view. This is because the portlet has only been partially implemented, and some of the existing resources point to other resources that haven't been implemented yet. Ignore these warnings, because you will fix them later.

© Copyright IBM Corporation 2005. All rights reserved

Part 4: Implement the Contact List Portlet Application

Most of the process involved with implementing a Struts Portlet is identical to the process of implementing a Struts application. Since this lab assumes that you already have experience with the Struts Framework, it will not describe any Struts-specific implementation details. Instead, this section will focus on new implementation options available to Struts-Portlet applications. To save time, you will import a large part of the implementation for your portlet.

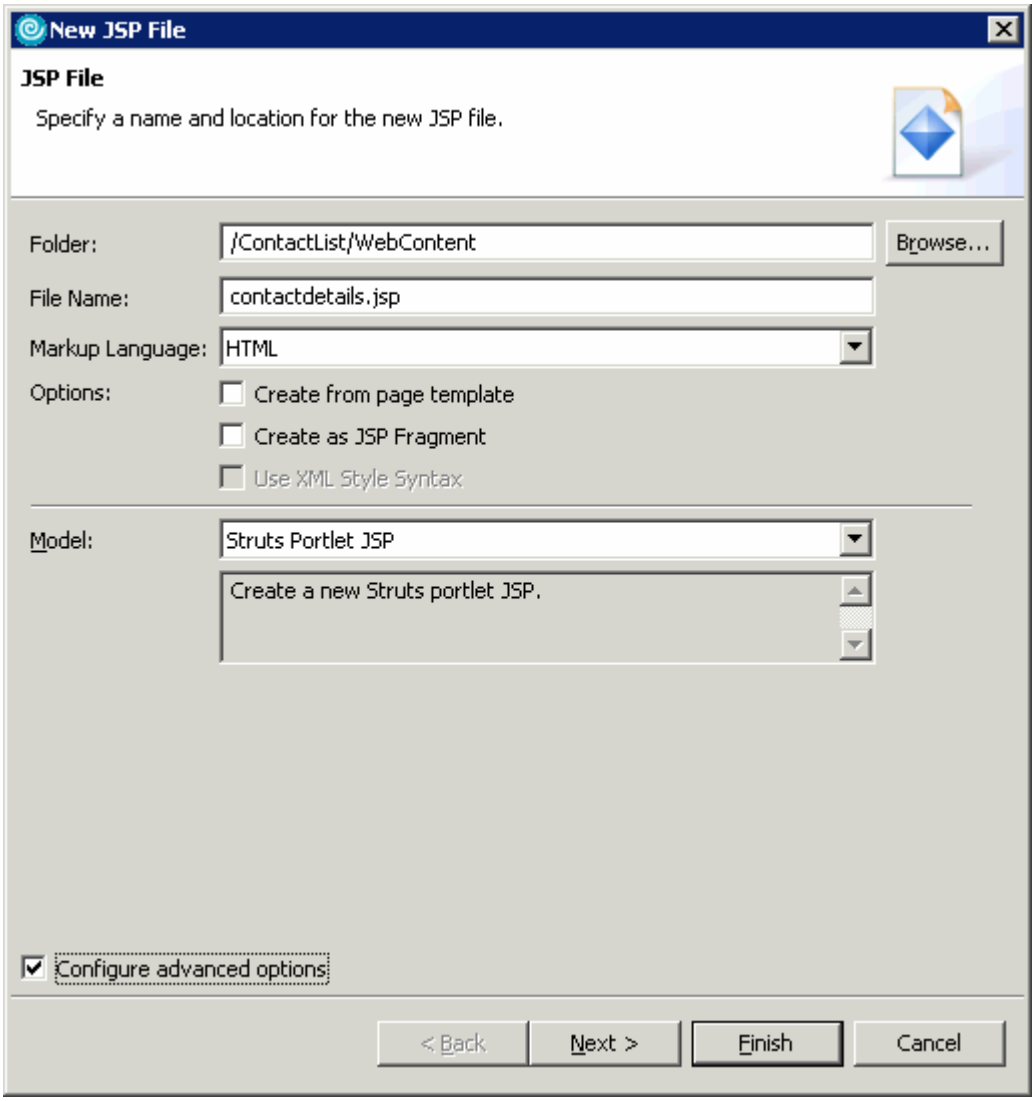
____ 1. Create the contactdetails.jsp page. This page displays detailed information about a selected contact.

__ a. Click **File > New > JSP File**. The New JSP File wizard will open up.

© Copyright IBM Corporation 2005. All rights reserved

__ b. In the JSP File panel, use the table below to fill in the fields and click **Next >** (see the screen capture below for more details).

Field Name	Value
Folder:	/ContactList/WebContent
File Name:	contactdetails.jsp
Model:	Struts Portlet JSP
Configure advanced options	Check this box



__ c. Click **Next >** three times

© Copyright IBM Corporation 2005. All rights reserved

__ d. In the Form Field Selection panel, uncheck the box next to '**Generate fields in a form**' (see the screen capture below) and click **Finish**.

__ e. The contactdetails.jsp JSP page is created and opened in the JSP editor.

____ 2. Explore the contactdetails.jsp JSP page.

__ a. In the contactdetails.jsp JSP editor, click the **Source tab**.

__ b. Note that the appropriate portlet and struts tag libraries are included in the page

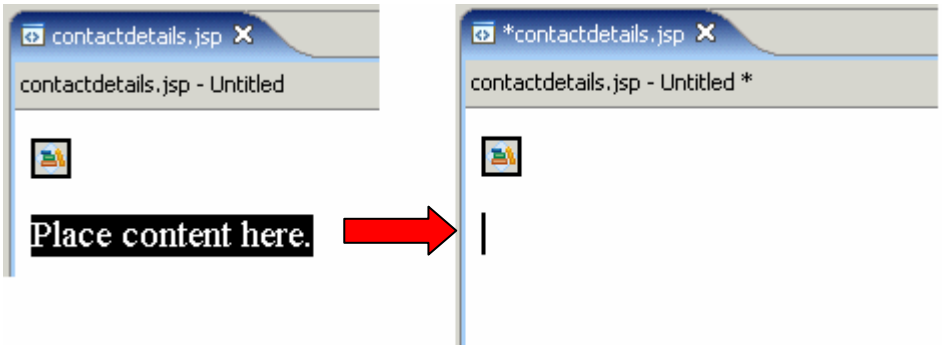
```
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI"%>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
```

__ c. Also note that the `<portletAPI:init />` tag is inserted at the beginning of the JSP file by default. This tag initializes the portlet characteristics of this JSP file, and it needs to precede any other portlet-specific tags on the page, it provides portlet variables that the JSP can use to access the corresponding objects of the portlet API.

