
Web Tools Demo #2 - Subfiles

In the first Web Tools demo #1 named Introduction to Web Tools, you saw how we created an e-business RPG customer inquiry application. When we ran our Web application a 'new' Web customer inquiry page opened in the workbench Web browser. We then entered a customer number and clicked the submit button. The customer details from a DB2/400 database appeared in the workbench Web browser. After seeing this demo you should have a good introduction to the Web Tools available in the Development Studio Client.

In this Web Tools demo #2, you see how we create a Web browser version of a subfile to display a list of customers and then display the details for each selected customer.

In demo #3, you will then see how we add these new input and output pages to the customer inquiry e-business RPG application we created in the first Web Tools demo.

The Steps in this Demo

- Create a Web project
- Use the Page Designer's iSeries Web Components
- Use the Web Interaction wizard
- Run the Web application using the WebSphere Test Environment

Creating a new Web project

In order to create a Web interaction, you will need to use the Web perspective in the Development Studio Client workbench. You will see how to build a subfile for a Web browser by:

- Creating a Web project
- Using the Page Designer's iSeries Web Components
- Using the Web Interaction wizard
- Testing the subfile by:
 - Using the **WebSphere Application Server** test environment
 - Invoking the built-in workbench browser and showing your Web page

Opening the Web perspective

The Development Studio Client shows the Remote System Explorer (RSE) perspective by default. This is the perspective that you would use to work with iSeries objects. It allows specifying connections to iSeries servers and provides for the programmer a similar interface to iSeries objects as the Program Development Manager (PDM) does in a green-screen environment.

You are a Web developer in this demo and that is the reason you will select the Web perspective.

The Web environment provides its own perspective since it needs to give its users access to unique views and tools targeted towards Web tasks.

To create a Web project you first need to open the Web perspective as shown in Figure 4.

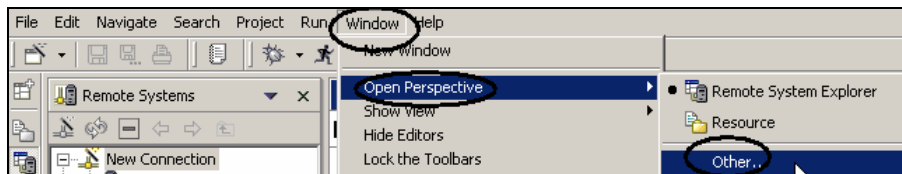


Figure 4: Opening a perspective

Do this by:

- Selecting the **Window** menu item on the workbench menu bar.
- Then select **Open Perspective**
- Select **Web**.

The workbench will now show the Web perspective with the J2EE Navigator view open on the left hand side. You are ready to create a Web project.

Creating a Web project

In the workbench now:

- Select the New wizard icon on the workbench menu bar to display the New wizard.
- Select the **Web** from the left hand pane
- Select **Web Project** from the right hand pane

- Click the *Next* > push button.

This will start the Web project wizard.

The Web Project Wizard

In the Web project wizard you will be presented with the initial page.

On the Web Project page:

Create a Web Project

Define the Web Project
Create a Web Project.

Project name:

Use default

New project location:

J2EE Web Project Static Web Project

Description:
In a J2EE Web Project you will be able to create content served by a traditional HTTP server (HTML, JavaScript, images, text..) as well as content to be served by a J2EE Application Server (Servlets, JSPs, EJBs..)

Web Project features:

- Add Struts support
- Create a default .cvsignore file
- Create a default CSS file
- Include Tag Libraries for accessing JSP ot
- Include Tag Libraries for database access
- Include Tag Libraries for internationalizati

Description:
Select this feature to have support for Struts added to your project

< Back Next > Finish Cancel

Figure 7: Completing the Create a Web Project page

- Type *wdscsubf* in the **Project Name** entry field.
- Make sure the **J2EE Web Project** radio button is selected.

- Select the *Add Struts support* checkbox.
- Click the *Next >* push button.

The *J2EE* Setting page opens.

J2EE Setting (Figure 8)

This page allows you to specify the J2EE level you want for the Web project to be created. You might wonder what that means to you. If you plan to use the WebSphere Application Server version 5 only then you should select J2EE level 1.3. If you are using previous versions of WebSphere Application Server in your environment and are planning to deploy the Web application to these versions then you should use J2EE level 1.2 since these older versions of WebSphere Application Server don't support the J2EE level 1.3. In this exercise you will use the Web application in the WebSphere Test Environment, which is version 5, so you will use J2EE level 1.3.

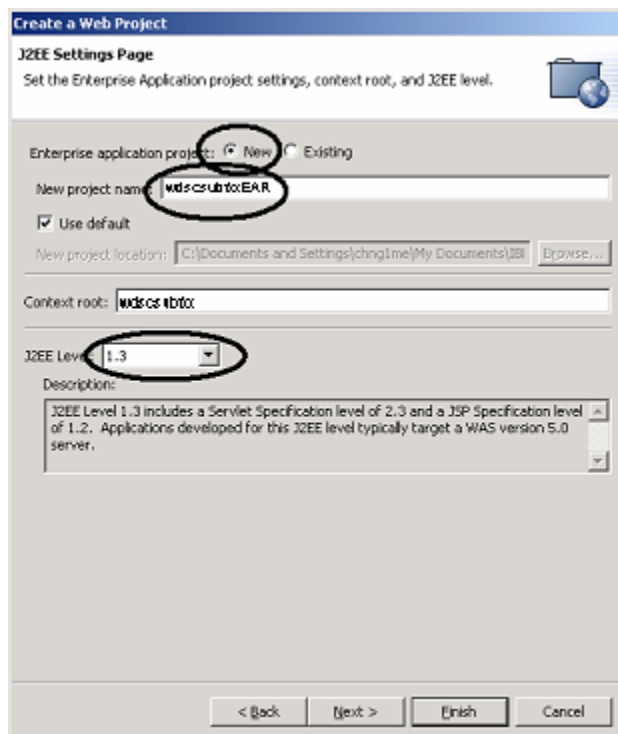


Figure 8: Page to select JSP level

- Select the **New** radio button, to specify that you want to create a new Enterprise application project
- Enter **wdscsubfxxEAR** in the **New project name** entry field
- Click on the **little arrow** beside the **J2EE level** combination box
- Select **1.3** from the list
- Click the **Finish** push button

Your Web project is created.

Using the Page Designer's iSeries Web Components

Lets create the Customer List Page:

- Navigate to the **Web Content** folder in your project
- Right mouse click on the folder and select **New > JSP File** from the pop-up menu.

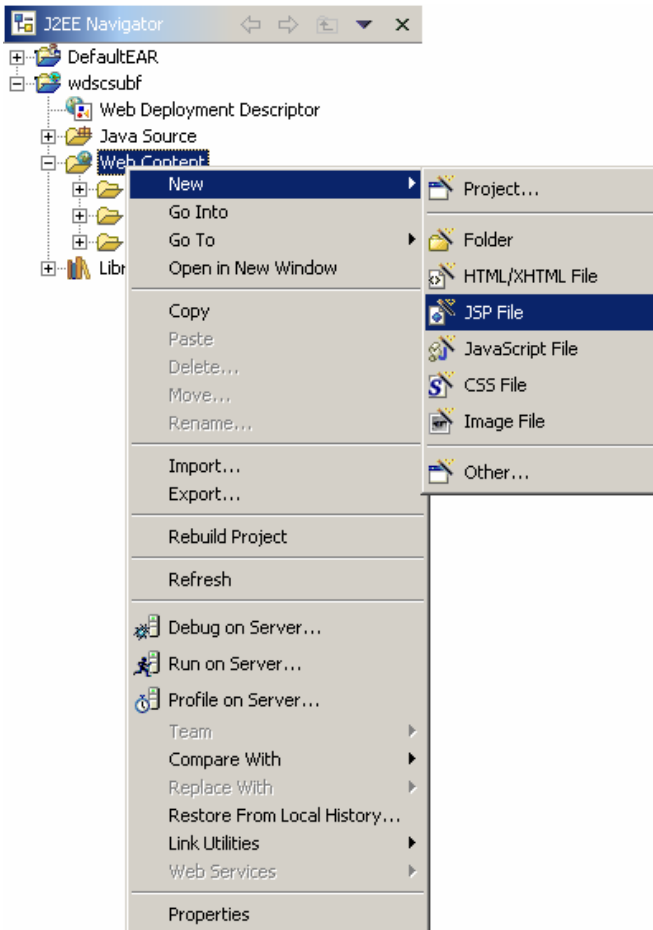


Figure 9: Creating a JSP File

- The *New JSP File* dialog will appear.

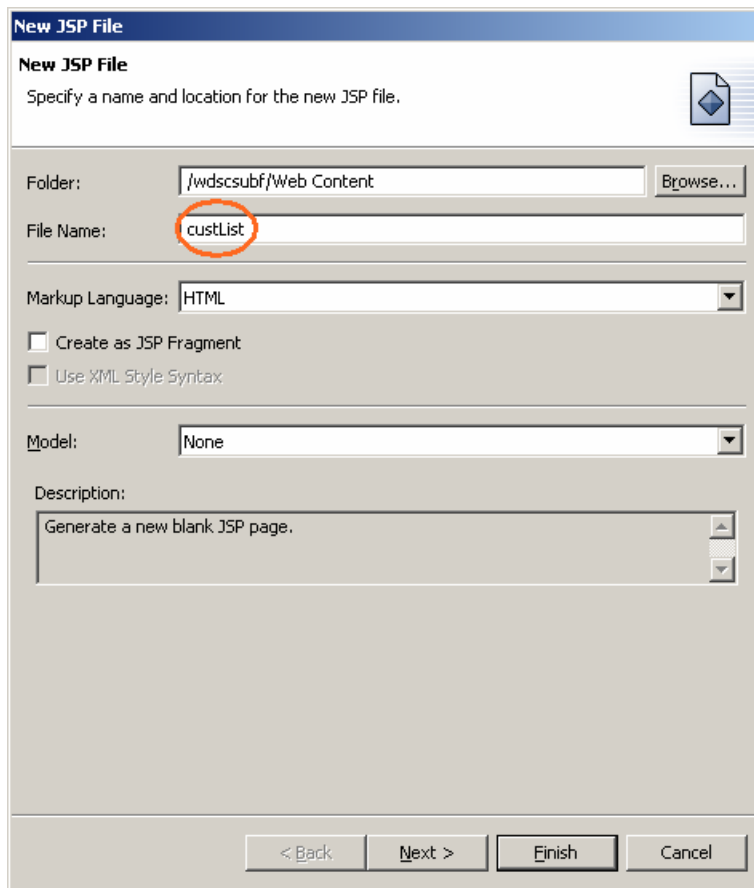


Figure 10: Completing the New JSP File page

- Type **custList** in the **File Name** entry field
- Click the **Finish** push button.

This will create a new JSP file and will automatically open the **Page Designer** in the workbench.

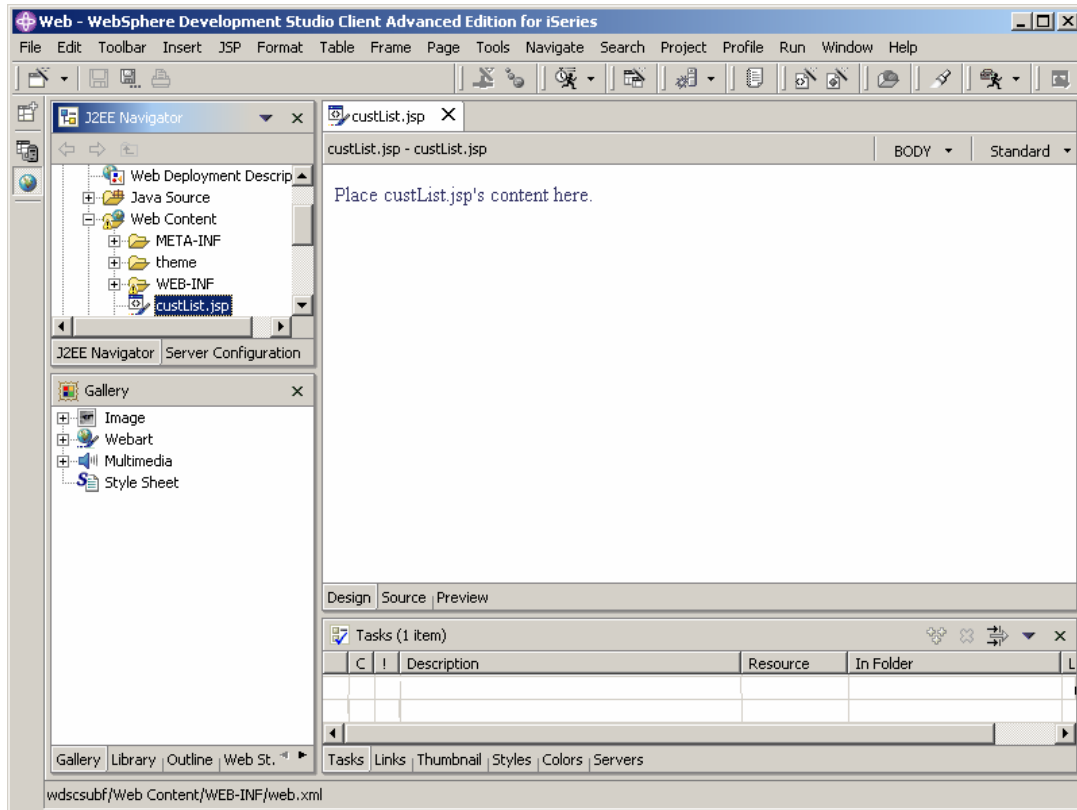


Figure 11: Page Designer in the Workbench

In the bottom left of the workbench, select the **Library** tab

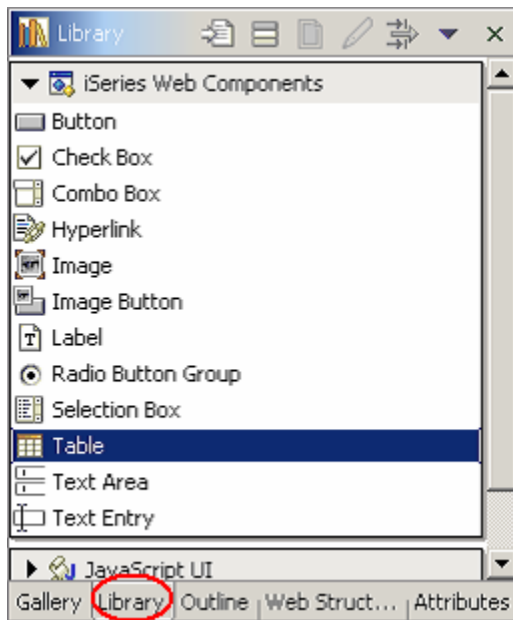


Figure 12: Selecting the iSeries Table Component

- Expand the **iSeries Web Components**.
- Select **Table** and drag it to your JSP under the text “**Place custList.jsp’s content here**”
- The Page Designer should look like this (Figure 13):

