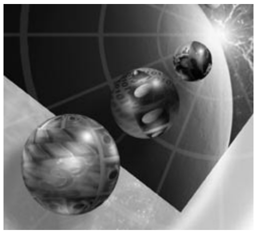


IBM WebSphere Development Studio client for iSeries

Intro to Web tools and iSeries
Building an *e*-business application with RPG using WDSc for iSeries

Student Exercises

Course code

WDSc V4 SP2

Claus Weiss
weiss@ca.ibm.com

IBM Toronto Laboratory

Version 4 SP 2

homepage: http://www.ibm.com/software/ad/iseries
newsgroup:
news://news.software.ibm.com/ibm.software/websphere/studio400

Fourth Edition (May, 2002)

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Comments concerning this notebook and its usefulness for its intended purpose are welcome. You may send written comments to:

IBM Canada Software Solutions

8200 Warden Ave. Markham, Ontario, L6G 1C7

Attention: Claus Weiss, WDSc Web tools for iSeries Introduction.

or e-mail to**: weiss@ca.ibm.com**

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 with IBM Corp**.**

Overall Lab Guide

The objective of the WDSc for iSeries lab is to have the students work with the Webtools in the WDSc workbench to create, build and execute an *e*-business application.

At the end of the lab, the student should know how to create a simple *e*-business application by creating a browser user interface, using the iSeries Wizards in WebSphere Developement Studio Client for iSeries to create the Servlets to connect to an RPG program that performs the business logic.

The Lab also shows how to run the applications in the WAS test environment that is part of WDSc.

All the exercises require that the previous labs are successfully completed.

.

Note:

The pictures in these labs show a similar application being built. Some of the names and icons may be different from the environment you are working with.

Prerequisites

Although not required, familiarity with the RPG language is preferred.

Also, it is helpful if the student is familiar with basic MS Windows operations such as working with the desktop and basic mouse operations such as opening folders and performing drag-and-drop operations. The student should also be familiar with a browser and know how to navigate through the Internet.

Trademarks

IBM is a registered trademark of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation.

AS/400
Application System/400
VisualAge
DB2/400
iSeries
ILE
Integrated Language Environment
IBM
OS/400
RPG/400
VisualAge
WebSphere

Trademarks of other companies as shown

'Lotus'	'Lotus Development Corporation'
'Microsoft'	'Microsoft Corporation'
'Windows'	'Microsoft Corporation'

Create a Simple HTML Page with WebSphere Studio

Exploring WebSphere Studio for iSeries

During this lab you will learn more about ***WebSphere Development Studio client (WDSc)*** for iSeries. In doing so you will be performing the following:

1. Explore WDSc for iSeries by:

   - Starting **WebSphere Development Studio client for iSeries**
   - Getting familiar with the **page designer** tool
   - Using parts and **page designer components** to create a simple browser interface

2. Test the files by:

   - Using the **WebSphere Application Server (WAS)** test environment
   - Invoking the built-in workbench browser and showing your web page

What You Should be Able to Do

As a result of this exercise you will create the first page of your *e*-business application. In doing so you will:

- Create a simple Java Server Page (JSP) to be used as the user interface in your application
- Create dynamic images on a web page
- Create a LOGO on your web page
- Use a Layout Frame to put moving pictures on your web page
- Use **Design Time Control's (DTC's)** to add capability to send data to an iSeries program
- Test the user interface

Starting WebSphere Development Studio client

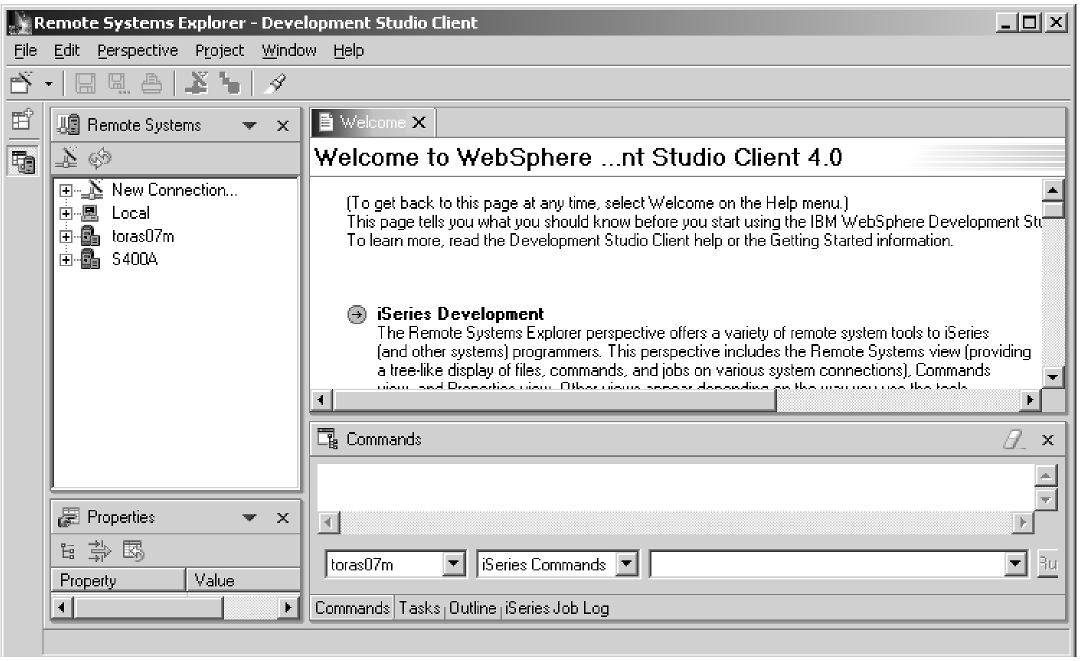
Before you start using WebSphere Development Studio Client, you will need to reset it. This will delete any changes to the environment due to previous use of the tool on the PC you are working on.
The instructor will tell you how to reset the WebSphere Development Studio Client environment by removing the **WORKSPACE** folder from the **WSSD** folder.

> **Note:**
> If you work on your own system be careful this will remove all project information stored in the workbench you will loose work if you reset your workbench.

After you have reset the workbench go ahead and start WDSc
To start the Development Studio for iSeries:

- Press the **Start** button on the task bar of your Windows desktop
- Choose ***Start→Programs→IBM WebSphere Development Studio Client for iSeries→ IBM WebSphere Studio Site Developer Advanced***

After a moment the workbench will appear on your desktop


Remote Systems Explorer - Development Studio Client
File Edit Perspective Project Window Help
Remote Systems
New Connection...
Local
toras07m
S400A
Properties
Property
Value
Welcome
Commands
toras07m
iSeries Commands
Commands Tasks Outline iSeries Job Log

Welcome to WebSphere ...nt Studio Client 4.0

(To get back to this page at any time, select Welcome on the Help menu.)
This page tells you what you should know before you start using the IBM WebSphere Development Stu
To learn more, read the Development Studio Client help or the Getting Started information.