____ 3. Add a table to the page to display the contents of the contact bean.

__ a. In the contactdetails.jsp JSP editor, click the **Design tab**.

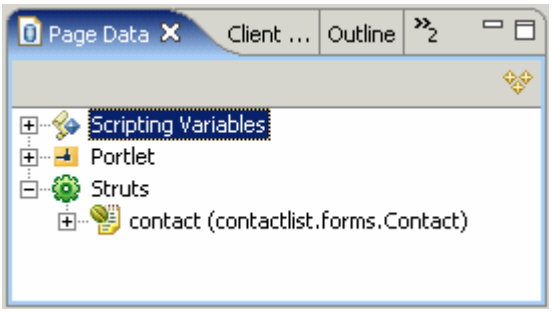
__ b. Highlight the "Place content here." sentence and press Delete.



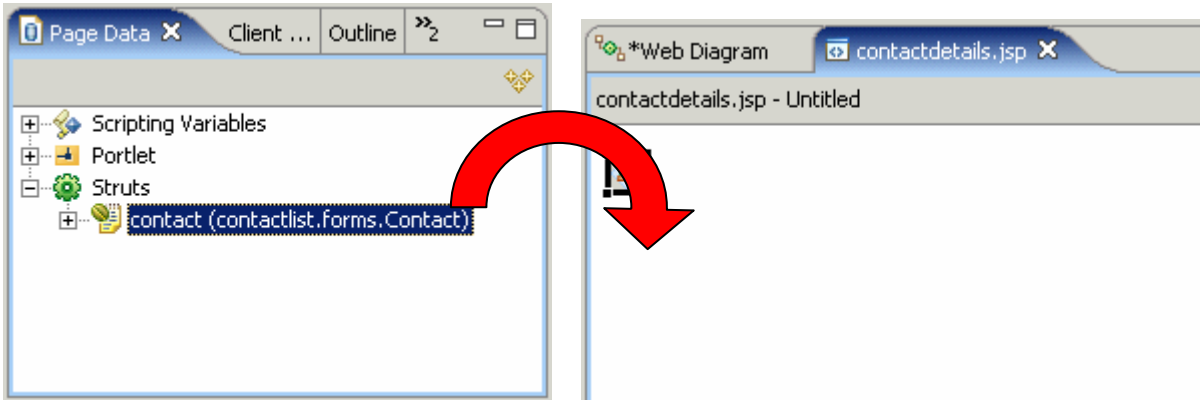
__ c. Switch to the Page Data view by selecting **Window > Show View > Page Data**.

© Copyright IBM Corporation 2005. All rights reserved

__ d. In the Page Data view, expand **Struts** to expose a list of available form beans. The **contact** form bean should be the only one available.



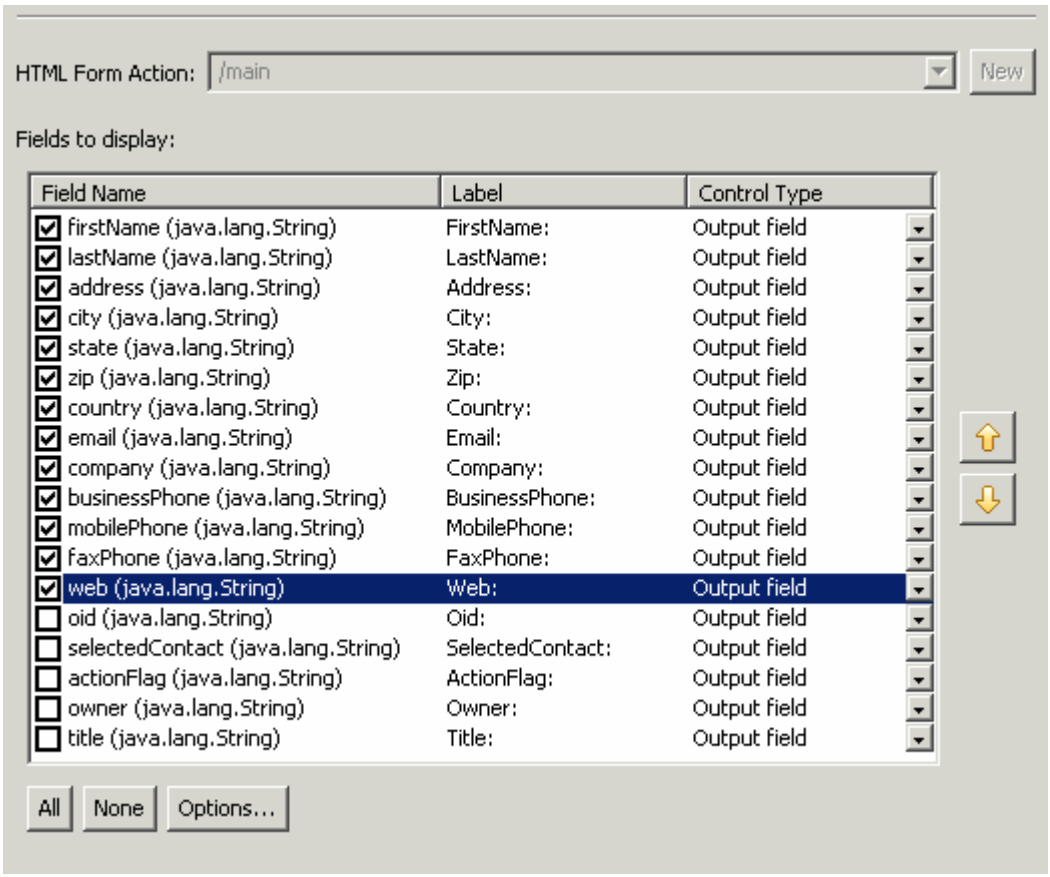
__ e. In the Page Data view, expand **Struts** and select **contact(contactlist.forms.Contact)**. Drag-and-drop the selected item from the Page Data view to anywhere in the JSP editor. See the screen capture below for an illustration.