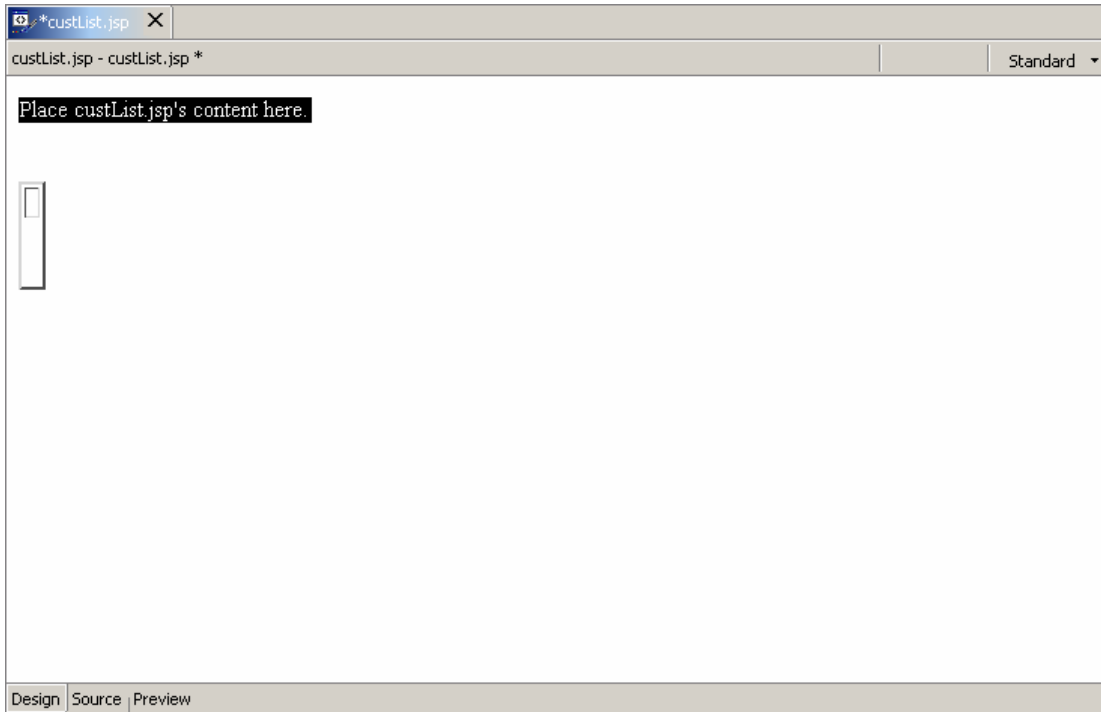


Figure 13: Placing the iSeries Table on the page

The *Attributes* pane in the bottom left should open automatically

- In the *attributes* window, select the *General* tab (Figure 15).

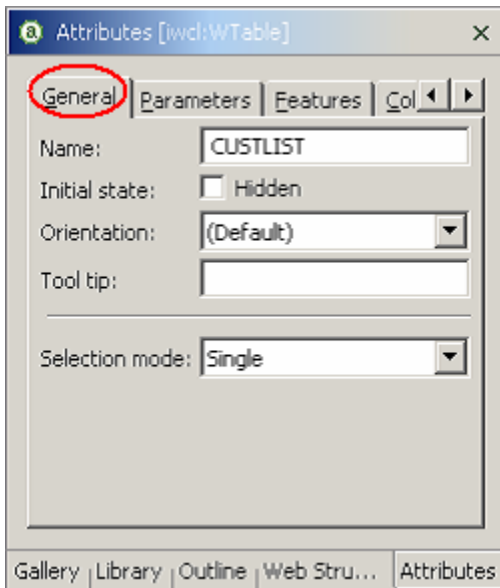


Figure 15: Filling in the General Attributes of the iSeries Table

- Fill in **CUSTLIST** as the **Name** of the table
- Select **Single** as the **Selection mode** of the table
- Now click on the **Parameters** tab (Figure 16).

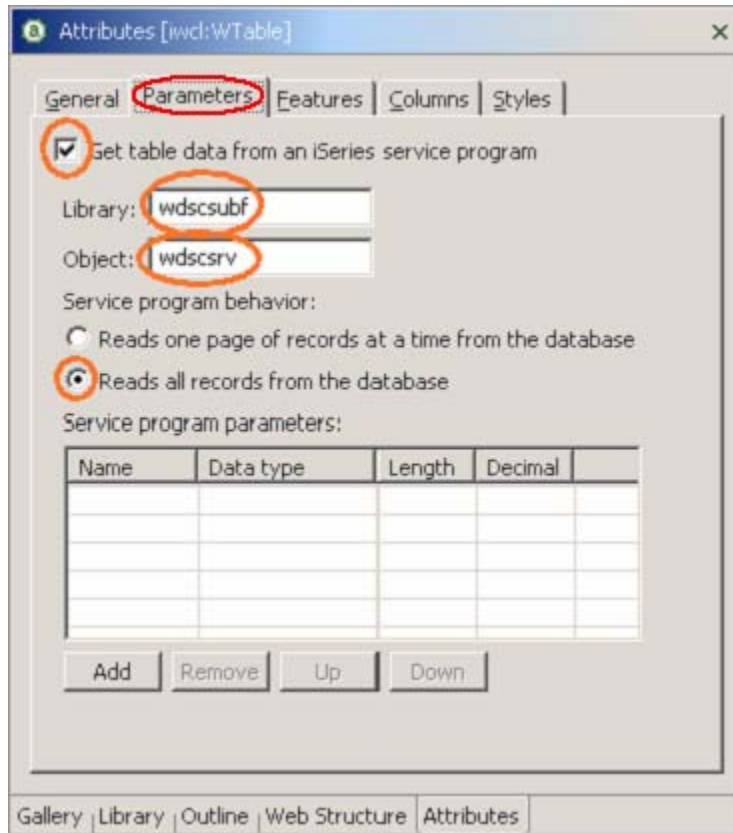


Figure 16: Setting the Parameters of the iSeries Table

- Check the checkbox labeled **Get table data from an iSeries service program**.
- Type **wdscsubf** in the **Library** entry field
- Type **wdscsrv** in the **Object** entry field.
- Select the radio button labeled **Reads all records from the database**
- Click the tab labeled **Columns**. This will switch to the column attribute pane (Figure 17).

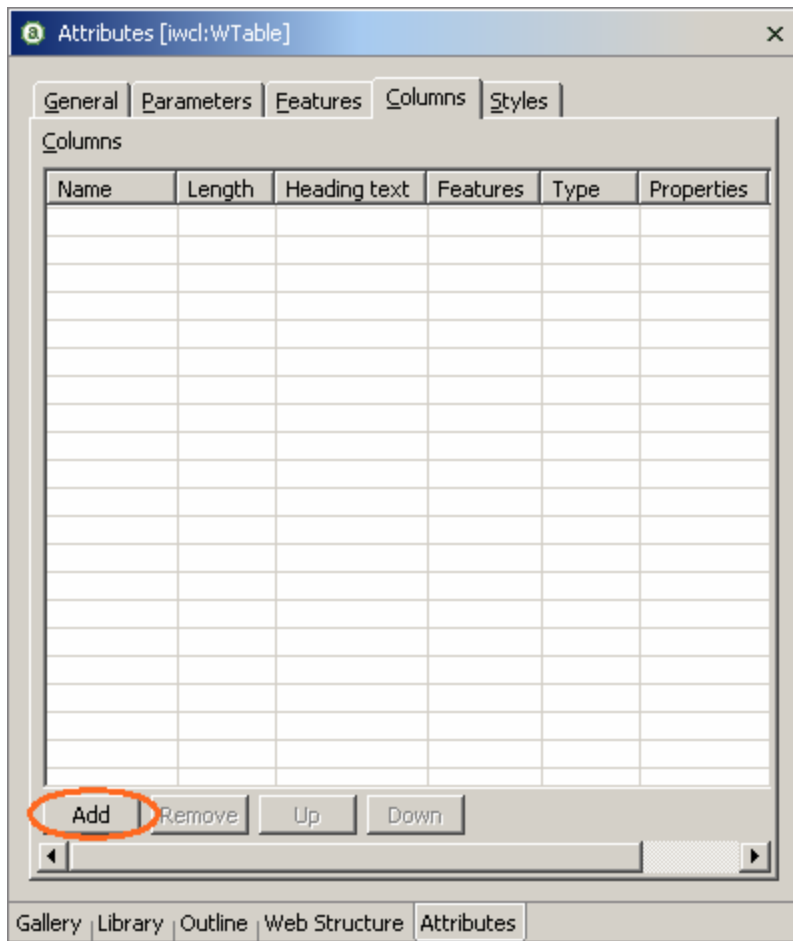


Figure 17: Setting the Columns of the iSeries Table

- Click the **Add** push button.
- Fill in the following, and click **Add** after each row.

NAME	LENGTH	HEADING TEXT
custNumber	7	Customer Number
custName	40	Customer Name
custPhone	17	Customer Phone Number

Your completed Attributes pane should look like this (Figure 18):

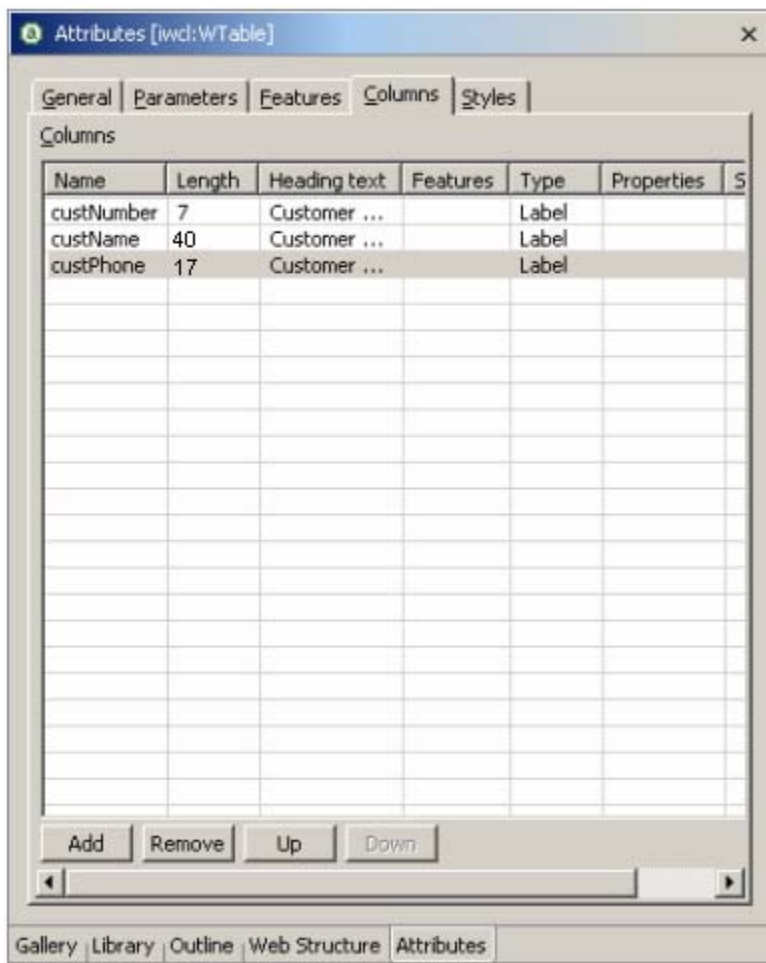


Figure 18: Completed Table Column attributes

- Now, place your cursor below the table in the Page Designer and select **Insert** → **Form and Input Fields** → **Form** (Figure 19)

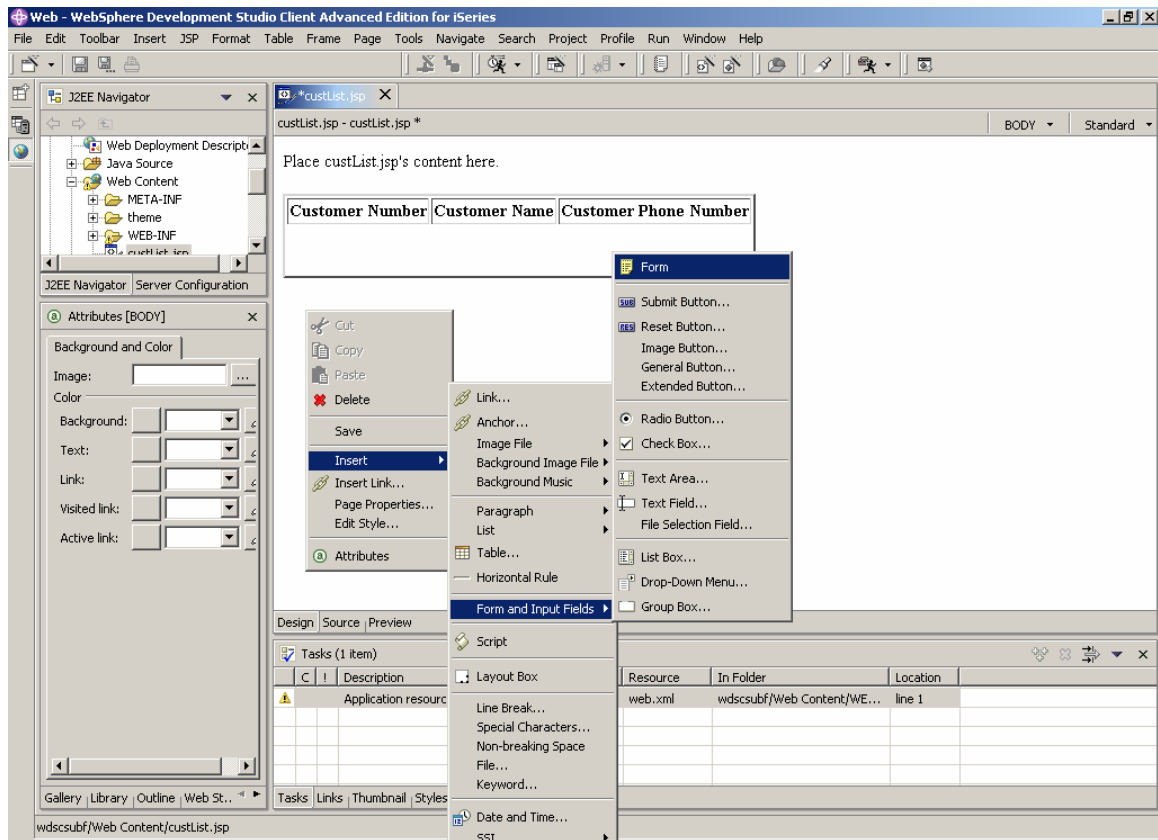


Figure 19: Inserting a Form on the page

- Place your cursor inside the dotted rectangle, which represents the newly created form (Figure 20).