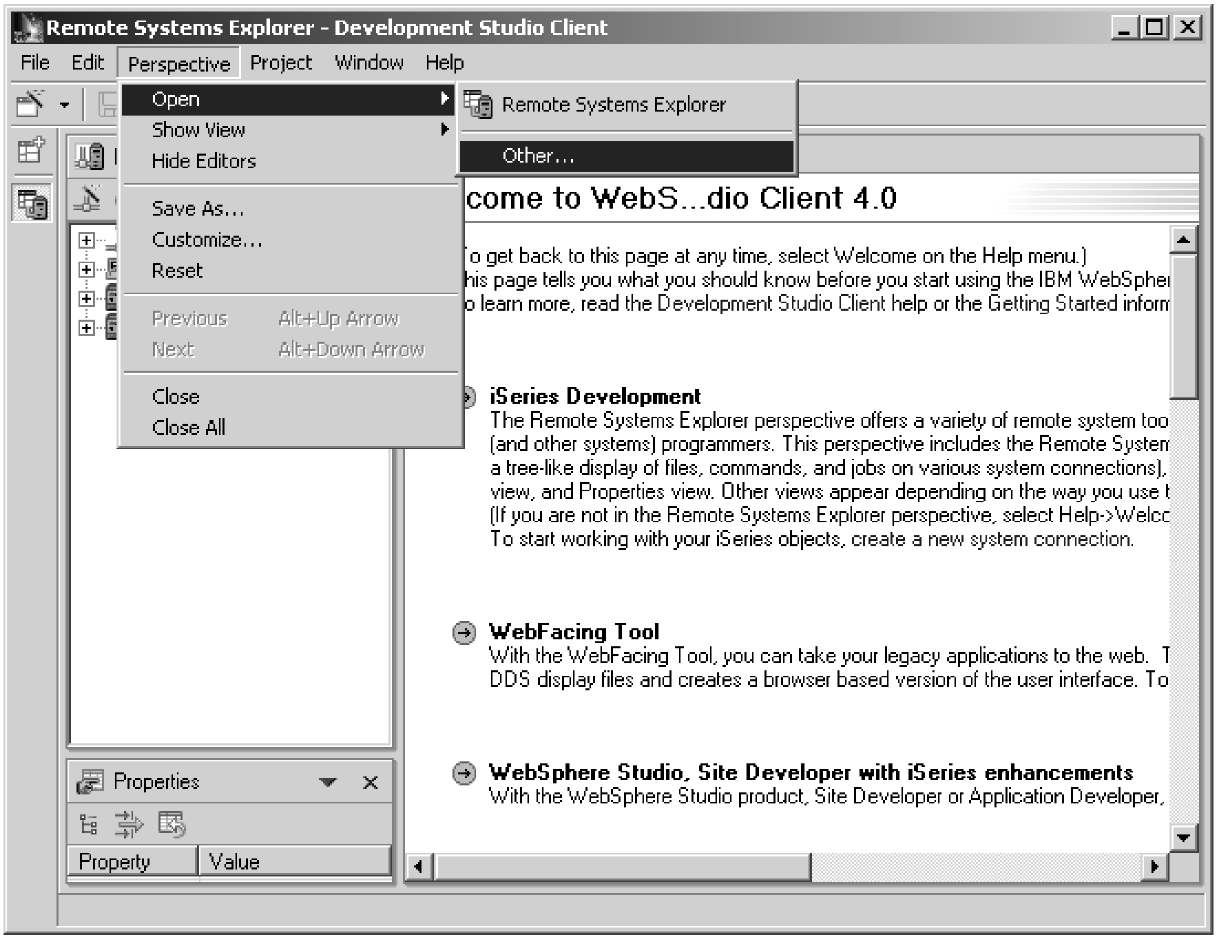
iSeries Development
The Remote Systems Explorer perspective offers a variety of remote system tools to iSeries (and other systems) programmers. This perspective includes the Remote Systems view (providing a tree-like display of files, commands, and jobs on various system connections), Commands

If the workbench has previously been used, it will look different than in the figure above. In case you didn't reset the workbench some pictures will differ from your real lab environment

Selecting the Web perspective

We want you to work with web tools, so the first step for you will be to get the perspective for a web developer started.

On the workbench menu bar



come to WebS...dio Client 4.0

o get back to this page at any time, select Welcome on the Help menu.)
his page tells you what you should know before you start using the IBM WebSpher
o learn more, read the Development Studio Client help or the Getting Started inforn

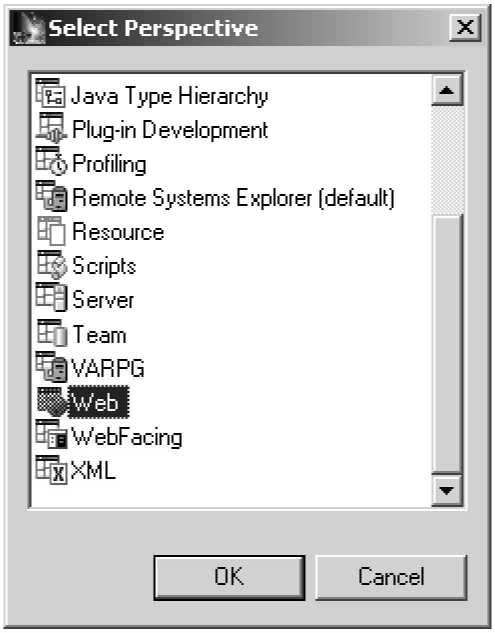
iSeries Development
The Remote Systems Explorer perspective offers a variety of remote system too
(and other systems) programmers. This perspective includes the Remote Systen
a tree-like display of files, commands, and jobs on various system connections),
view, and Properties view. Other views appear depending on the way you use t
(If you are not in the Remote Systems Explorer perspective, select Help->Welco
To start working with your iSeries objects, create a new system connection.

WebFacing Tool
With the WebFacing Tool, you can take your legacy applications to the web. T
DDS display files and creates a browser based version of the user interface. To

WebSphere Studio, Site Developer with iSeries enhancements
With the WebSphere Studio product, Site Developer or Application Developer,

- Select the **Perspective** menu option
- Select **Open** from the sub menu
- Select **Other...** from the next sub menu .

The Select Perspective dialog will appear



- Select **Web** from the list of available perspectives
- Press the **OK** push button on this dialog

What is a perspective

Before we go further, lets discuss perspectives. Perspectives are a collection of views and tools assembled to help different kind of users to do their job when they use the workbench.

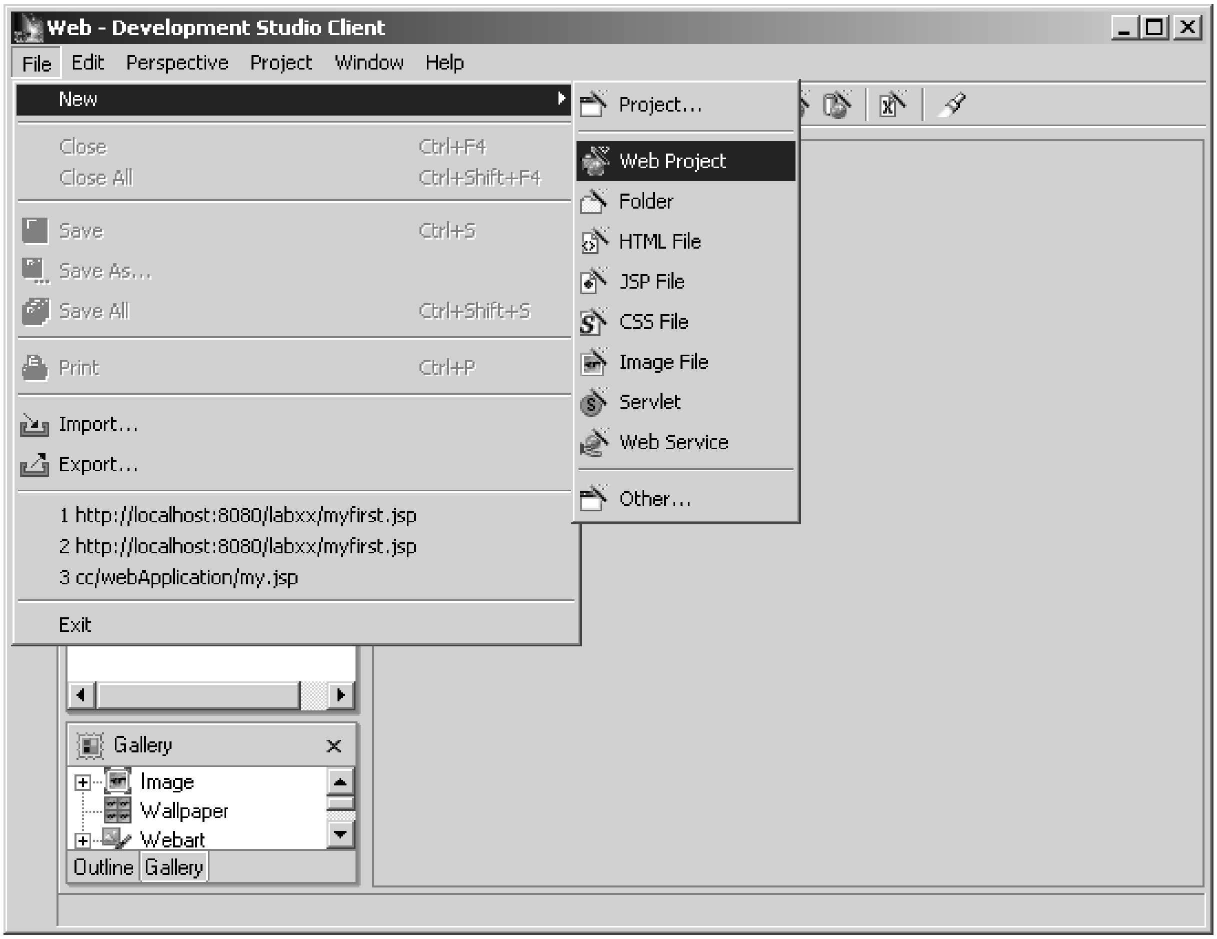
These workbench users will have different jobs (roles), one needs to work with iSeries objects related to RPG/COBOL, another one is dealing with JAVA programs, and another with Web page development. Each of these different user types will need different views into the files/objects they are working and they need different tools.