© Copyright IBM Corporation 2005. All rights reserved

__ f. The Insert Data Object wizard that comes up allows you to change the columns that will be displayed in the table. Using the screen capture below as a guide, organize and deselect rows in the Fields to display field. Click **Finish** when you are done.

NOTE: Use the up and down arrows on the right to organize the fields as they are shown in the following screen capture.



© Copyright IBM Corporation 2005. All rights reserved

__ g. The end result of the above steps should look like the screen capture below.

____ 4. Add a return link to the contactdetails.jsp page.

__ a. In the contactdetails.jsp JSP editor, click the **Source tab**.

__ b. Add a return link after the closing </TABLE> tag. Scroll to the bottom of the page, and add the line highlighted in the screen capture below to the file. For your convenience, these lines of code can be found in **<LAB_FILES>\<LAB_NAME>\snippets\snippet1.txt**.

```
            <TD><bean:write name="contact" property="web"></bean:write></TD>
        </TR>
    </TBODY>
</TABLE>

<P><html:link action="/main">Return to Main</html:link></P>
```

__ c. Save contactdetails.jsp. Type **Ctrl-S**.

__ d. Close the JSP editor.

____ 5. Explore the main.jsp JSP page.

__ a. In the Project Explorer view, ensure **Dynamic Web Projects > ContactList > WebContent** is expanded. Right click on **main.jsp** and select **Open** to open it in Page Designer.

© Copyright IBM Corporation 2005. All rights reserved

__ b. Switch to the **Source** tab. Scroll the editor down until you find the line of code that begins with "href=". This hyperlink uses the Portlet API to encode the URL correctly for the portal environment. The <portletAPI:EncodeNamespace> tag is used to encode the URL.

__ c. **Close** the editor.

____ 6. Create the /modify action mapping.

__ a. Open the HTML edit mode's Web diagram. In the Project Explorer view, expand **Dynamic Web Projects > ContactList > Web Diagrams**. Double-click on **html-edit.gph**.

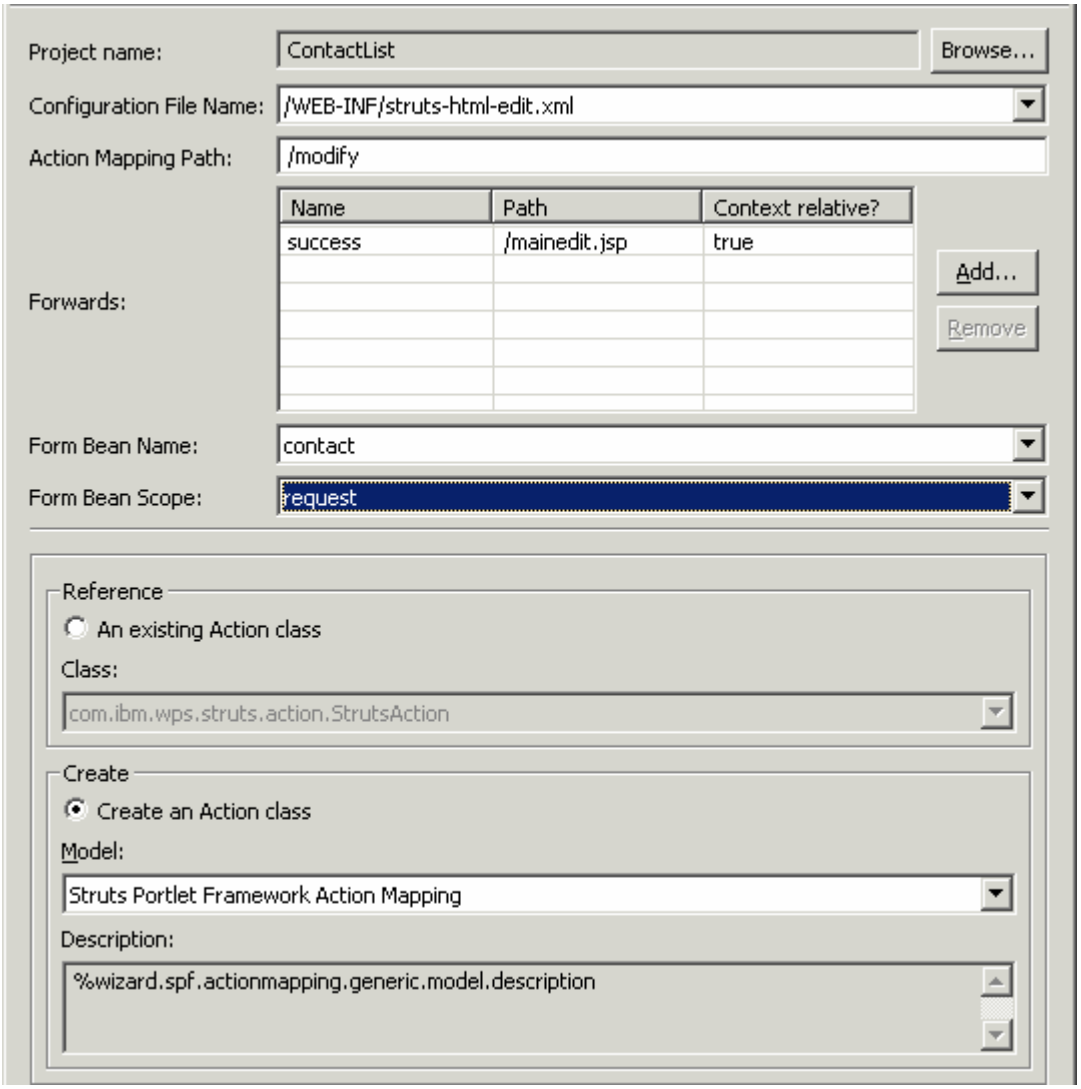
__ b. In the Web diagram editor for **html-edit.gph**, right-click on /**modify** and select **Create Action Mapping**. The New Action Mapping wizard will open up.