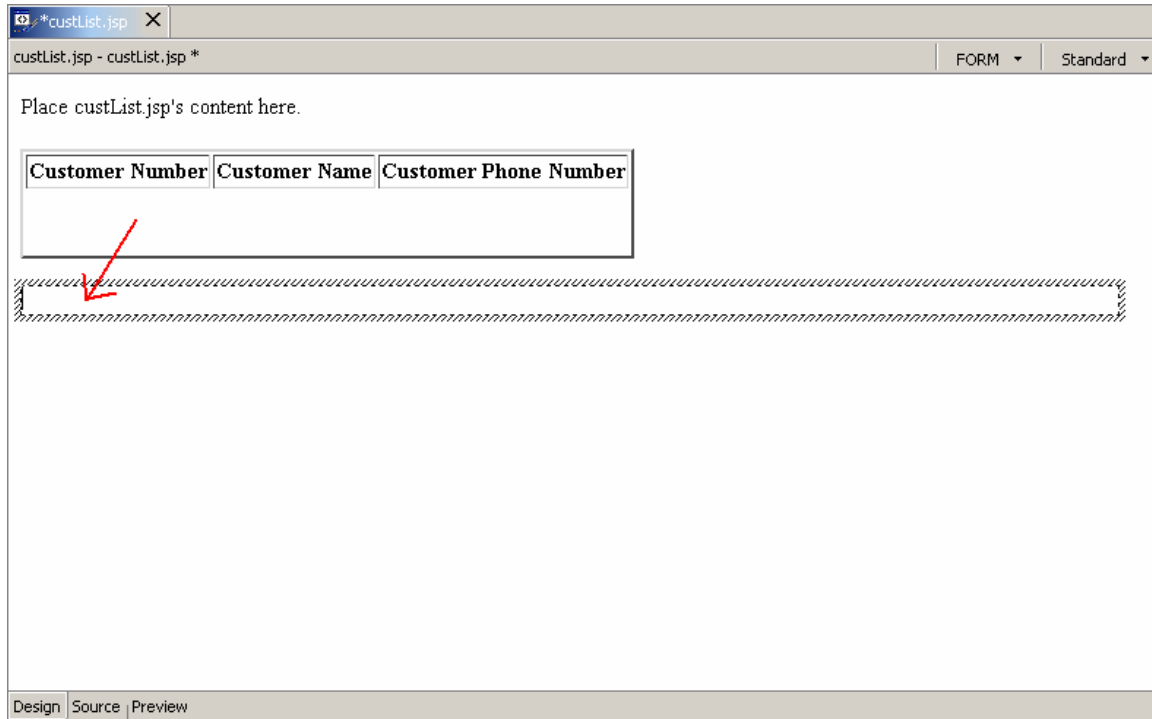


Figure 20: Newly created form

- Navigate back to the **Library** view by clicking the **Library** tab at the bottom left (Figure 21).

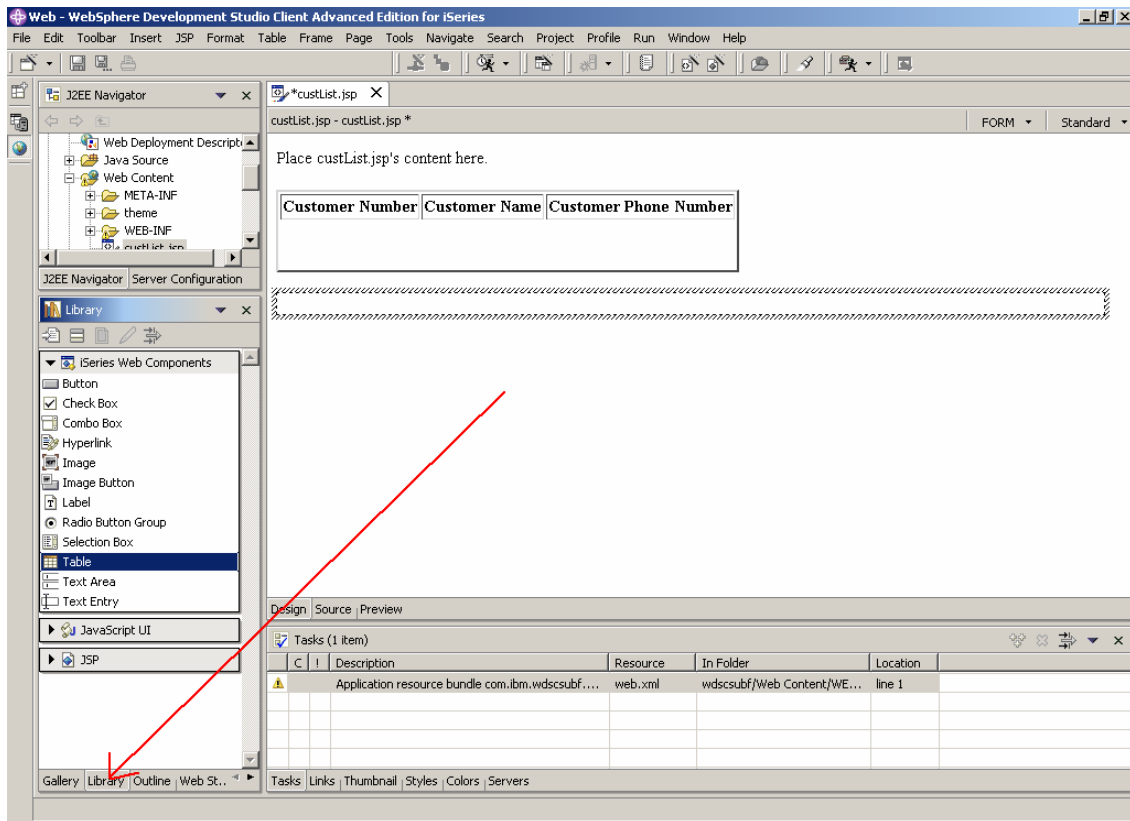


Figure 21: Accessing Library pane

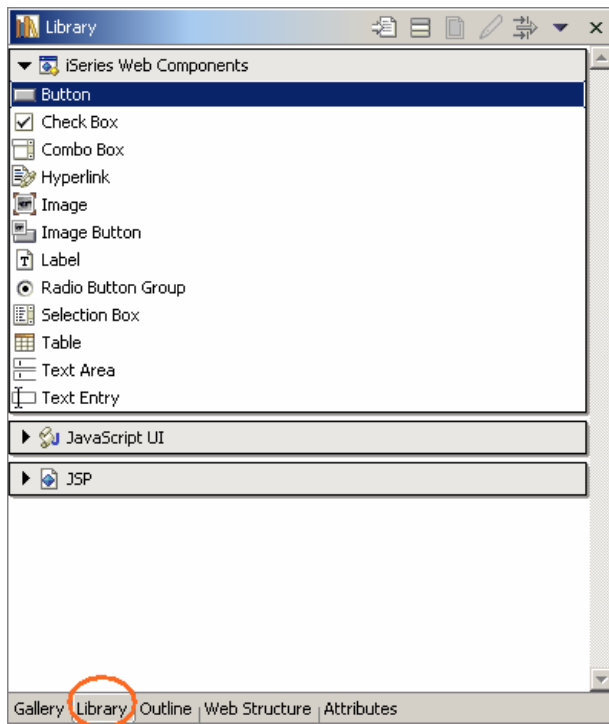


Figure 22: Selecting an iSeries Button Component

- Now, expand **iSeries Web Components**.
- Select **Button** then drag it to the form and drop it.

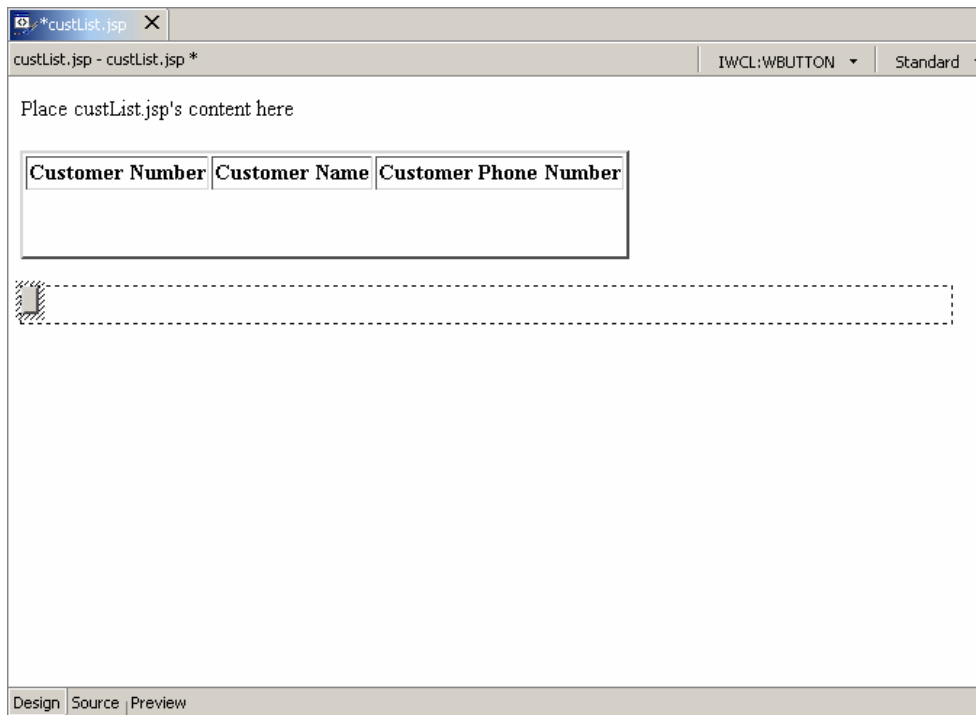


Figure 23: Placing the iSeries Button inside the form on the page

- The *Attributes* pane for the button opens.

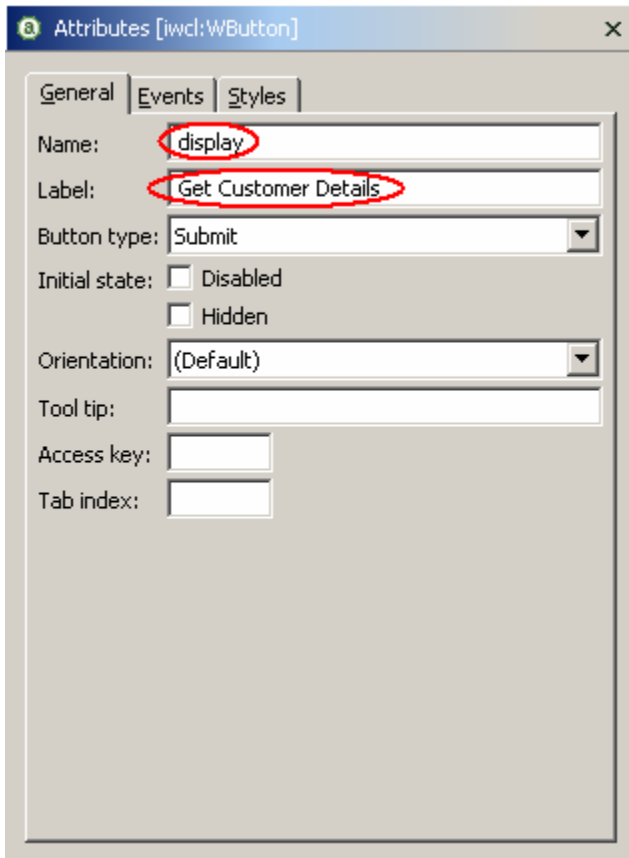


Figure 24: Completing the iSeries Button attributes

- Type **display** in the **Name** entry field.
- Type **Get Customer Details** in the **Label** entry field.

You have finished creating and modifying the iSeries Web Components. Let's add a heading to this page.

- Select the text *Place custList.jsp's content here*. Replace it with the text **Customer List**
- In the attributes pane on the bottom left, select **Heading 2**.

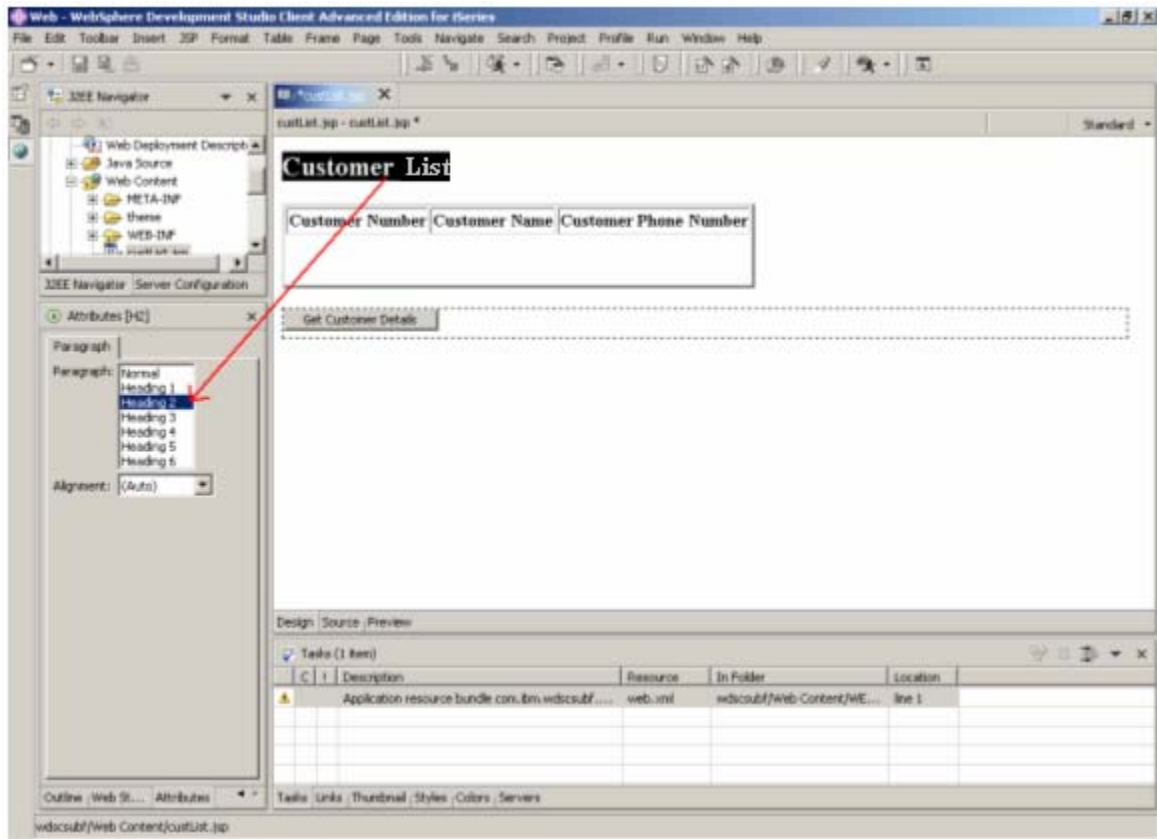


Figure 25: Selecting Heading 2

You are done creating the JSP. Your page should look like this:

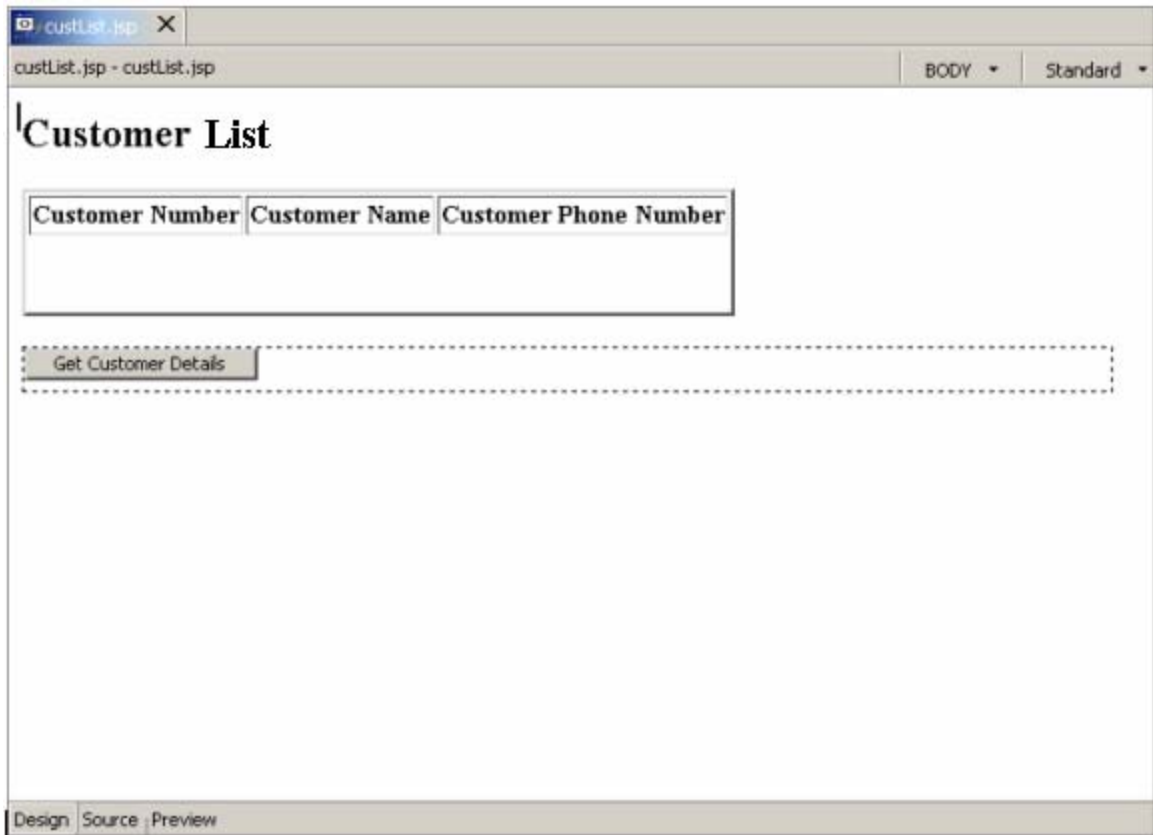


Figure 26: The finished page.

- If you haven't done so already, save the JSP file.

In order to proceed, we need to set up our connection settings to the iSeries machine.

Setting up the iSeries server information

Before you can start creating your application, you will need to define where the RPG program you will connect to later on in this demo, is located. You need to specify the **iSeries server name** and the **user-id/password** to be used when starting the job on the iSeries to run your program.

Go back to the Web perspective.

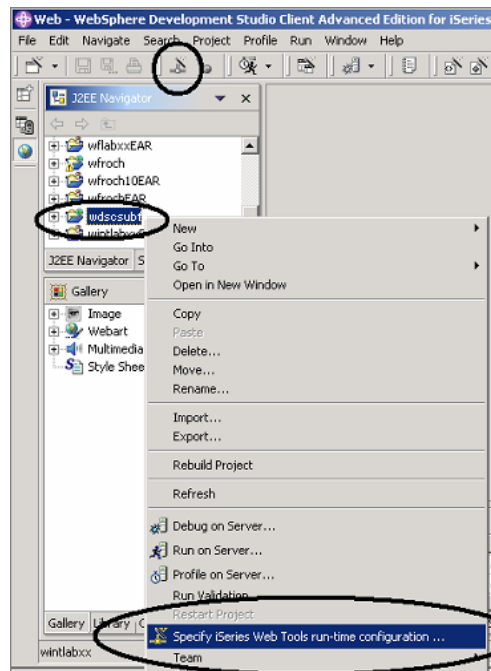



Figure 27: Specifying iSeries configuration

- Select your project ***wdscsubf***
- Right mouse click on ***wdscsubf***
- Select the **Specify iSeries Web Tools run-time configuration** option from the pop up menu

Tip: You could also use the  button from the workbench toolbar to get the Runtime configuration wizard

The following wizard will show:

iSeries Web Tools Run-time Configuration

Specify the run-time configuration for program call of iSeries Web development tools

iSeries host name: toras07m

User ID: wdscsubfxx

Password: *****

Runtime library list

WDSCSUBFXX	

Display detailed runtime errors

Use program call JCA connector

Specify the JNDI name: _____

Buttons: Add, Remove, Move up, Move down, Finish, Cancel

Figure 28: Setting the configuration

- Enter your **iSeries server name** into the **iSeries host name** entry
- Enter your user-id in the **User ID** entry field
- Enter your password in the **Password** entry field
- We need to add the **WDSCSUBF** library to the **Runtime library list**. Click the **Add** push button.
- Fill in **WDSCSUBF**
- Click the **Finish** push button on this dialog.

Now you are ready to use the Web Interaction wizard.

Using the Web Interaction Wizard

Now you are going to link the input page (custList.jsp) we created earlier to an iSeries RPG service program, which will receive the choice of which customer to display, then populate the detail page with the customer's full details. We can do this easily through the Web Interaction Wizard.

Creating a Web Interaction

Let's create a new Web Interaction to handle this request.

- In the *Web Perspective*, right mouse click the Web content folder and select **New > Other...** from the pop-up menu

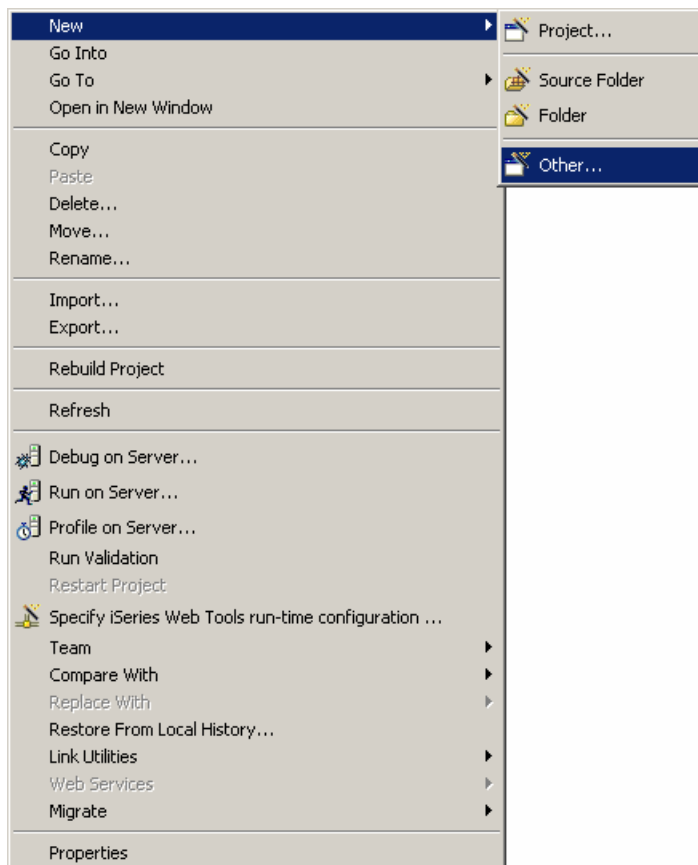


Figure 29: Creating a new Web Interaction

- In the *New* dialog, select **Web** from the left and **Web Interaction** from the right.

Tip: You can also select the  icon from the toolbar

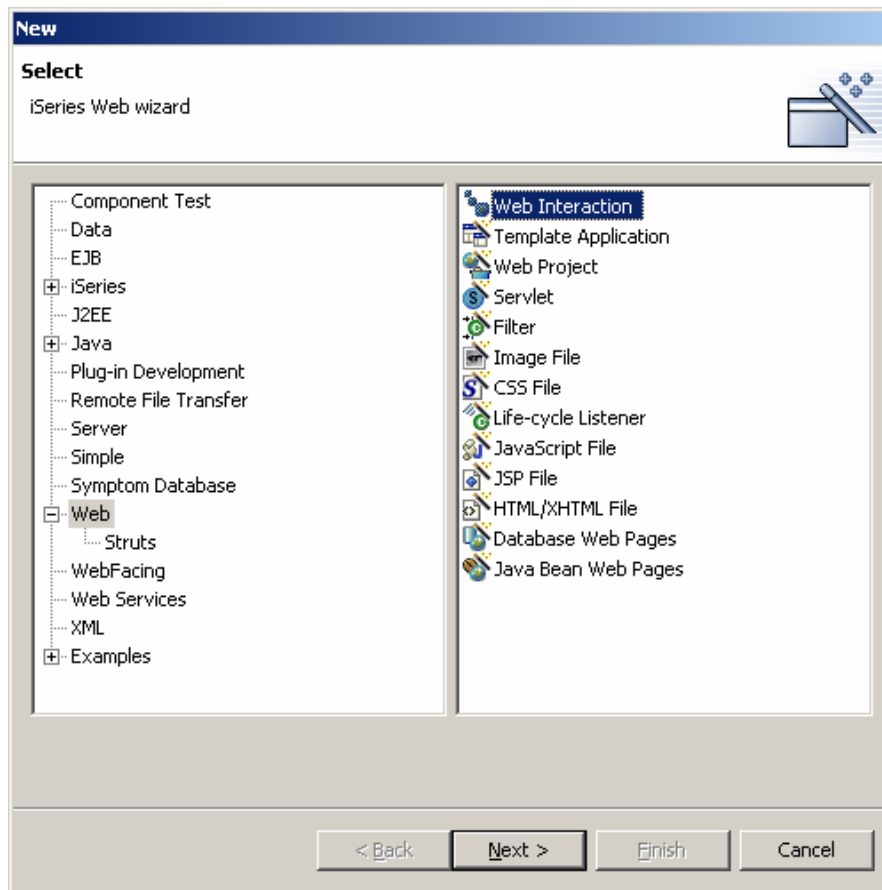
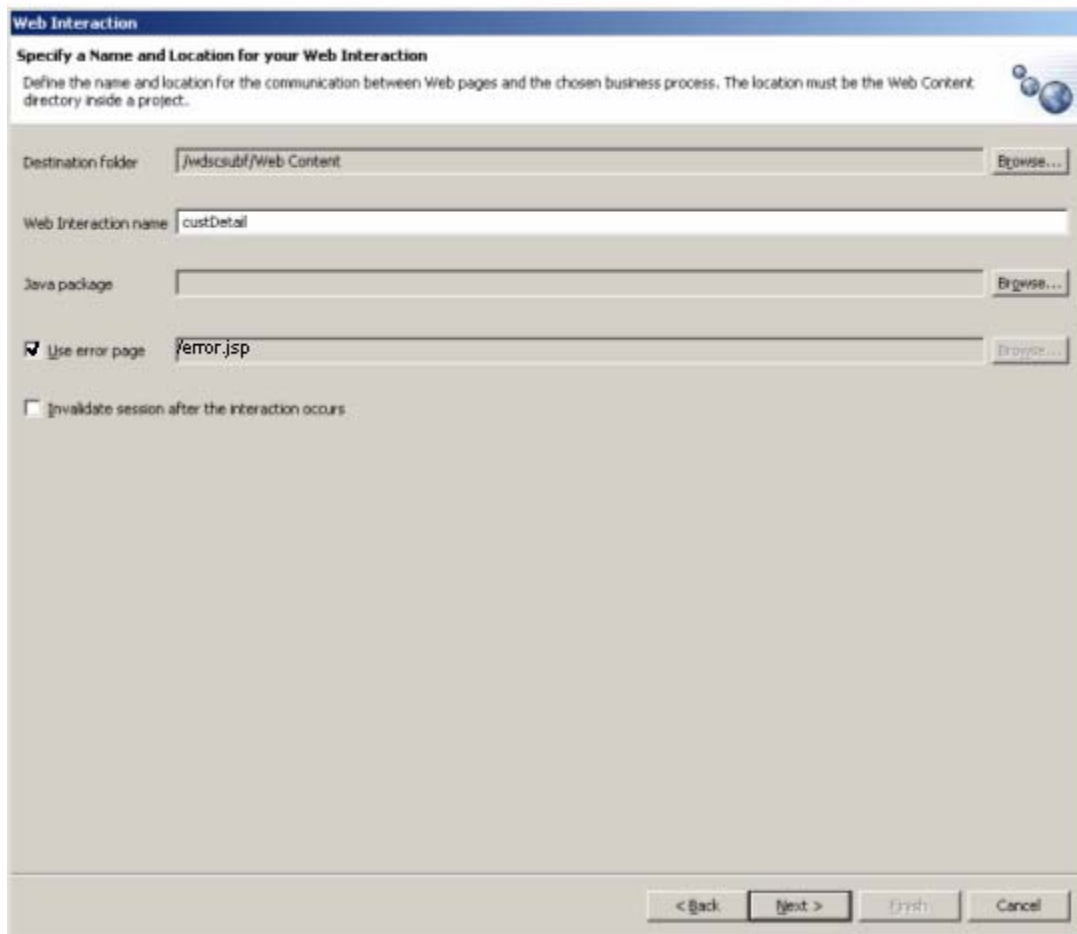


Figure 30: Creating a new Web Interaction (continued)

- Click the *Next* > push button.



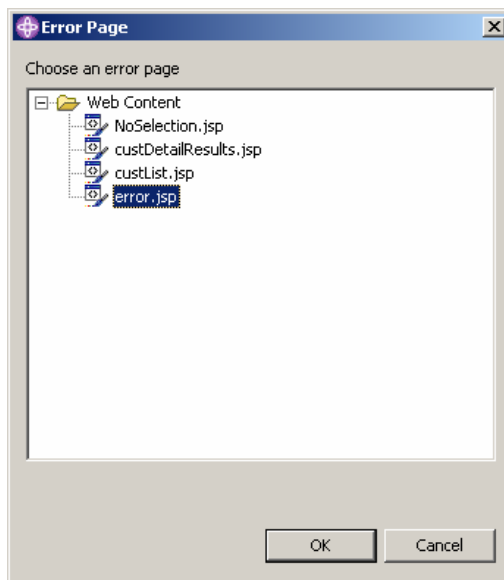
The screenshot shows a dialog box titled "Web Interaction" with a subtitle "Specify a Name and Location for your Web Interaction". Below the subtitle is a brief instruction: "Define the name and location for the communication between Web pages and the chosen business process. The location must be the Web Content directory inside a project." The dialog contains several input fields and checkboxes:

- Destination folder:** A text box containing "/wdscsubf/Web Content" with a "Browse..." button to its right.
- Web Interaction name:** A text box containing "custDetail".
- Java package:** An empty text box with a "Browse..." button to its right.
- Use error page:** A checked checkbox followed by a text box containing "error.jsp" and a "Browse..." button to its right.
- Invalidate session after the interaction occurs:** An unchecked checkbox.

At the bottom of the dialog, there are four buttons: "< Back", "Next >", "Finish", and "Cancel".