The perspectives present the user with a selection of specific views/tools geared towards the different roles the users have. The workbench has many predefined perspectives, like the **Web** perspective for web developers, or the **Remote System Explorer** perspective for iSeries programmers. You can also create your own perspective by using the **Save as** option under the **Perspective** menu in the workbench.

Now that you know what a perspective is we can go ahead in our exercise, you are a web developer in this hands on lab and that is the reason you selected the **Web** perspective.

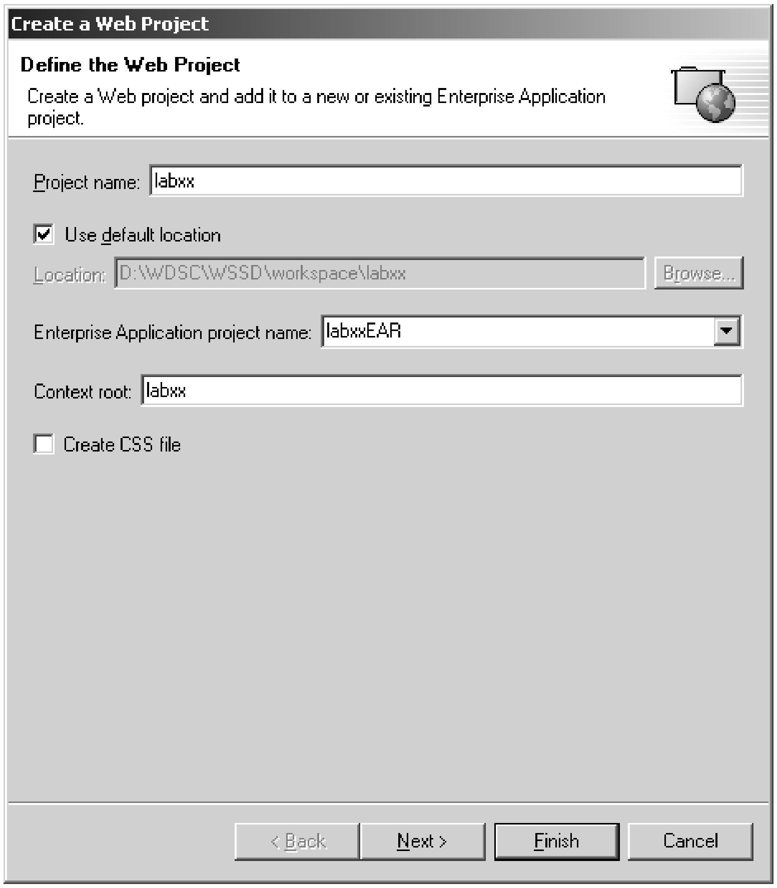
Create a Web Project

Lets create a new Web project



- Select the **File** menu item on the workbench menu bar
- Select **New** from the pull down menu
- Select **Web Project** from the sub menu

At this point the **Create a Web project** dialog appears.



Create a Web Project

Define the Web Project

Create a Web project and add it to a new or existing Enterprise Application project.

Project name: labxx

☑ Use default location

Location: D:\WDSC\WSSD\workspace\labxx

Enterprise Application project name: labxxEAR

Context root: labxx

□ Create CSS file

This dialog prompts you to provide more information about your first web project.

- Specify project name **Labxx**, where **xx** denotes your team group number:

for example, if your are in group 40 use: Lab40

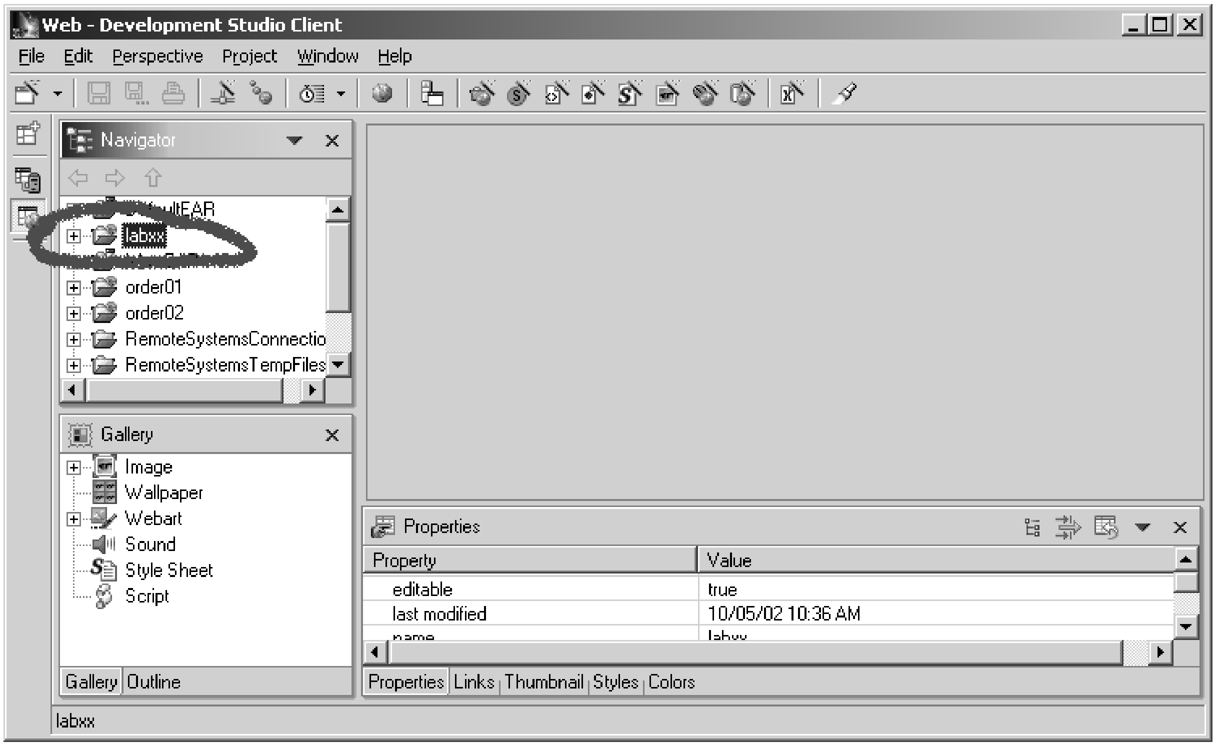
Note: **If you don't have a group number, please ask your instructor**

- Also specify the Enterprise Application project name **labxxEAR**

  Again **xx** being your group number
- Leave the other **defaults** as they are
- Press the **Finish** push button

  .

Now you are back in the **workbench** and your new project shows in the Navigator view.