© Copyright IBM Corporation 2005. All rights reserved

__ c. In the New Action Mapping panel, use the table below to fill in the fields and click **Next >** (see the screen capture below for more details).

Field Name	Value
Form Bean Name:	contact
Form Bean Scope:	request
Model:	Struts Portlet Framework Action Mapping

NOTE: Notice that this action mapping is automatically associated with the Edit mode.



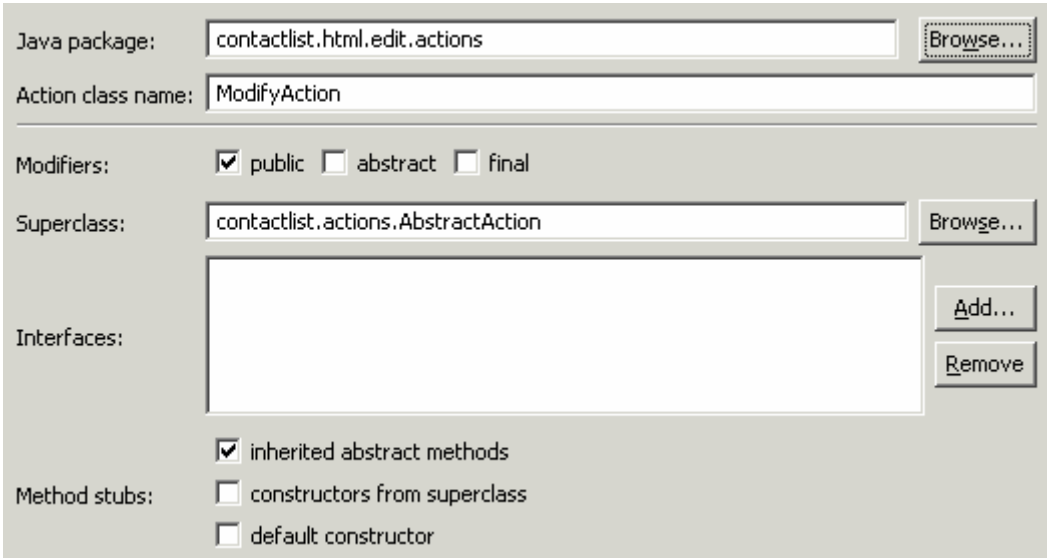
__ d. In the Create an Action class for your mapping panel, fill in the following information and click **Finish**.

1) Java package: **contactlist.html.edit.actions**

© Copyright IBM Corporation 2005. All rights reserved

2) Superclass: **contactlist.actions.AbstractAction**

NOTE: The AbstractAction class extends the default com.ibm.wps.struts.action.StrutsAction class. It provides additional support for our ContactList application.



__ e. The modify action mapping is created and opened in the Java editor.

__ f. Save and close the html-edit.gph Web diagram editor.

____ 7. Copy the entire contents of the **<LAB_FILES>\<LAB_NAME>\snippets\snippet2.txt** file into the ModifyAction class.

__ a. Open the **<LAB_FILES>\<LAB_NAME>\snippets\snippet2.txt** file in NotePad.

__ b. Type **Ctrl-A** and then **Ctrl-C** to copy the entire contents of the text file.

__ c. In Application Developer, ensure that the ModifyAction Java editor is selected.

__ d. In the Java editor, type **Ctrl-A** and then hit the **Delete** key.

__ e. After the original text has been deleted hit **Ctrl-V** to paste the entire contents of the snippet into the Java editor.

__ f. Save the changes. Type **Ctrl-S**.

__ g. Close the Java editor.

____ 8. Import the remaining implementation details for the portlet.

__ a. In the Project Explorer view, expand Dynamic Web Projects. Right-click on **ContactList** and select **Import… > Import…**. The Import wizard opens up.