Figure 31: Web Interaction Wizard

- Type *custDetail* in the **Web Interaction name** entry field.
- Select the **Use error page** check box.



- From the *Choose an error page* dialog, select *error.jsp*.
- Click the **OK** push button
- Click the *Next* > push button.

Now we will define what our input and output pages will be in our Web Interaction. We have already created the input page (custList.jsp) in the previous exercise, so we need to declare that in the following page of the wizard. We will let the Interaction wizard generate the output page for us.

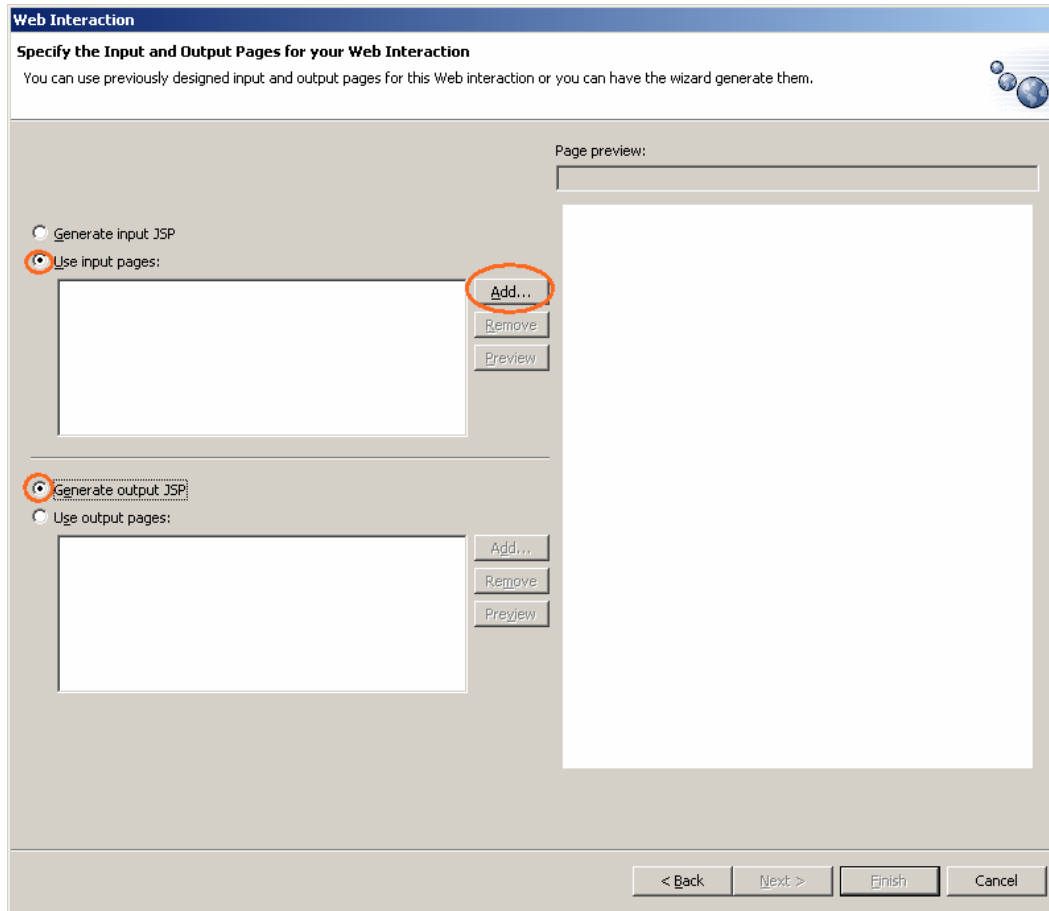


Figure 32: Defining input and output pages

- Select the **Use *input pages*** radio button.
 - Click the **Add...** push button in the input page section of the page
- The Input page dialog opens (Figure 33).

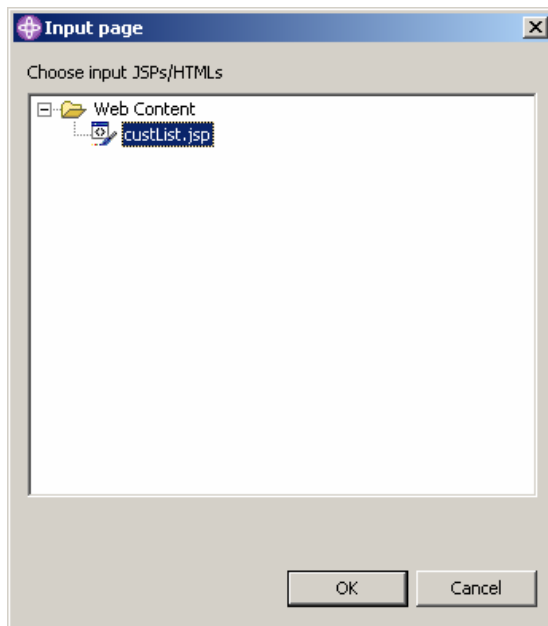


Figure 33: Selecting an input page

- Expand the **Web Content** list item by clicking on the + beside its icon.
- Select the JSP file **custList.jsp** so it is highlighted.
- Click the **OK** push button.

Your wizard page should now look like this:

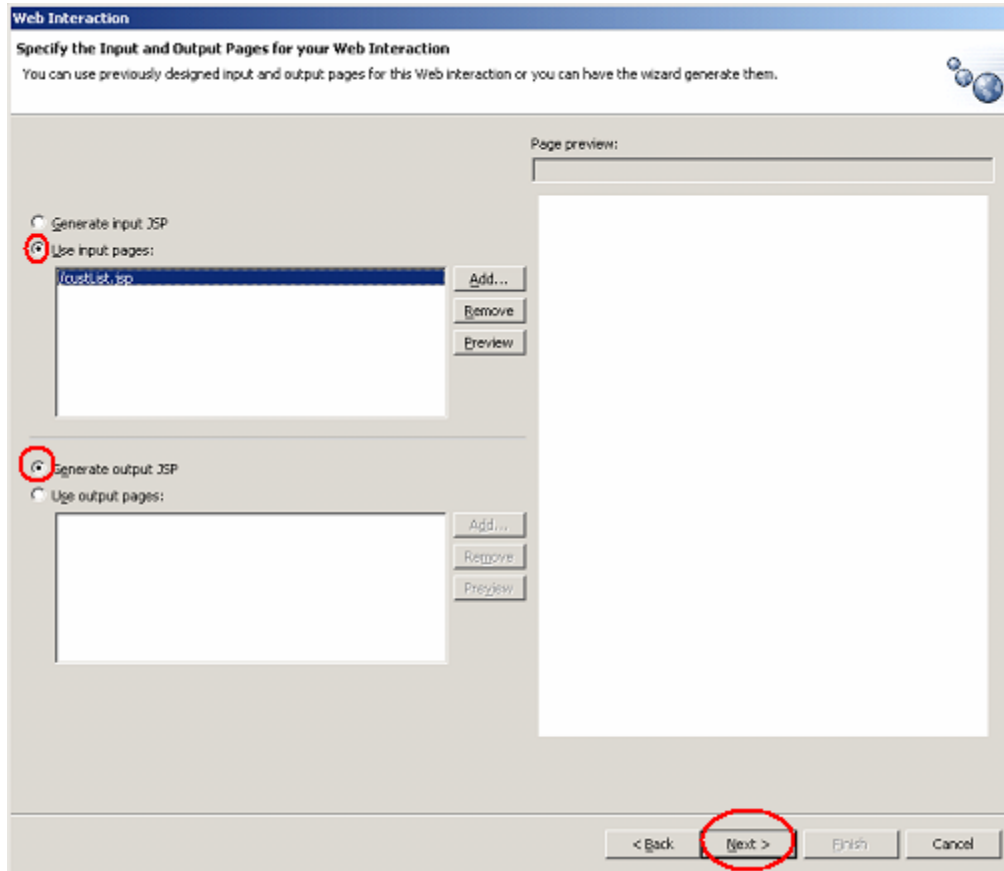


Figure 34: Defining input and output pages (continued)

- Select the **Generate output JSP** radio button
- Click the *Next* > push button.

We have now defined our input and output pages. Next we will select the RPG service program we will use to handle our Web interaction.

Adding an iSeries ILE Program to the Interaction

Web Interaction

Specify the Input and Output Parameters for your iSeries Host Program
Use this page to define the input and output parameters for your iSeries host program.

Use an iSeries ILE program Use a Java bean No program call

Program call definitions

Add Program... Add Parameter... Add Structure...

Add Program

Program alias:

Program object: **Browse...**

Library:

Program type: *PGM

Entry point:

CCSID of entry point:

Return type: void

Thread safe: False

Source location: **View...**

Associate this program with the interaction

Import PCML... Synchronize All

OK Cancel

< Back Next > Finish Cancel

Figure 35: Defining an iSeries ILE program

- Select the **Use an iSeries ILE program** radio button.
- Click the **Add Program...** push button.
- Click the **Browse...** push button to browse for the service program we will use.

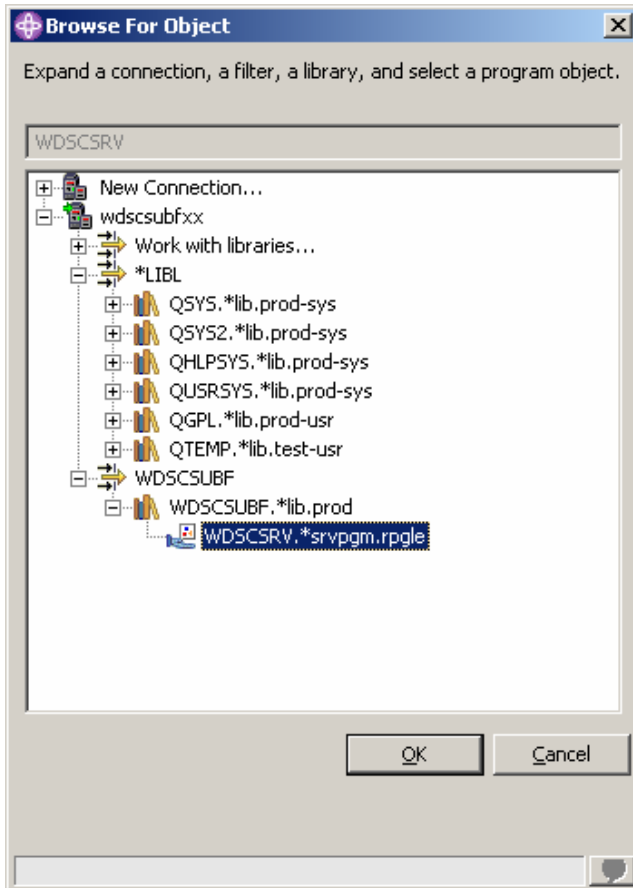


Figure 36: Selecting the RPG service program

- Expand the connection.
- Expand the **Wdscsubf** library as shown and select the **Wdscsrv** service program.
 - If you don't see the wdscsubf library, click on **Work with libraries...** and enter **Wdscsubf** in the dialog, then click the **OK** push button.
- Click the **OK** push button.
- Most of the fields have now been filled in for you. However you must fill in the following:

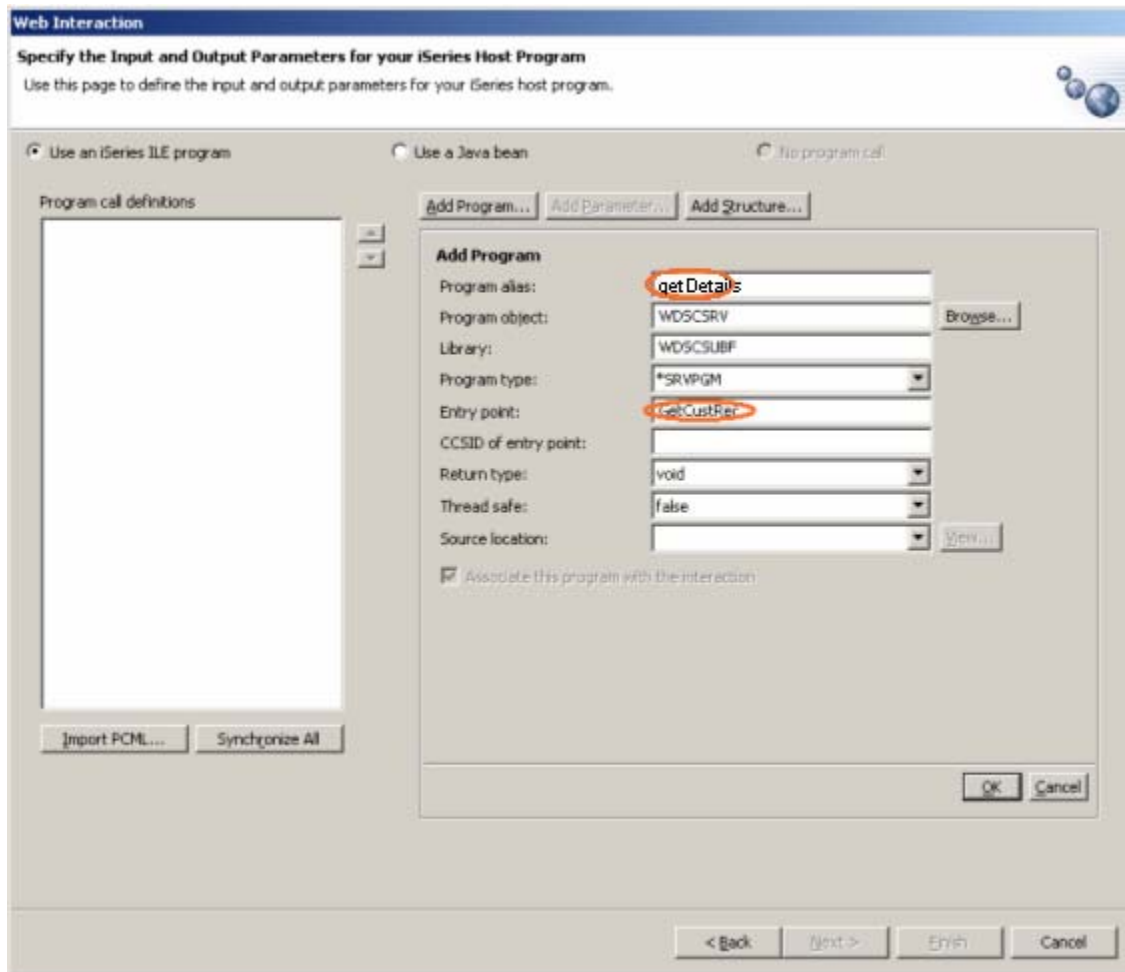


Figure 37: Filling in the Program parameters

- Type *getDetails* in the **Program alias** entry field.
- Type *GetCustRec* (case sensitive) in the **Entry point** entry field.
- Click the **OK** push button.

Defining a Structure

We need to define a structure to display the customer details in the output page.