Property	Value
editable	true
last modified	10/05/02 10:36 AM
name	labxx

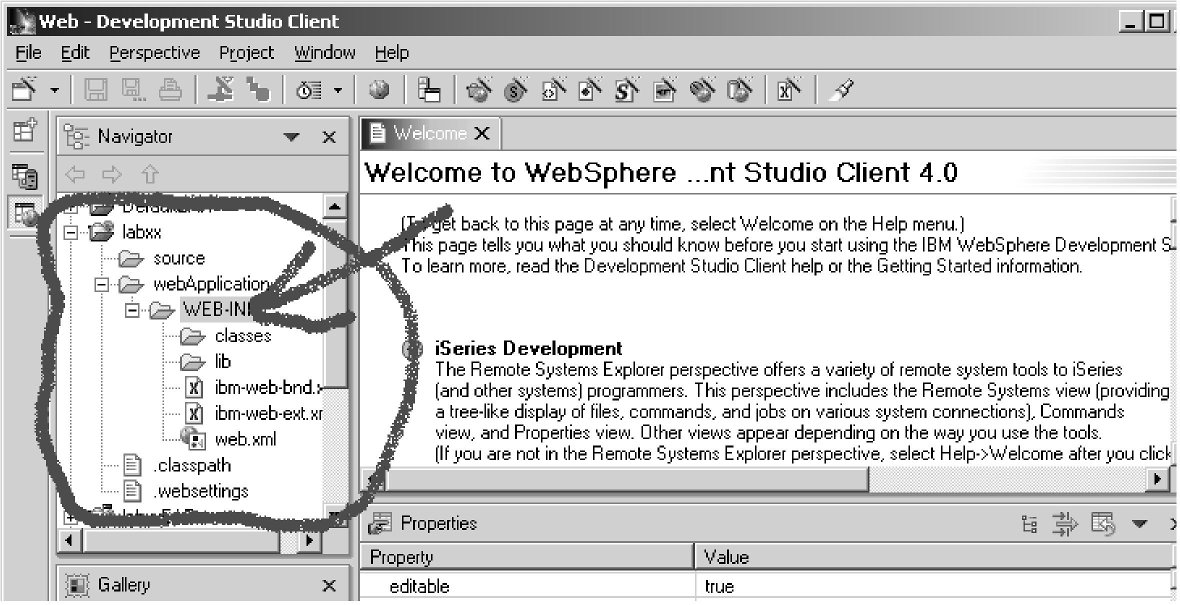
You are now ready to build your first web project.

Using the Page Designer tool

We want you to create your first web page. The tool you'll be using is called **Page Designer** and it comes with **WDSc**. It is like **SDA** for 5250 panels or a **GUI builder** for GUI windows, but as its name indicates it helps with designing web pages.

As in **SDA** or **CODE designer**, instead of keying in source statements in an editor, **page designer** will create the source for you from your design choices. Here, the source generated is **html or jsp** tags, instead of the **DDS** statements that would be created for a display file by **SDA**.

To invoke **page designer** do the following:



Welcome to WebSphere ...nt Studio Client 4.0

(To get back to this page at any time, select Welcome on the Help menu.)
This page tells you what you should know before you start using the IBM WebSphere Development S
To learn more, read the Development Studio Client help or the Getting Started information.

iSeries Development
The Remote Systems Explorer perspective offers a variety of remote system tools to iSeries (and other systems) programmers. This perspective includes the Remote Systems view (providing a tree-like display of files, commands, and jobs on various system connections), Commands view, and Properties view. Other views appear depending on the way you use the tools. (If you are not in the Remote Systems Explorer perspective, select Help->Welcome after you click

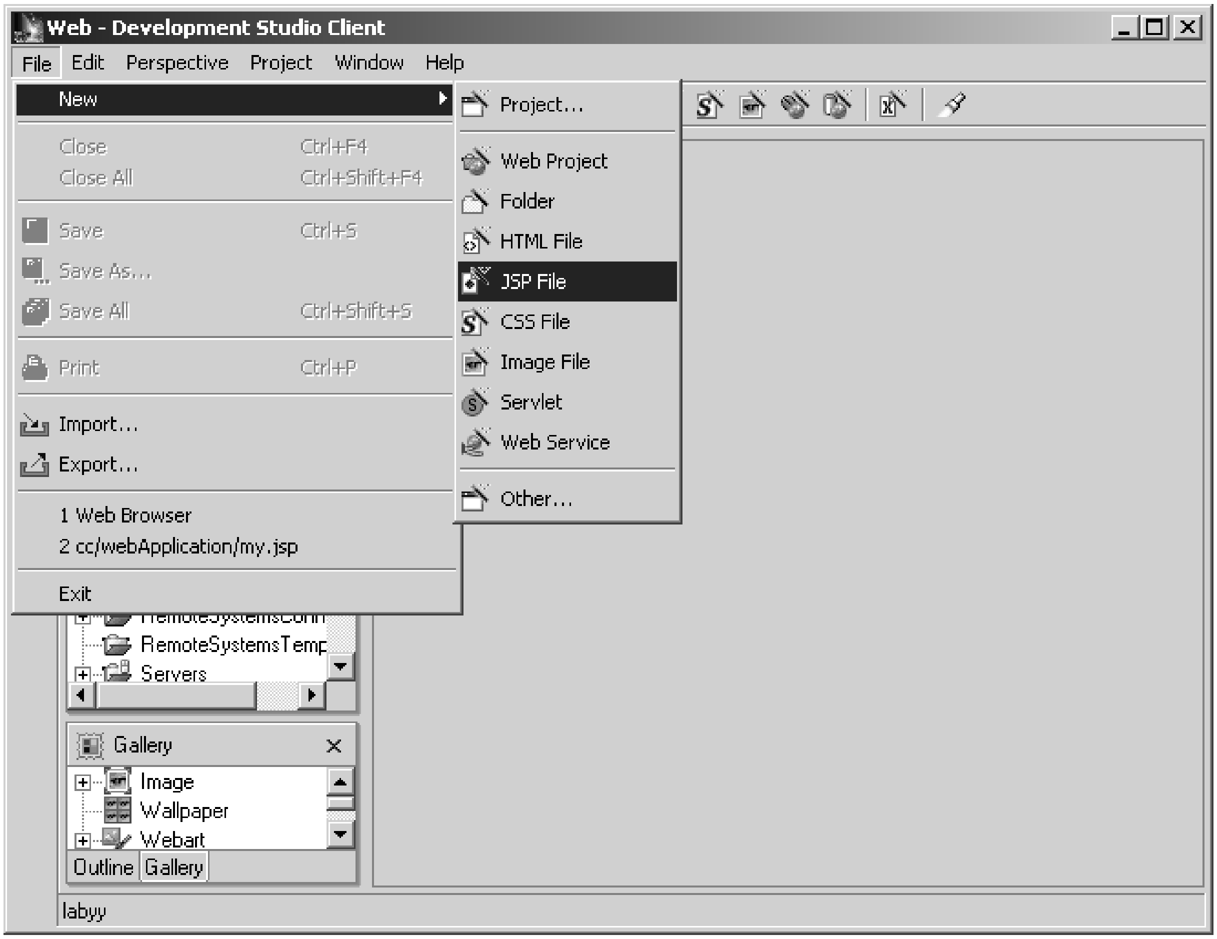
Property	Value
editable	true

- Locate your project in the workbench Navigator list.
- Expand the project by clicking on the ✚ sign beside the project icon