__ b. In the Select panel, select **Zip File** and click **Next >**.

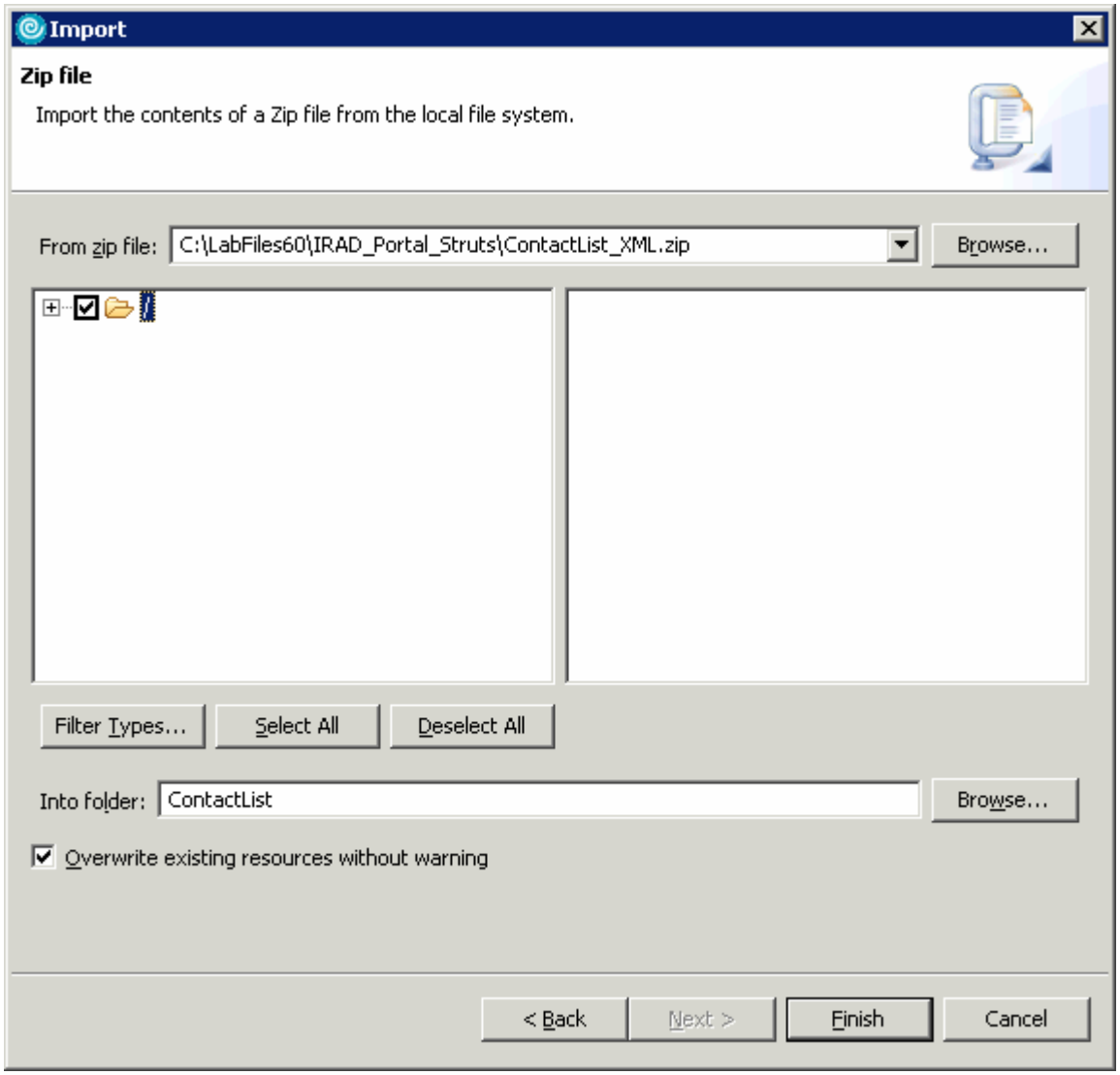
__ c. In the Zip file panel, fill in the following values (see the screen capture below for more details) and click **Finish**.

1) From zip file: **<LAB_FILES>\<LAB_NAME>\ ContactList_XML.zip**

© Copyright IBM Corporation 2005. All rights reserved

2) Into folder: **ContactList**

3) Place a check mark in the box next to **Overwrite existing resources without warning**



__ d. The errors and warnings in the Problems view will disappear.

© Copyright IBM Corporation 2005. All rights reserved

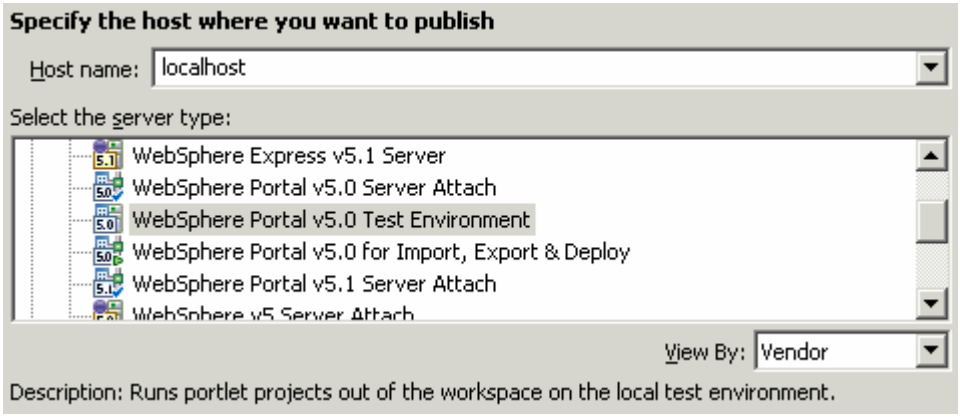
Part 5: Test the Struts-Portlet Application on the Portal Test Environment

In this part, you will use the Portal Test Environment to run and test the Struts-Portlet application that you created above.

____ 1. Run the Struts Portlet Application on a new server configuration.

__ a. In the Project Explorer view, expand **Dynamic Web Projects**. Right-click on **ContactList** and select **Run > Run on Server…**.

__ b. In the Server Selection window, select '**Manually define a server**'. In the Select the server type box, select **WebSphere Portal v5.0 Test Environment**. Click **Finish**.