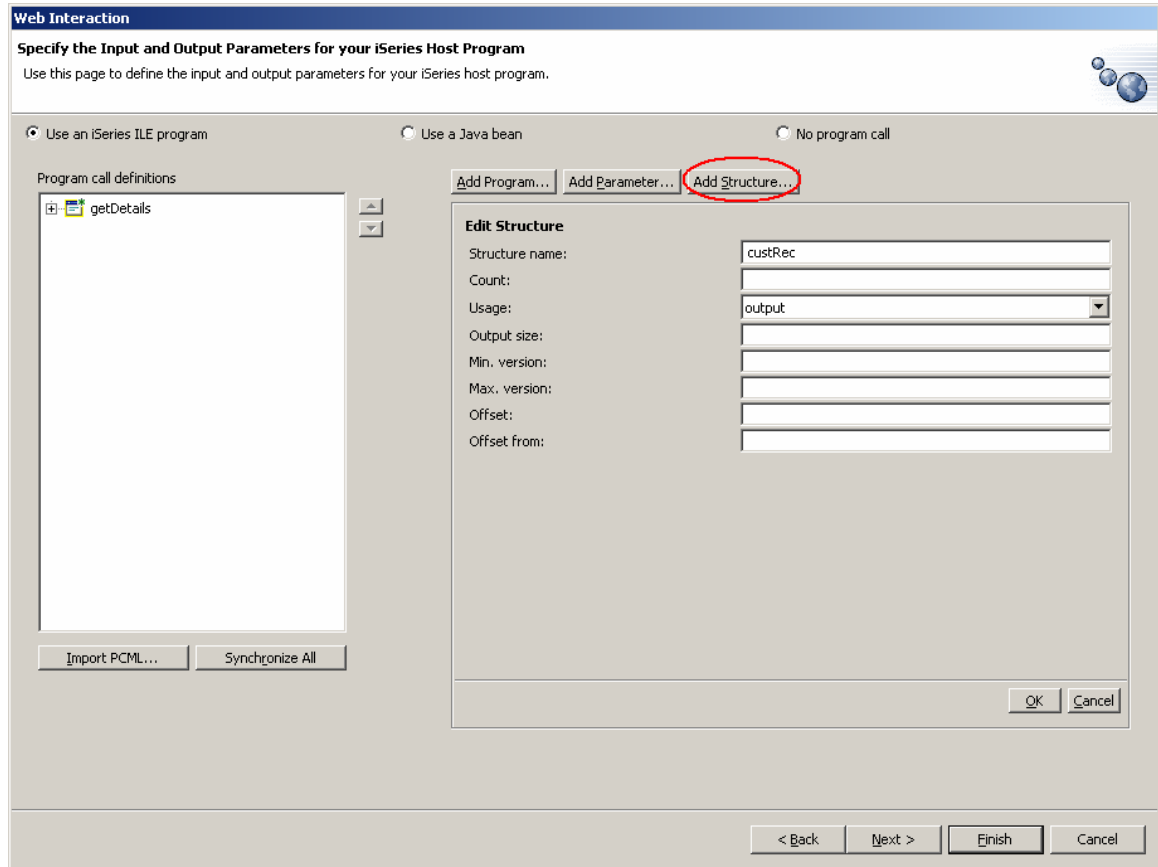


Figure 38: Adding a Structure

- Click the **Add Structure...** push button.
- Fill in **custRec** as the **Structure name**.
- Select **Output** as **Usage**.
- Click the **OK** push button.

Now we need to define this structure by adding parameters to it.

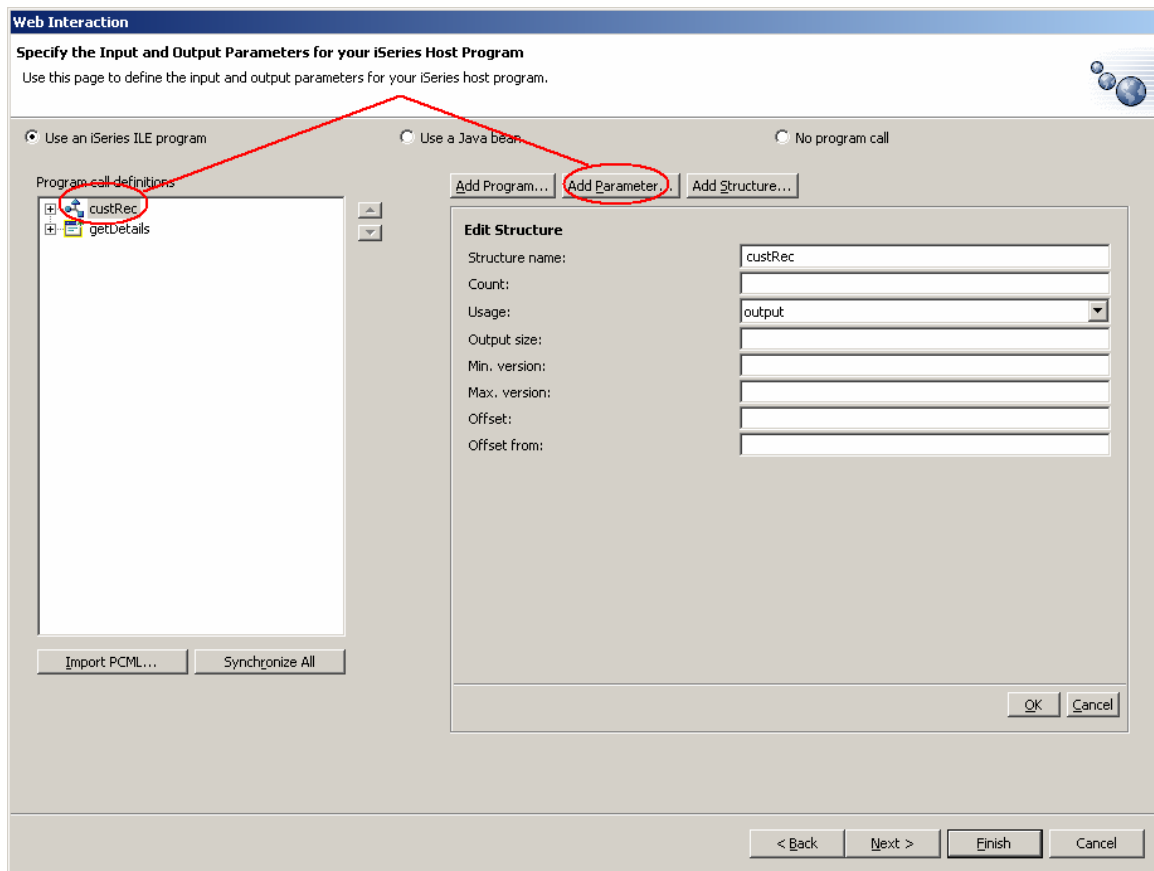


Figure 39: Selecting structure and clicking the Add Parameter... push button

- Select the newly created structure in the left, and then click the **Add Parameter...** push button.
- Click on the **Specify...** push button.

Now we will use the database to define the subfields for the structure.

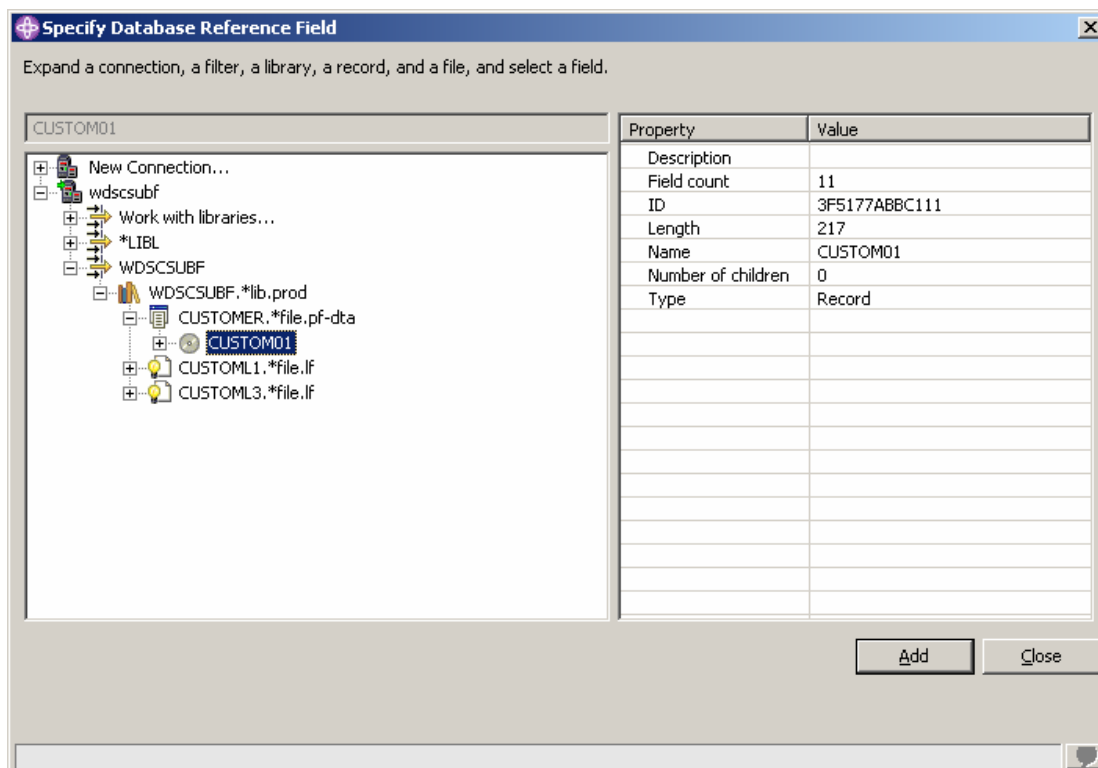


Figure 40: Defining a structure using the database

- Expand your iSeries connection and navigate to the *WDSUBF* library.
- Navigate as shown above, to the *CUSTOM01* record and select it.
- Click the *Add* push button.
- Now click the *Close* push button.

Adding a Structure as a Parameter

Let's add this structure as a parameter to our service program we added earlier.

- Select the *getDetails* program.
- Click the *Add Parameter...* push button.
- Select *structure* for the *Data type*
- Select our structure *custRec* as the *Structure name*
- Type *CustRec* as the *Parameter name* entry field.
- Select *Output* as the *Usage*.
- Click the *OK* push button

Your wizard should now look like this:

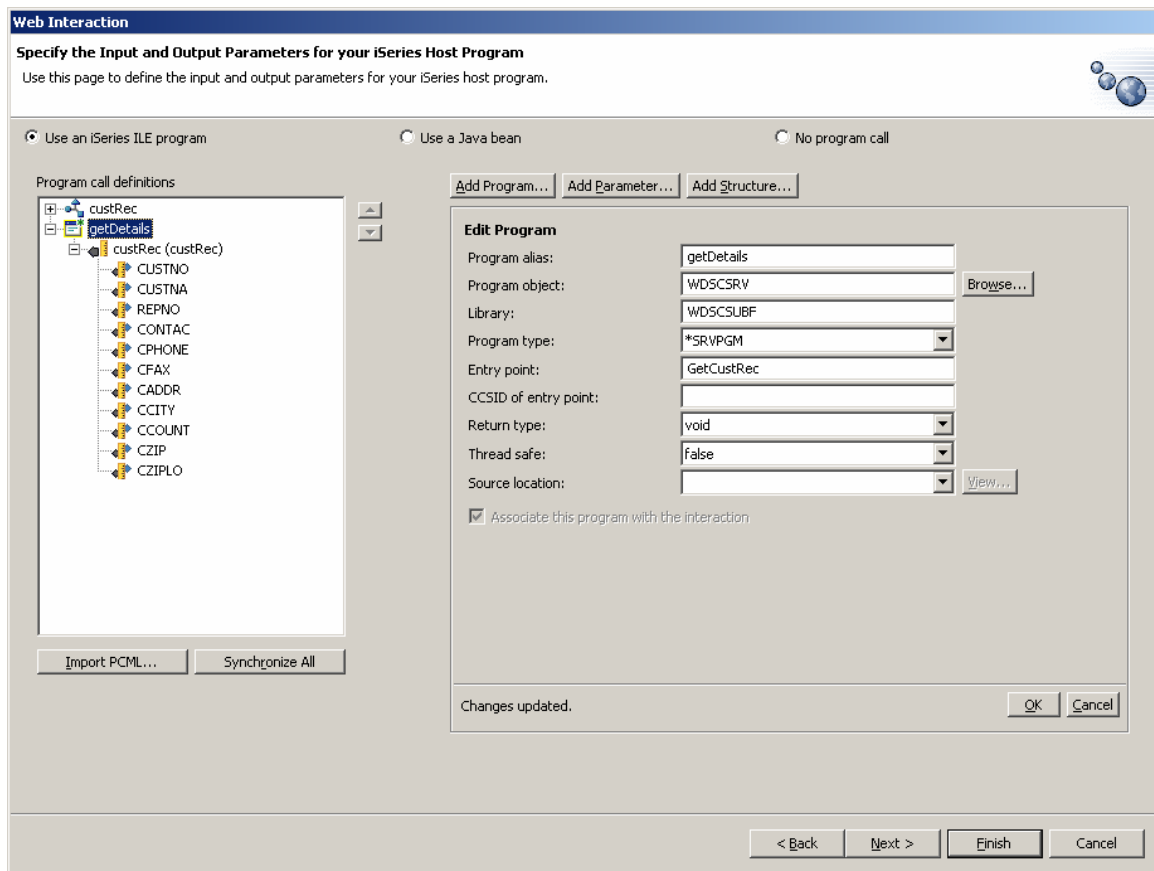


Figure 41: Adding the structure as a parameter to the service program

We are done specifying the program parameters. Now we need to setup the JSP's.

- Click the *Next >* push button.
- We have no input parameters, so we don't need to modify the Input JSP. Click the *Next >* push button again.

Modifying the Output Page in the Web Interaction Wizard

Now we are going to customize the output page as we don't want the page to display with generated field labels shown in Figure 42.