You see the predefined web project structure

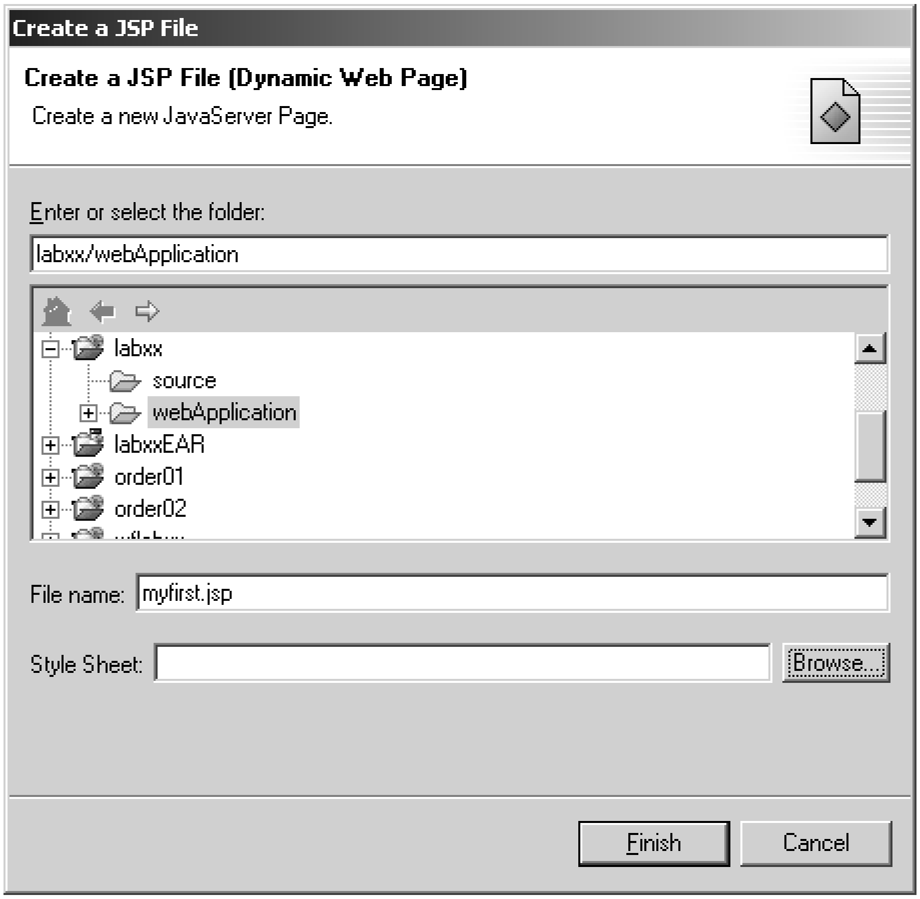
Notice that there is a **webApplication** folder, that will contain your web related files

In the workbench:



- Select the **File** menu item in Workbench menu
- Select **New** from pull down menu
- Select **JSP File** from the sub menu

The **Create JSP file** dialog will appear:



Create a JSP File

Create a JSP File (Dynamic Web Page)

Create a new JavaServer Page.

Enter or select the folder: labxx/webApplication

File name: myfirst.jsp

Style Sheet:

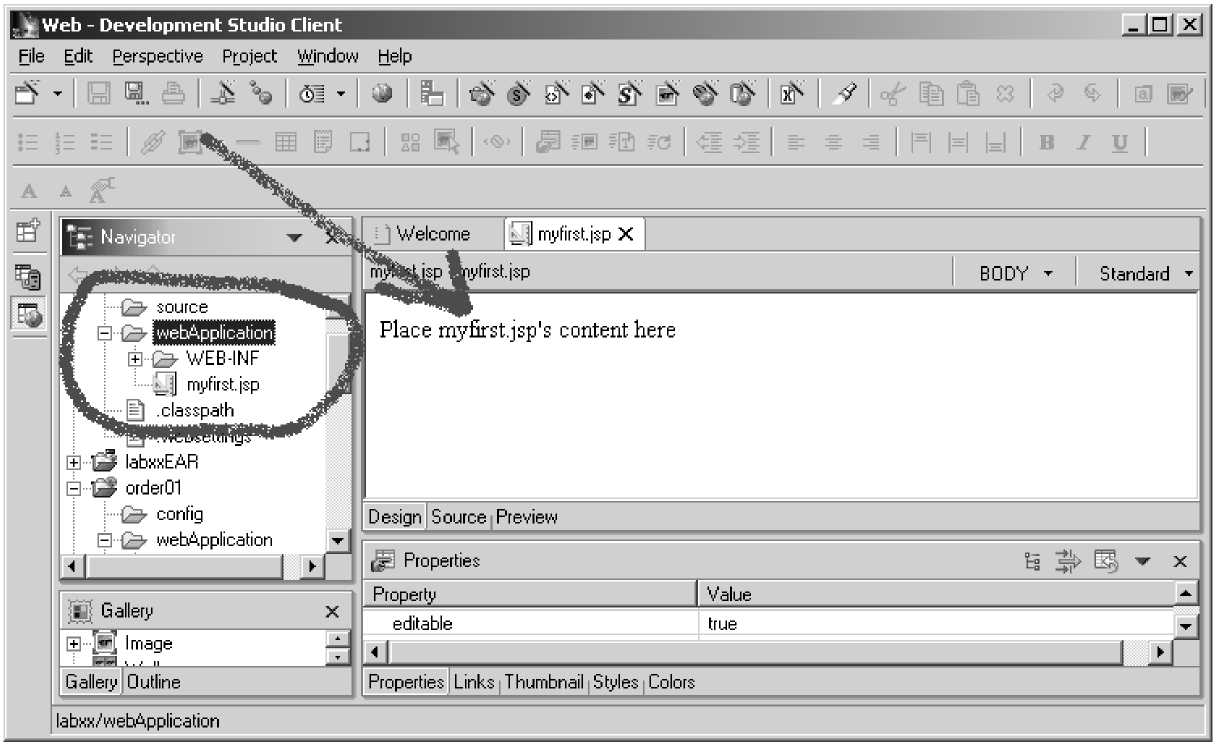
In the list box locate your web project **labxx:**

- Expand **labxx** by clicking on the **+** beside it, if it is not expanded already
- Select the **webApplication** folder in the list

  This selection will appear in the entry field above the list
- Locate the **File name** entry field on this dialog
- Enter **myfirst.jsp** as the file name for the jsp you will create in this Lab
- Leave the **Style Sheet** entry field blank
- Press the **Finish** push button

The design window

Back in the workbench window:



myfirst.jsp : myfirst.jsp

Place myfirst.jsp's content here

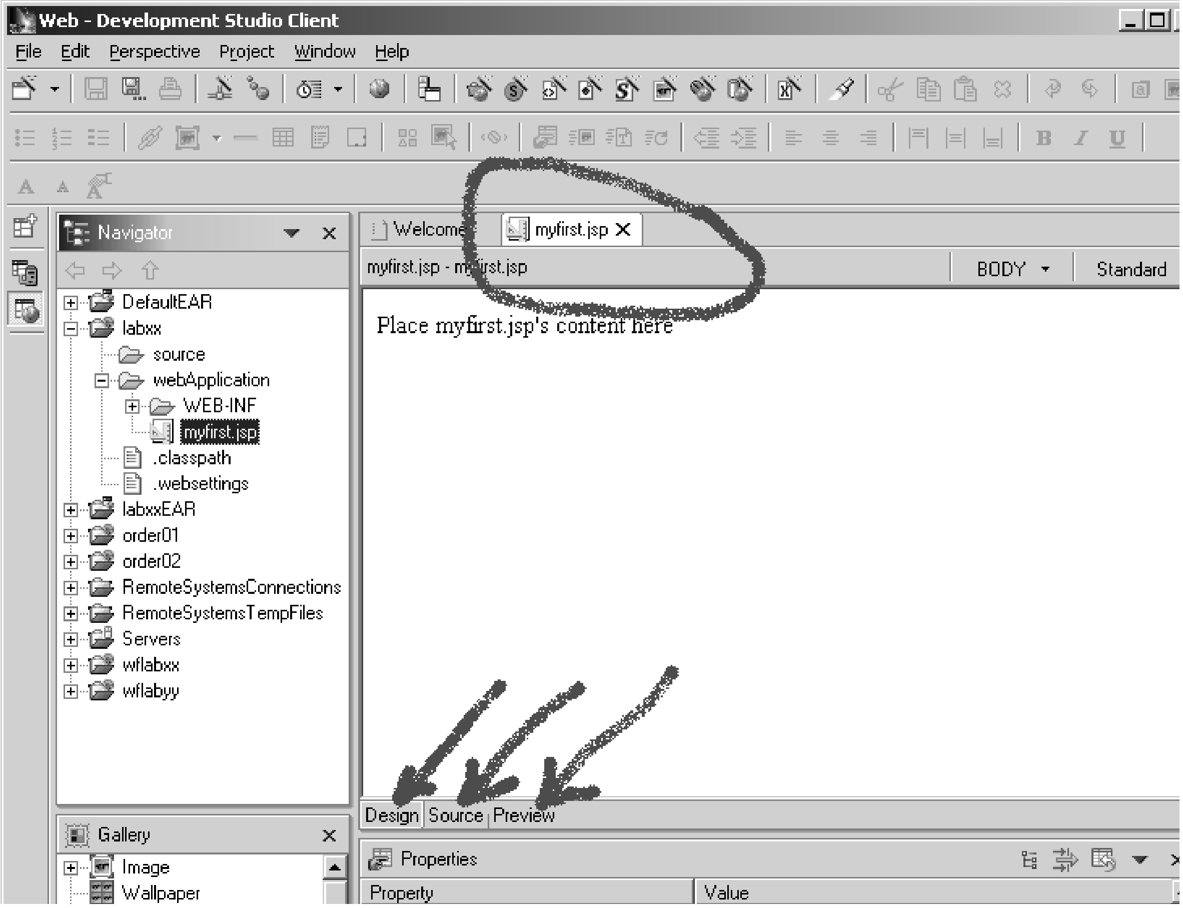
Property	Value
editable	true

- Expand your webApplication folder by clicking on the **+** beside it

Notice the **myfirst.jsp** file that got added to your **webApplication** folder

Also notice that in the **right hand side pane** the Page Designer tool has been opened for you, so you can design your web page. The arrow in the figure above points to the page designer window.

This is the window you will be using to design the first web page.



In the **Page Designer** window itself, you will notice that it allows different views for the file it is currently working with. If you look at the bottom of the **page designer** window you will see 3 tabs for 3 different views. The initial view shows the normal design window.

The following list shows all options available to you

The arrows in the figure above point to the tabs

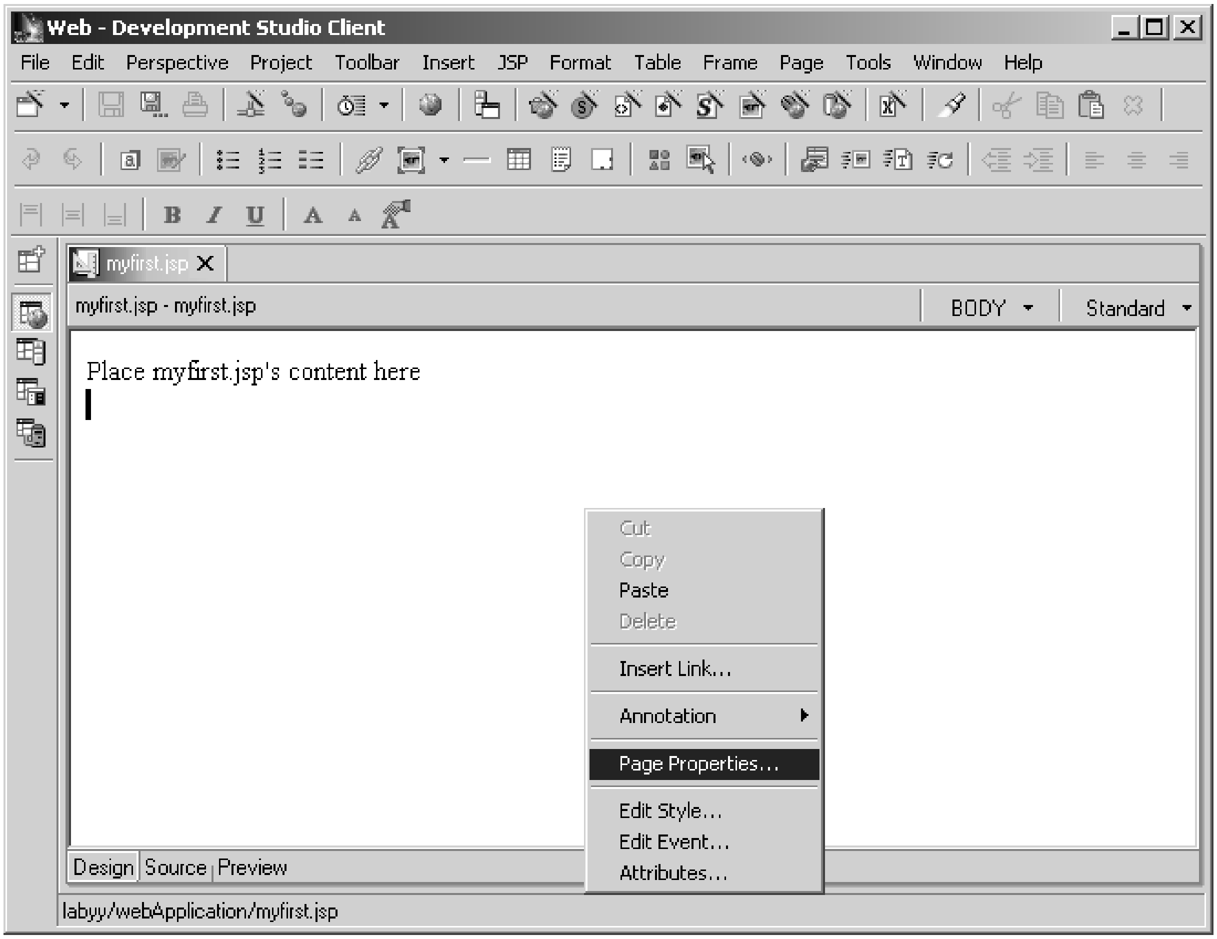
1. The design window
2. An HTML source editor window
3. A Preview window

To enlarge the area of the **page designer** window you can double click on its title bar, the circle in the figure above points to the title bar

Since you are using a blank JSP template file your page is pretty much empty.

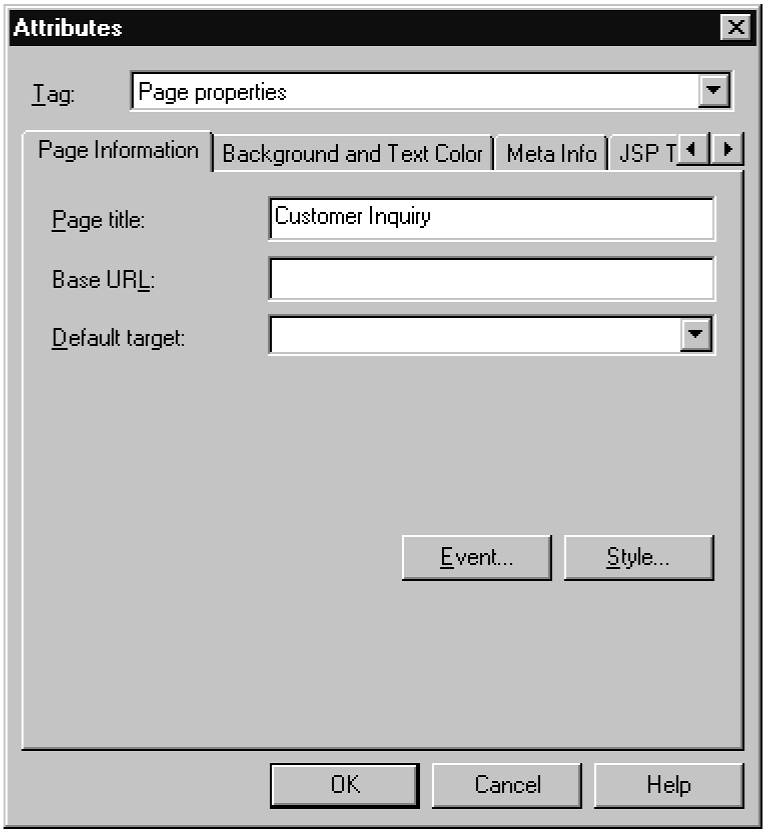
Working with page properties

In the **page designer** window


Web - Development Studio Client
File Edit Perspective Project Toolbar Insert JSP Format Table Frame Page Tools Window Help
myfirst.jsp
myfirst.jsp - myfirst.jsp
BODY
Standard
Place myfirst.jsp's content here
Cut
Copy
Paste
Delete
Insert Link...
Annotation
Page Properties...
Edit Style...
Edit Event...
Attributes...
Design
Source
Preview
labyy/webApplication/myfirst.jsp