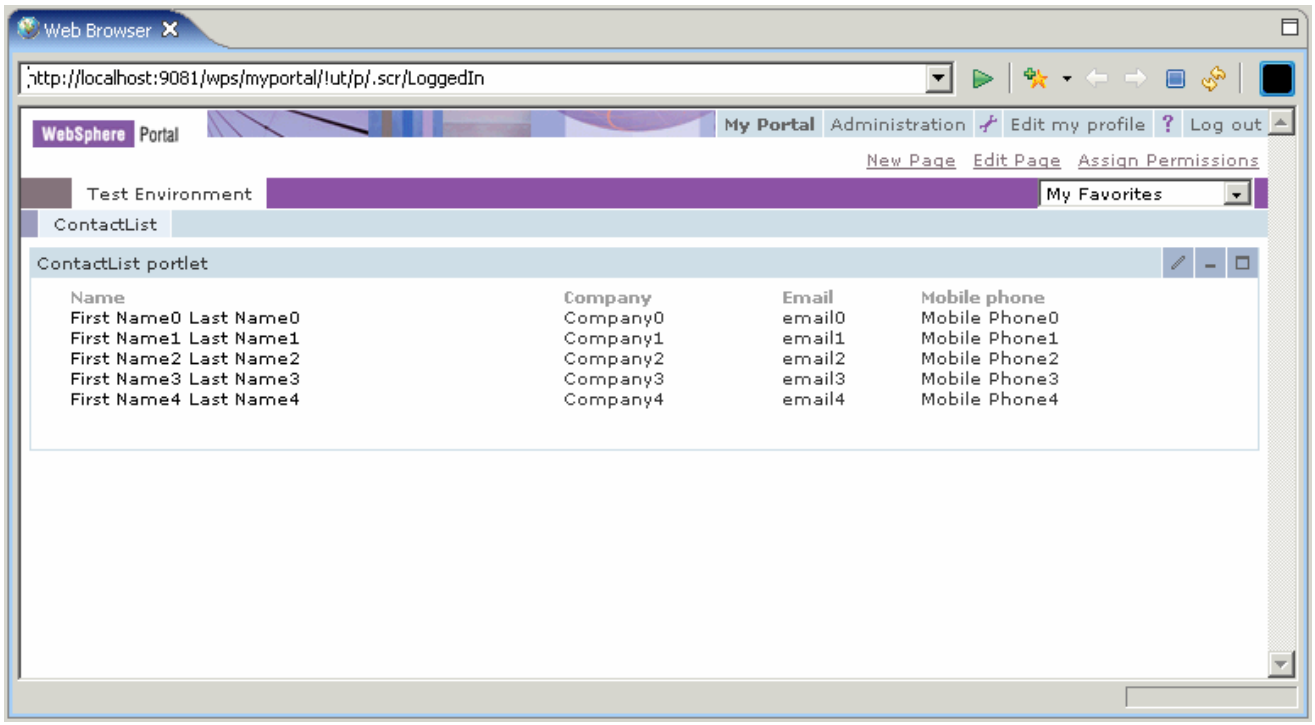
__ c. A new server and server configuration will be automatically generated, and the Portlet Project will be added to this server. The server will then start, and a web browser will open to display your portlet.

NOTE: Depending on the size of the system on which you are working, the starting of the Portal server may take up to ~5 minutes.

____ 2. Explore the View mode of your portlet.

© Copyright IBM Corporation 2005. All rights reserved

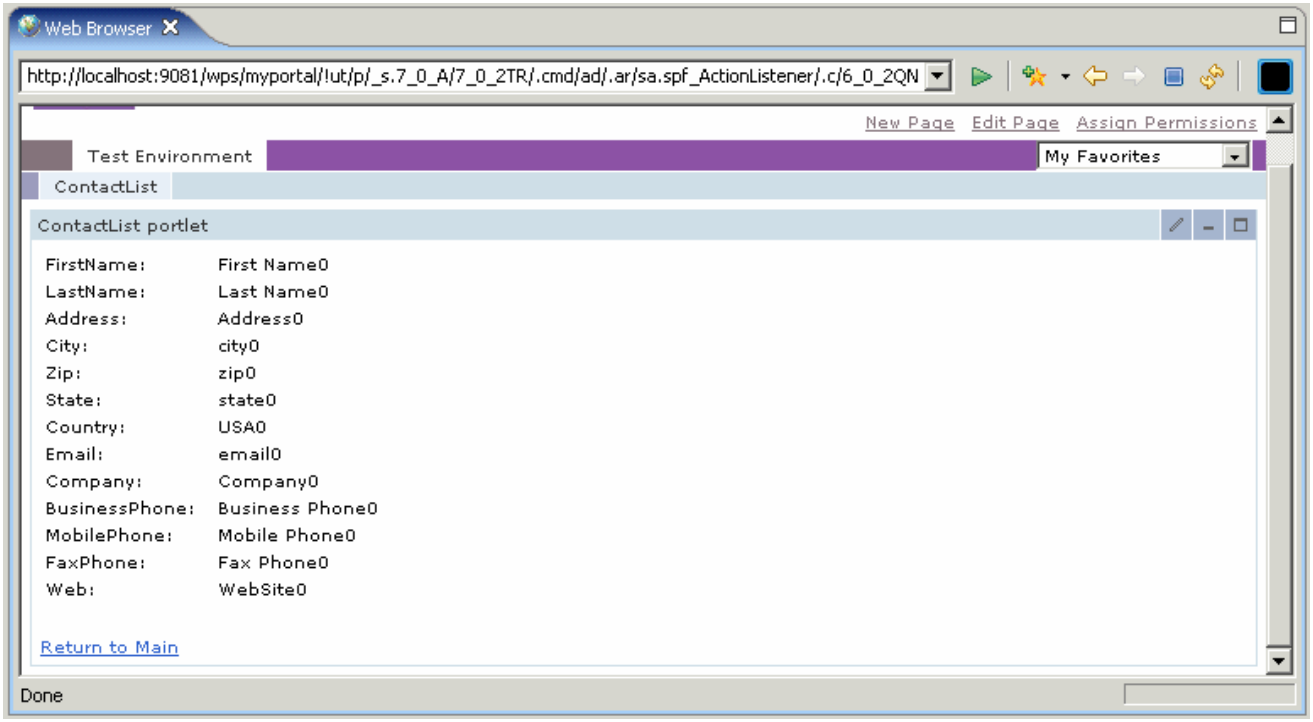
__ a. When the ContactList portlet comes up, the View mode of the portlet is displayed by default. Remember that the view mode initially calls the index.jsp JSP page, which forwards on to the main.jsp JSP page. As a result, the main.jsp JSP page displays a list of contacts.