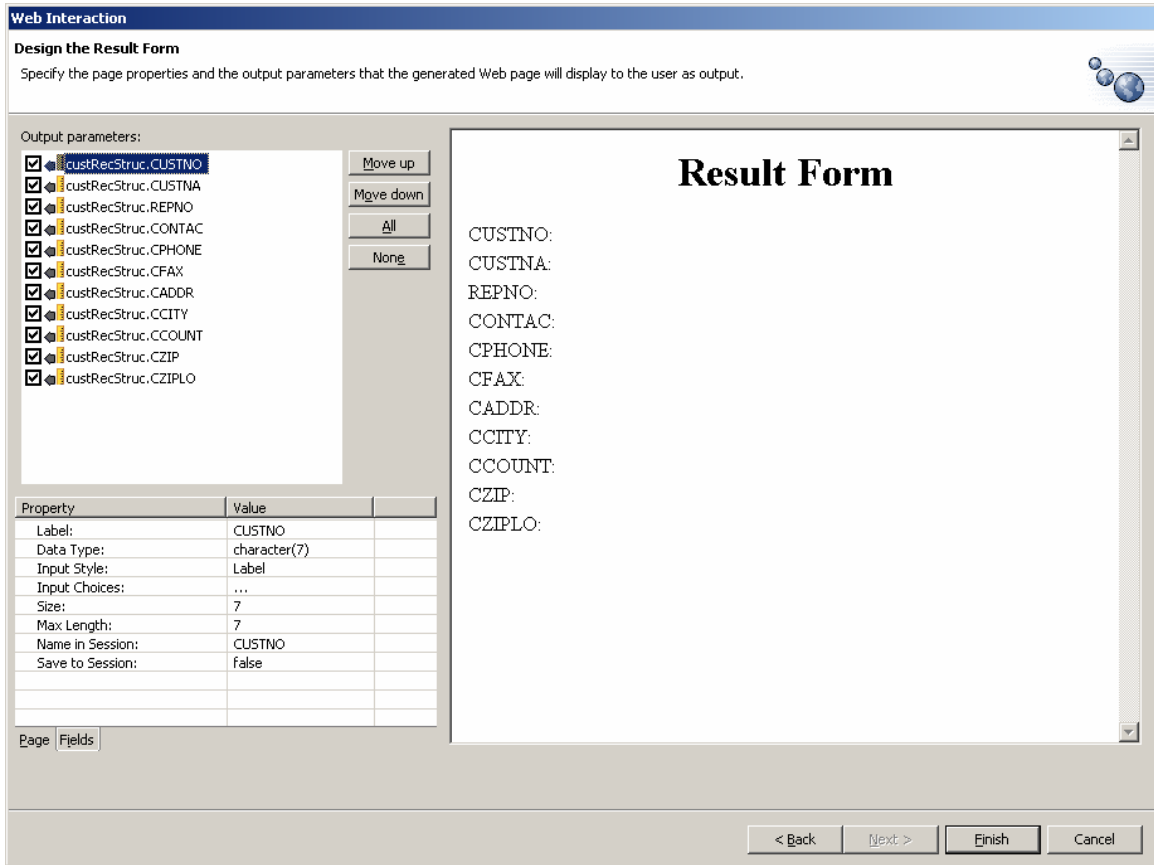


Figure 42: Default result page

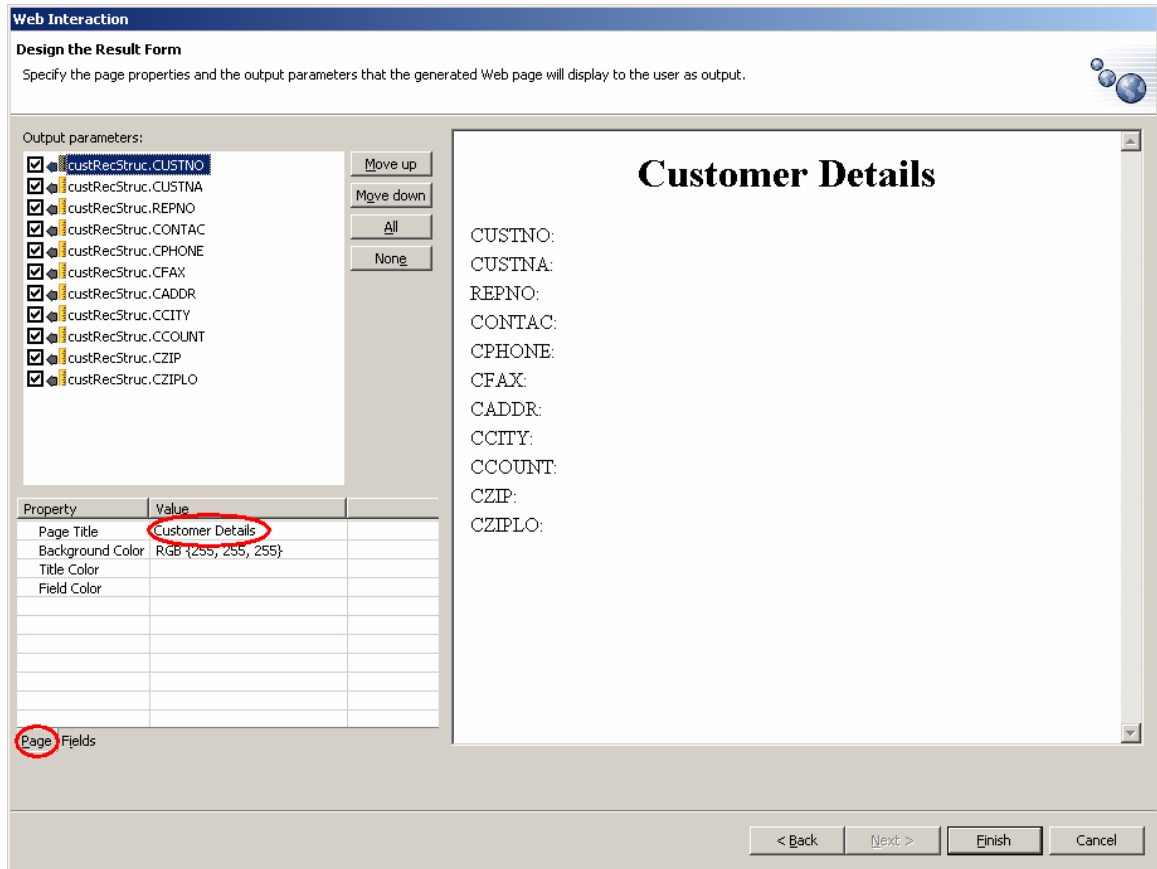


Figure 43: Modifying the output page in the Web Interaction Wizard

- Change the title of the page by clicking the **Page** tab at the bottom left then changing the **Page Title** field to **Customer Details**.

We need to modify the labels of the fields that show up on the output JSP.

- Now click the **Fields** tab.
- Select each field in the top left, and for each one, change the **Label** value to the values described in the table below.

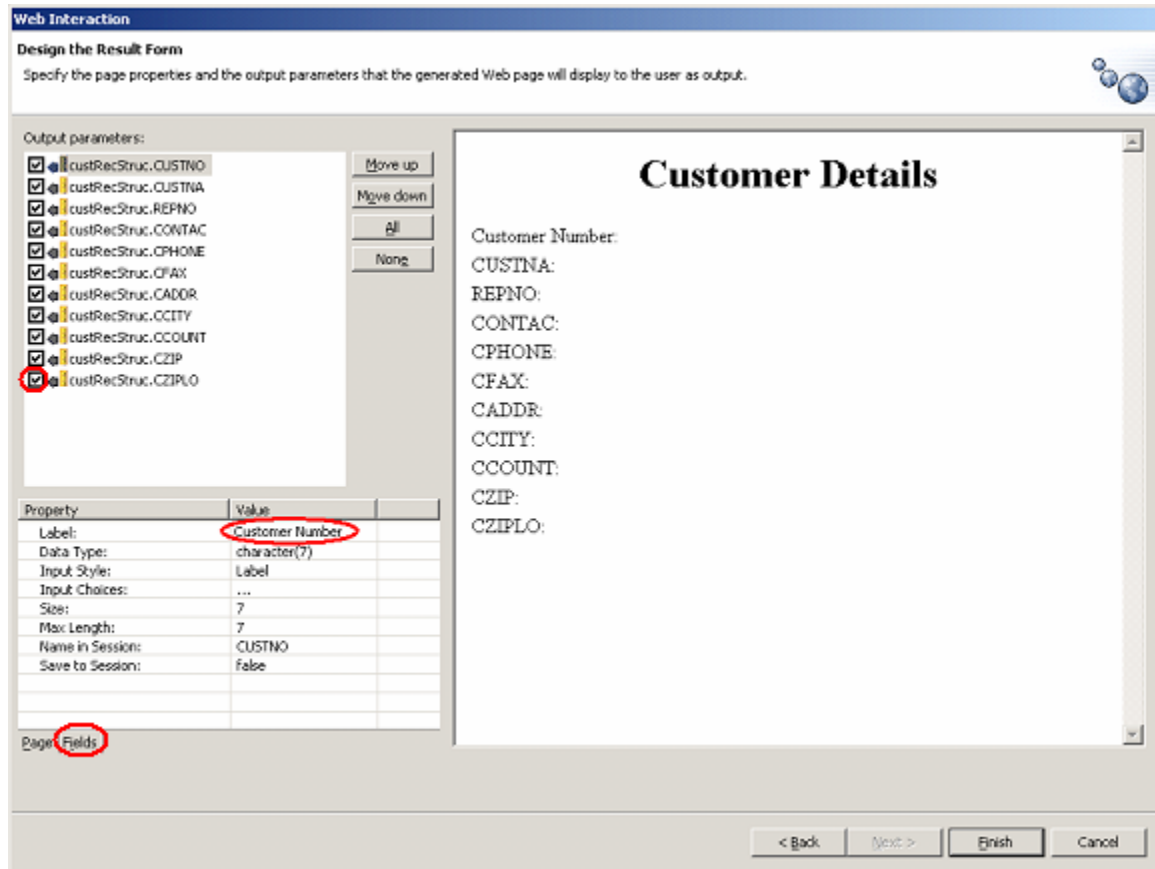


Figure 44: Changing the parameter labels

CUSTNO	Customer Number
CUSTNA	Customer Name
REPNO	Representative Number
CONTAC	Contact
CPHONE	Phone Number
CFAX	Fax Number
CADDR	Address

CCITY	City
CCOUNT	Country
CZIP	Zip

- Uncheck the **CZIPLO** data field in the left, we don't need to display that (Figure 44).
- Click the **Finish** push button.

Customizing the Output Page with Page Designer

Let's use the Page Designer to customize our output page further.

- In the **Web Perspective**, find the **custDetailResults.jsp** page under the **Web Content** folder in your project.
- Double click on the file. This will automatically open it with the **Page Designer**.

First, let's add a link that will take the user back to the **Customer List**. Since we won't be changing the data in the **Customer Details** page, we can delete the **Submit** and **Reset** buttons.

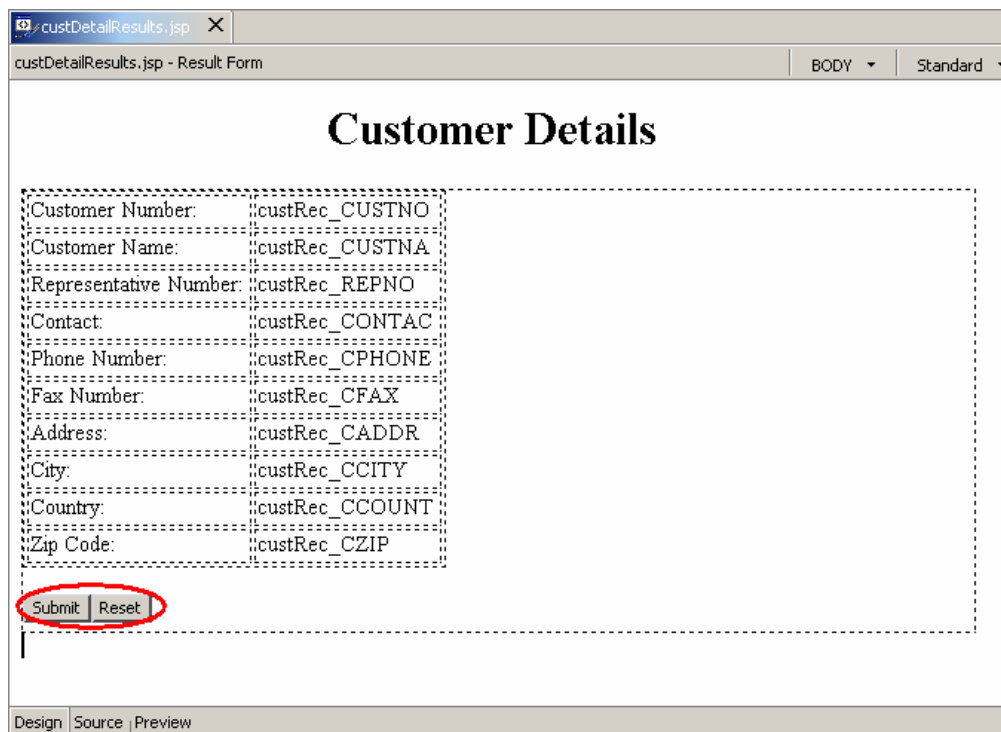


Figure 45: Customizing the output page using Page Designer

- Select each of the buttons and press the *Delete* key on the keyboard.
- Type the text **Back to Customer List** and select it.

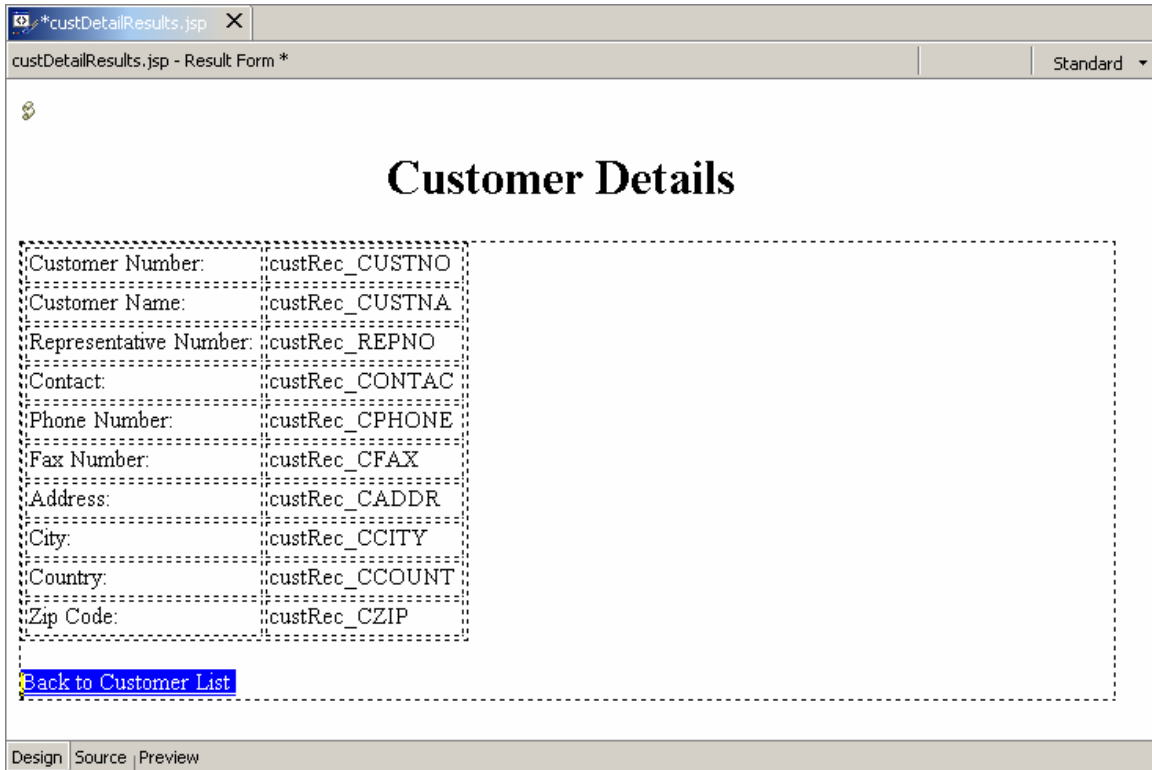


Figure 46: Creating a Link

- Now right mouse click on the selected text and select *Insert Link...* as shown below



Figure 47: Inserting a Link

- Type *custList.jsp* in the *URL* entry field in the dialog as shown below

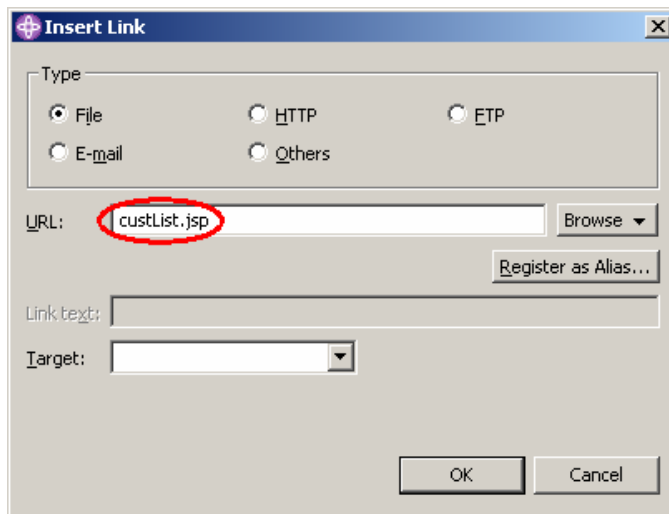


Figure 48: Defining the URL for the link

- Click the **OK** push button.

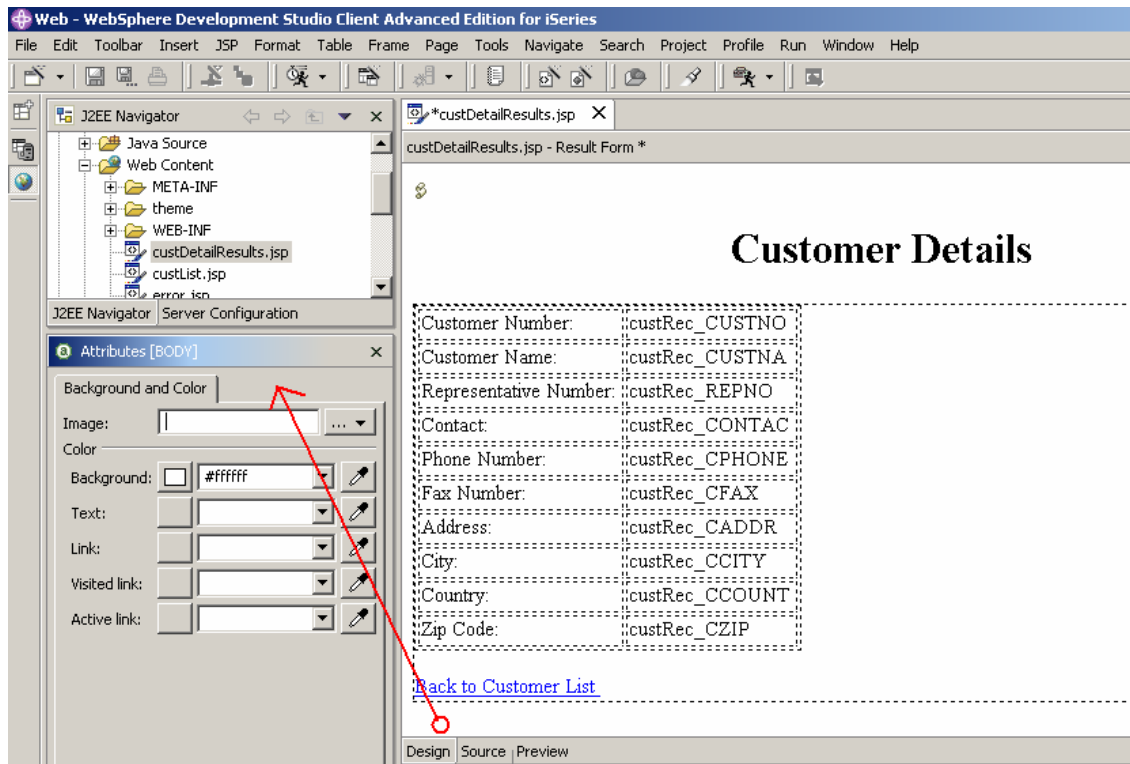


Figure 49: Selecting page colors

- Select a blank area on the page, outside the dotted rectangle. The page attributes will appear in the Attributes view.
- Choose any color you wish, or leave it as it is.
- Save the file.

You are now ready to run the Web application.

Running the Web Application

Now you are going to run your Web application in the WebSphere version 5.0 Test Environment. This test environment allows us to test Web Applications with iSeries machines without actually installing WebSphere Application Server version 5.0 on a server.

- In the *Web Perspective*, right mouse click on *custList.jsp* and select *Run on Server...* from the pop-up menu.

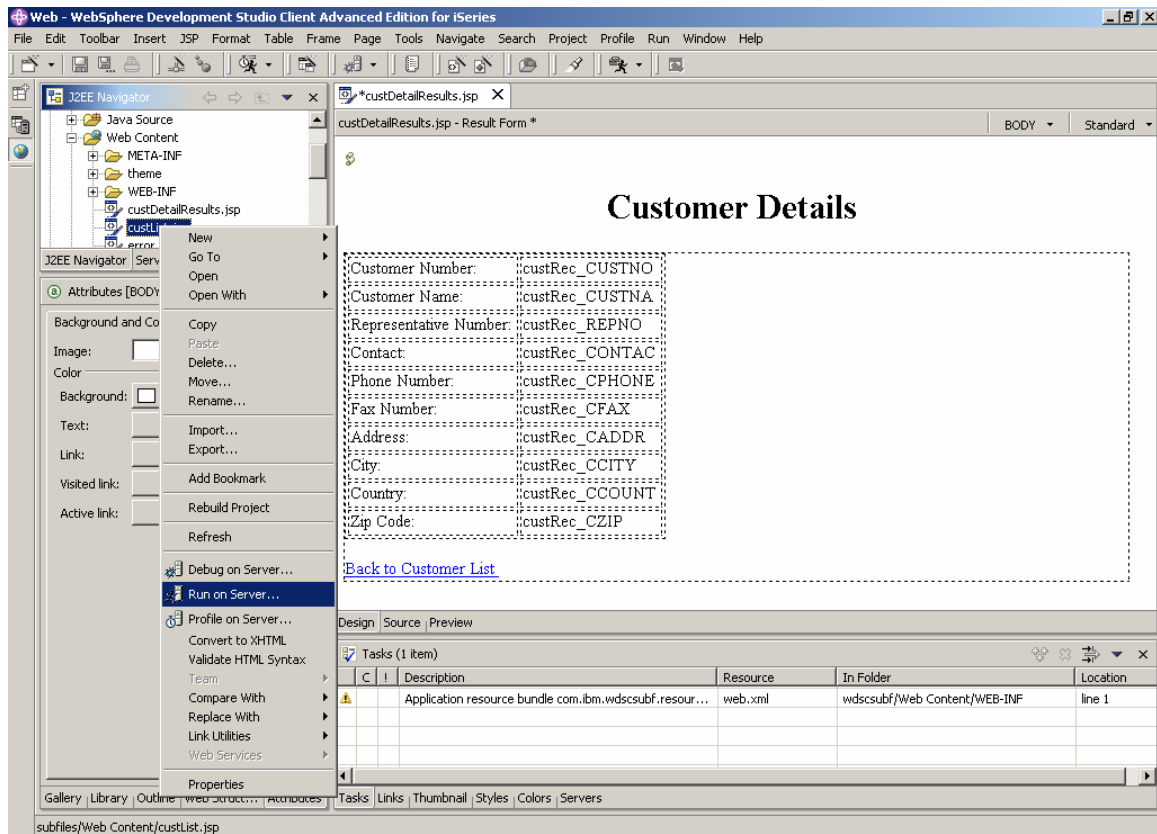


Figure 50: Running the Application on the Server

- You will get a message to select the Server to run in the WebSphere Test Environment (Figure 59):

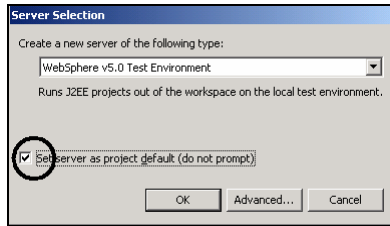


Figure 51: Server selection

Note: This only happens if you have no Servers configured. If a server has been configured you will get a different prompt to select the server to use.

- Make sure the WebSphere V5.0 Test Environment is selected in the **Create new server of the following type** entry field
- Select the **Set server as project default (do not prompt)** check box
- Click the **OK** push button. Starting the server will take a few moments.

A Web browser will open and display the **custList.jsp** file.

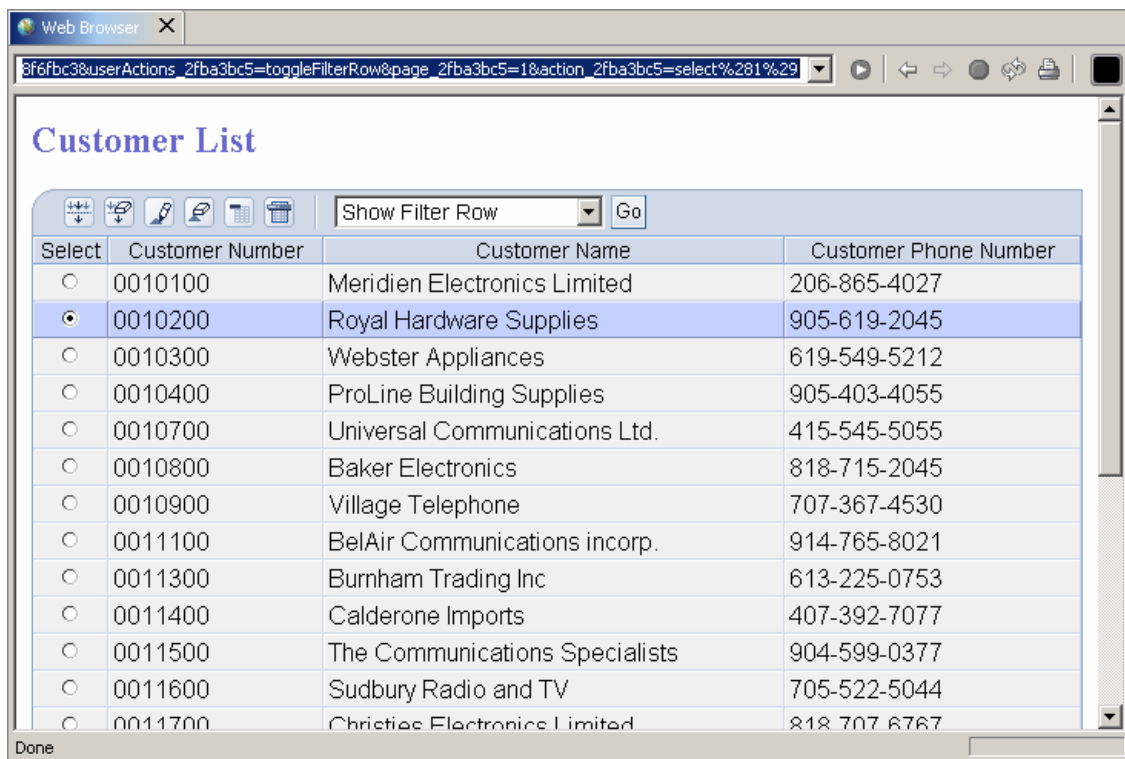


Figure 52: The input page

- Test the application by selecting a customer and clicking the *Get Customer Details* button. You should now see the output page with the details.

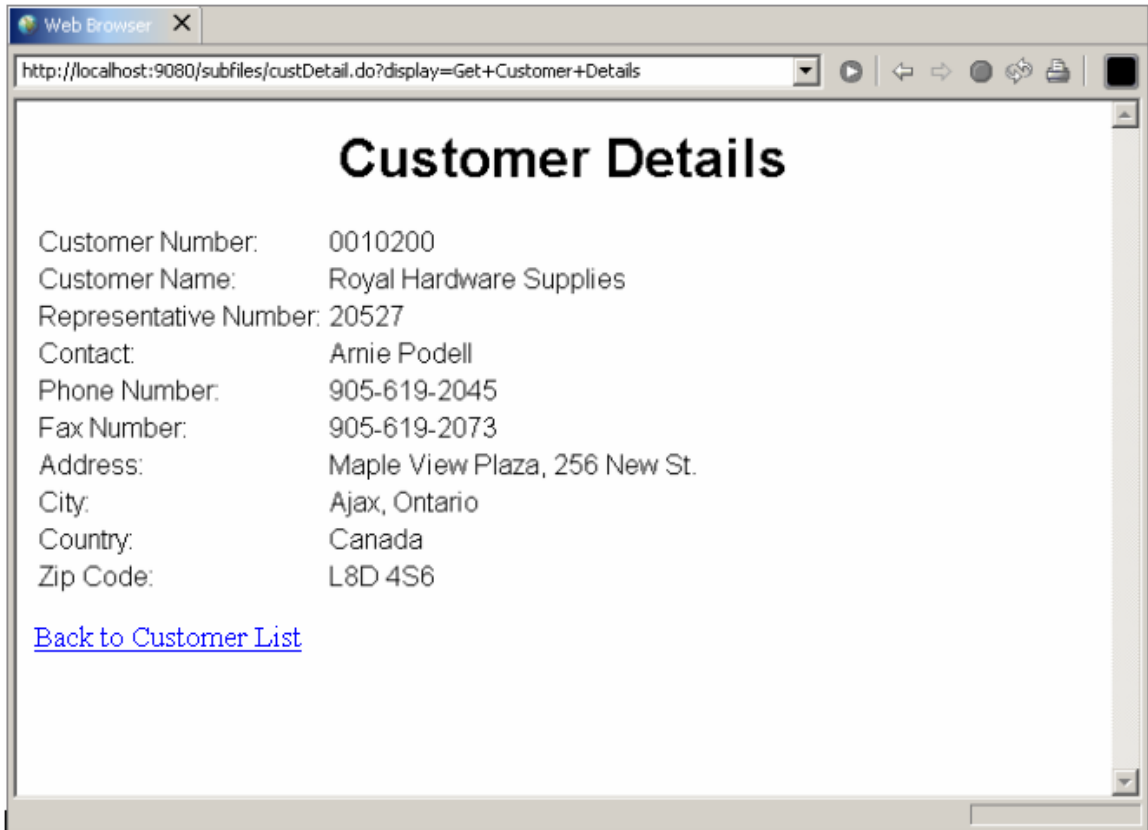


Figure 53: The output page

Now we know that our Web customer details application works!

WebSphere Development Studio Client for iSeries – Web Tools Demo#2 Subfiles Summary

See how easy it is to create a Web browser version of a subfile in order to display a list of customers and then display the details for each selected customer.

This concludes this demo. We hope you **enjoyed** the experience. For more details, tips and techniques, and service and support information, be sure to visit the Development Studio Client Web site at

www.ibm.com/software/awdtools/series

Have fun with Development Studio Client for iSeries!