© Copyright IBM Corporation 2005. All rights reserved

__ b. Click on any name in the table to view the contact's detailed information.

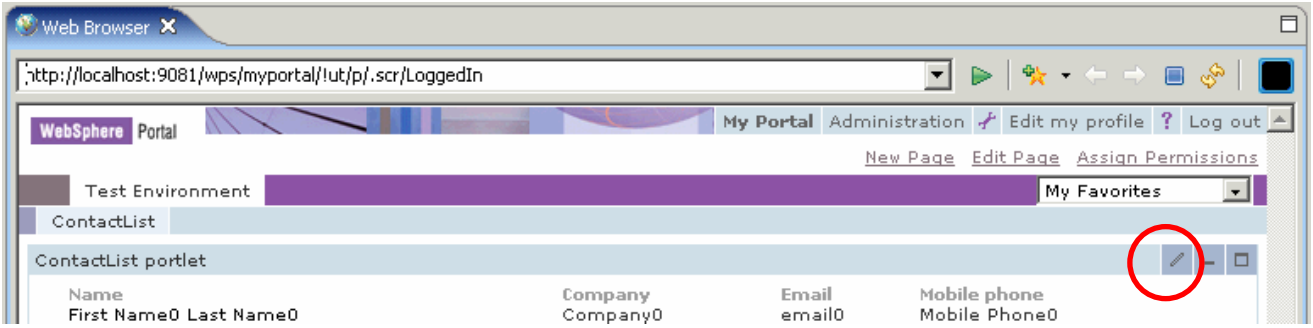
NOTE: Notice that the view mode of the application does not allow you to edit the contact's information.



__ c. Click the **Return to Main** link at the bottom of the portlet to return to the list of contacts.

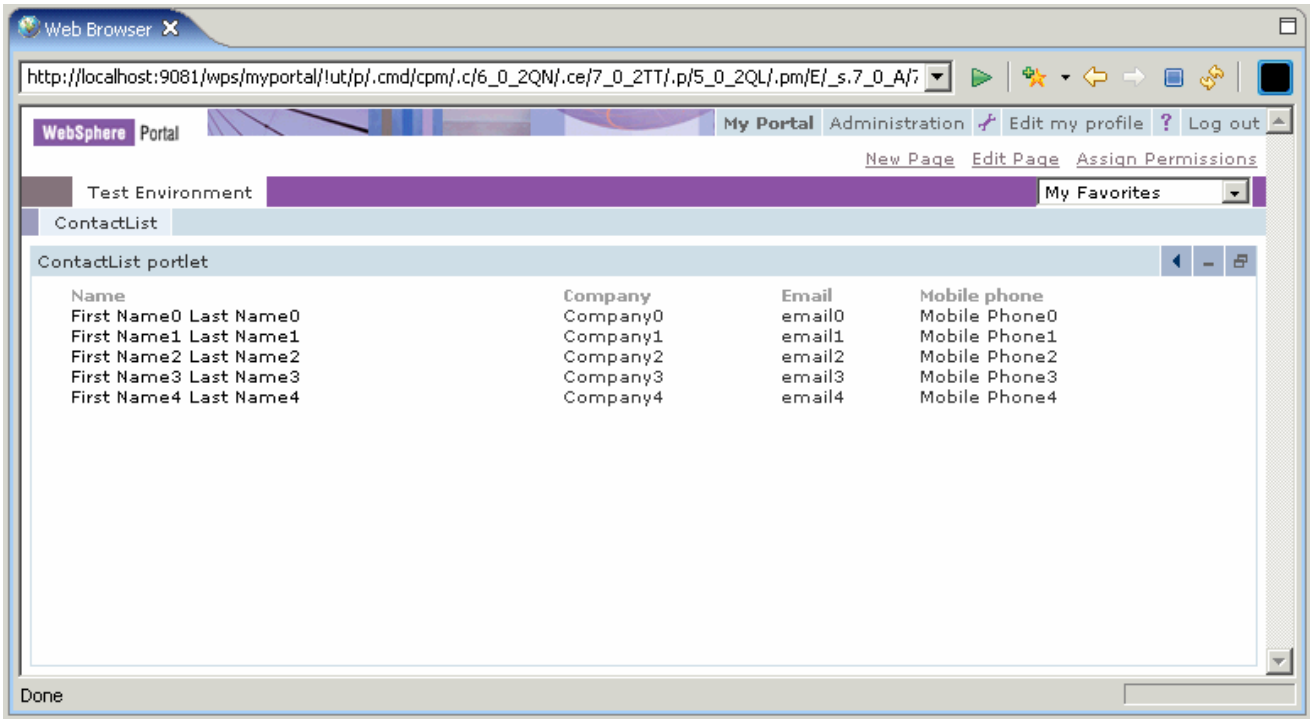
____ 3. Explore the Edit mode of the Contact List portlet.

__ a. Click on the **Edit button** (circled in the screen capture below) icon in the upper right corner of the portlet.



© Copyright IBM Corporation 2005. All rights reserved

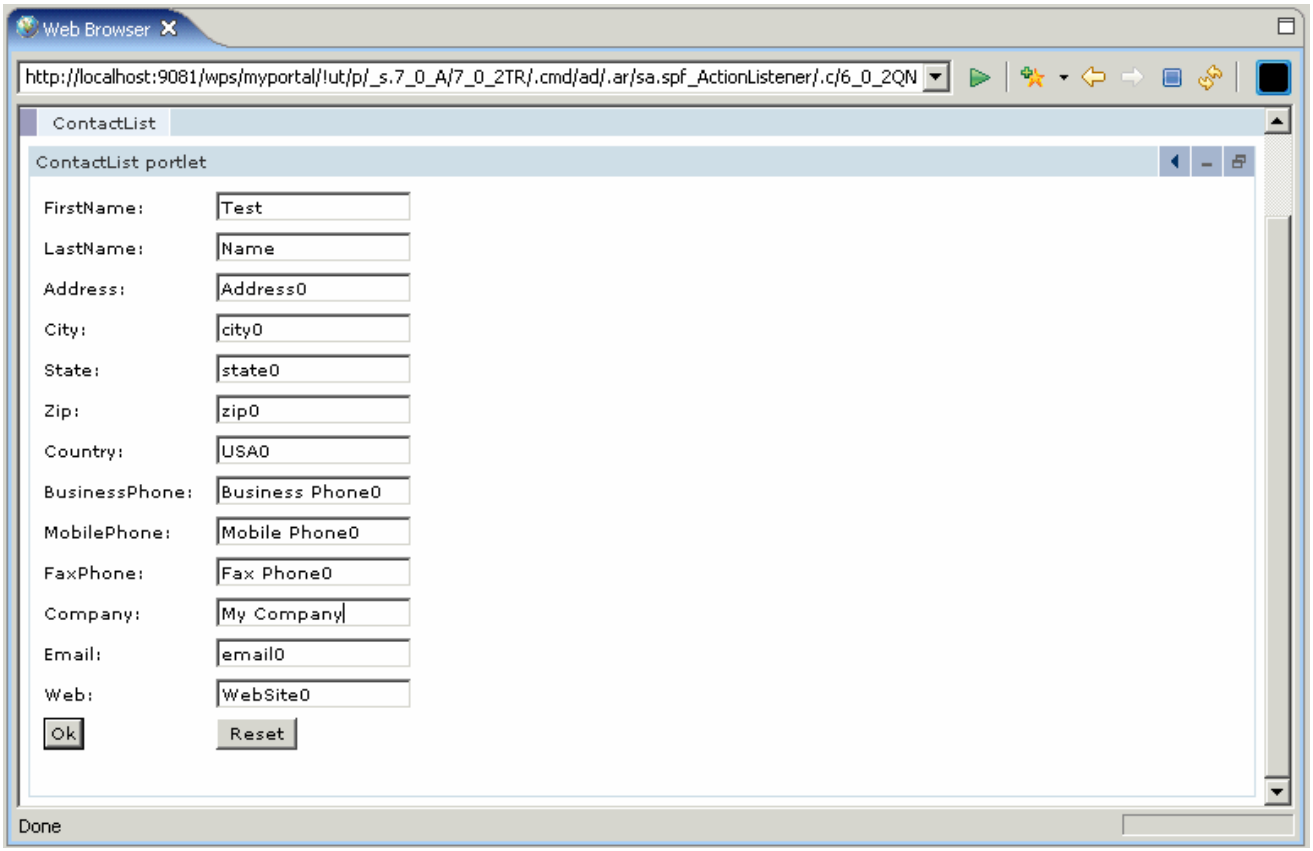
__ b. The initial page of the Edit mode is displayed. The application is designed to display the same list of contacts as the main page of the View mode.



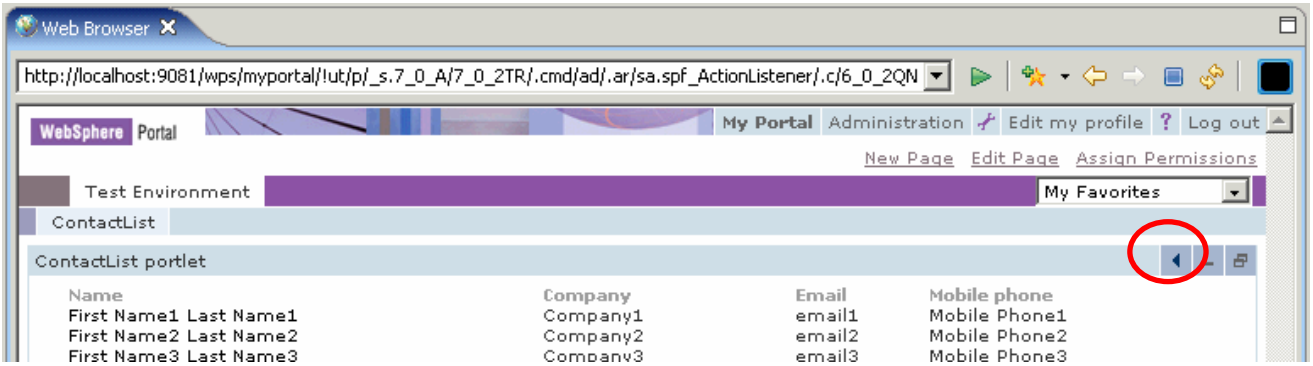
__ c. Click on any name in the table to view the contact's detailed information.

© Copyright IBM Corporation 2005. All rights reserved

__ d. Edit the information for the contact, and click the **Ok** button at the bottom of the form (see the screen capture below for more details).



__ e. Return to the portlet's view mode. Click on the **Back button** (circled in the screen capture below) icon in the upper right corner of the portlet.



© Copyright IBM Corporation 2005. All rights reserved

__ f. On the portlet's view page, notice that the changes made in the edit mode are now reflected in the contact list table.

__ g. Continue to explore the application until you are satisfied with what you have seen.

____ 4. Stop the test server.

__ a. Switch to the Servers view by selecting **Window > Show View > Servers**.

__ b. Right-click on **WebSphere Portal v5.0 Test Environment @ localhost** and select **Stop**.

__ c. Watch the Console view and wait for the server to come to a complete stop.

____ 5. Exit Application Developer. Select **File > Exit**.

© Copyright IBM Corporation 2005. All rights reserved

What you did in this exercise

In this exercise, you used Rational Application Developer v6.0 to build a Struts portlet application.

© Copyright IBM Corporation 2005. All rights reserved

Solution Instructions

____ 1. Start Rational Application Developer.

__ a. Select **Start > Programs > IBM Rational > IBM Rational Application Developer V6.0 > Rational Application Developer**.

__ b. A dialog box will be displayed allowing you to select the location where you would like the workspace directory to be stored. Enter **<LAB_FILES>\<LAB_NAME>\solution\workspace** for the location and select **OK**. Application Developer will start with an empty workspace. An empty workspace will leave your existing workspace untouched and help avoid name conflicts between what you may already have in your workspace and what you will be creating in this lab.

____ 2. Import an EAR that contains the completed portlet.

__ a. Select **File > Import…**. The Import wizard will open up.

__ b. In the Select panel, select **EAR file** and click **Next >**.

__ c. In the Enterprise Application Import panel, for the EAR file field enter **<LAB_FILES>\<LAB_NAME>\solution\ContactListEAR.ear**. Change the Target server field to **WebSphere Portal v5.0**. Click **Finish**.

____ 3. Test the portlet application.

__ a. Switch to the Web Perspective. Select **Window > Open Perspective > Web**.

__ b. Follow the steps in Part 5 to test the completed portlet application.

© Copyright IBM Corporation 2005. All rights reserved

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.

© Copyright IBM Corporation 2005. All rights reserved

This page intentionally left blank