- Right mouse click **anywhere on the white space (background**)
- Select **Page Properties**.... from the pop up menu

The **Attributes** dialog will appear



Tag: Page properties

Page title: Customer Inquiry

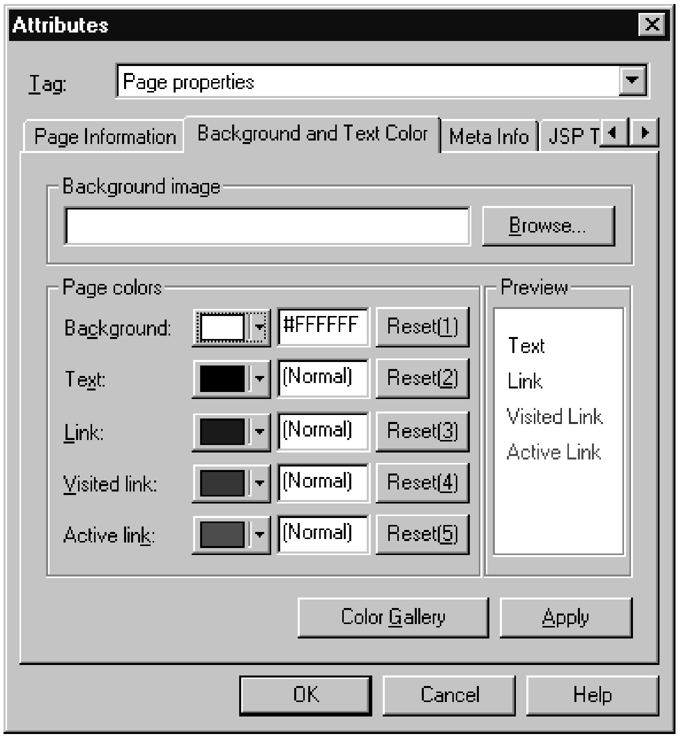
Base URL:

Default target:

- Change the **Page title** entry field to: **Customer inquiry**

- Click on the **Background and Text Color** Tab

The Background and Text Color page appears



Tag: Page properties

Background: #FFFFFF

Text: (Normal)

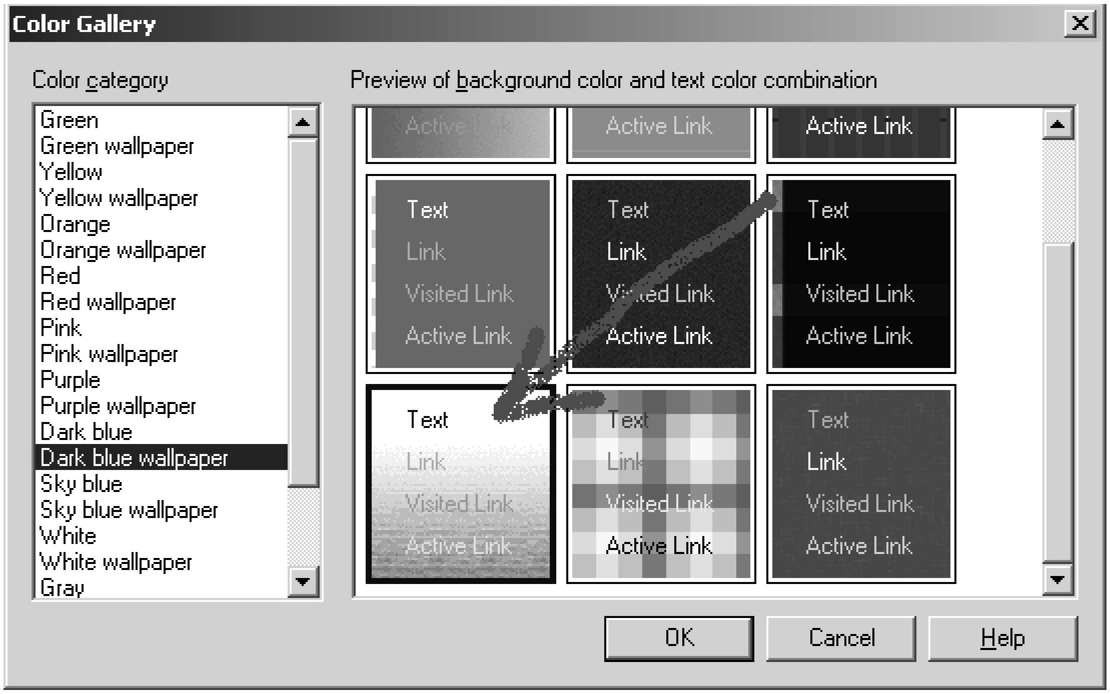
Link: (Normal)

Visited link: (Normal)

Active link: (Normal)

- Press the **Color Gallery** push button

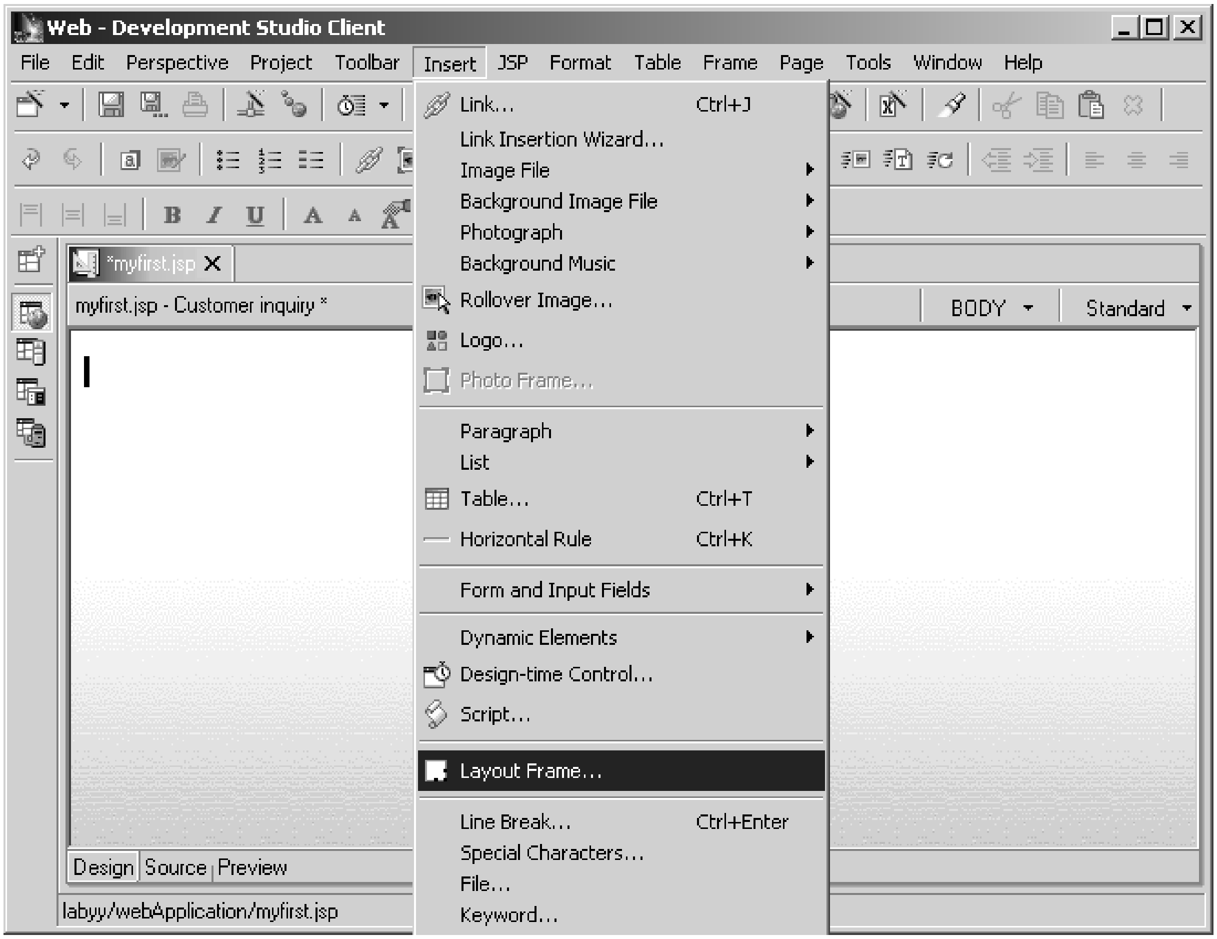
You will see the following dialog:



- Select **Dark Blue wallpaper** from the list on the left side of the dialog
- Select the color choice on the **last row left column** from the **Preview of background color and text color combination** list on the right side of the dialog. As indicated by the arrow in the figure above.
- Press the **OK** push button on this page
- Press the **OK** push button on the **Attributes** dialog to close it

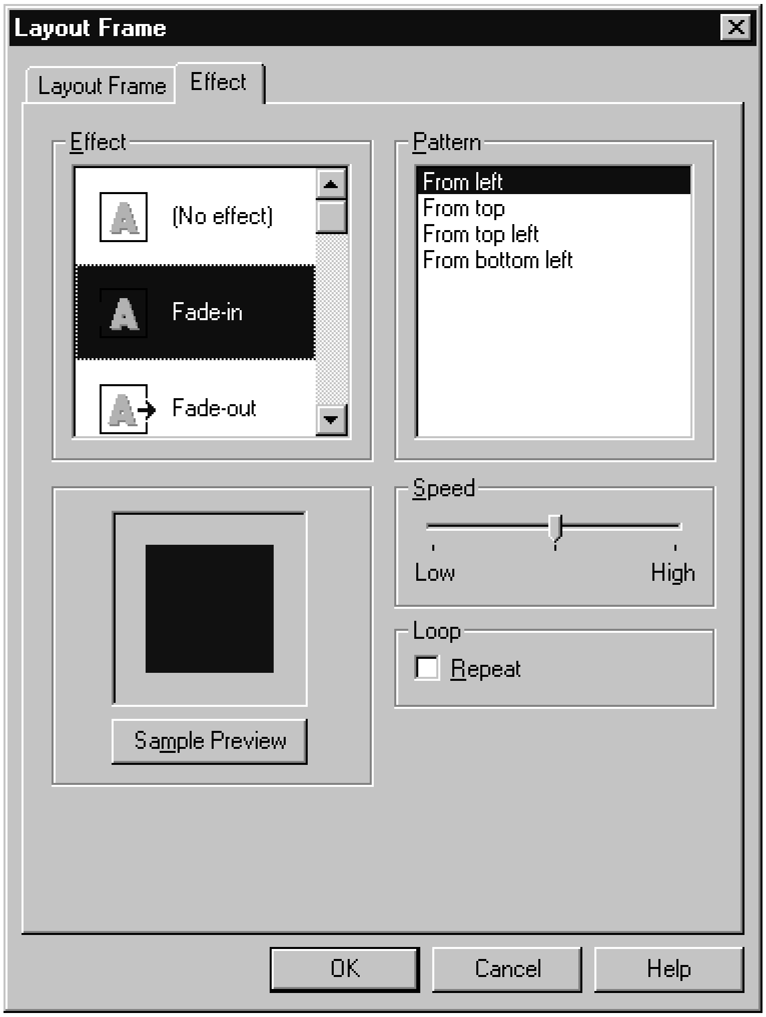
Working with a layout frame

Now you will be adding a **layout frame** to your page. A layout frame allows you to specify an area of your page and lay it out as you see it instead of using html tags or tables to do the layout. A **layout frame** also allows some dynamic design to happen on the page when viewed in the browser.



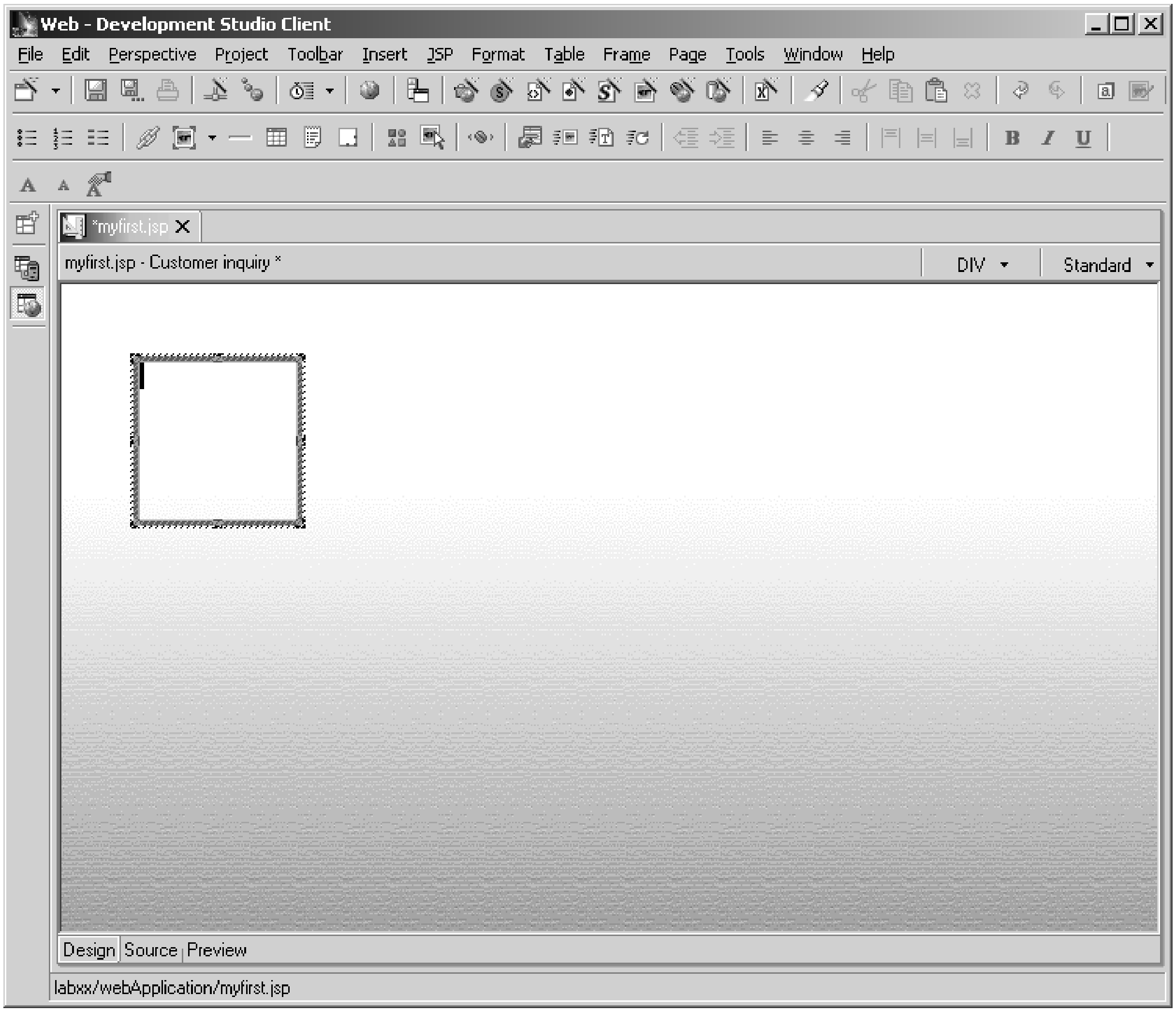
- First, select the text: **Put myfirst.jsp’s content here,** in the design window
- Delete the **selected text,** by pressing the **Delete key** on your keyboard
- Select **Insert** from the **workbench** menu bar
- Select **Layout Frame...** from the pull down menu

The **Layout frame** dialog will appear



- Click on the **Effect Tab**
- Select **Fade-in** from the Effect list
- Press the **OK** push button

Your Design window now contains a Layout Frame


Web - Development Studio Client
File Edit Perspective Project Toolbar Insert JSP Format Table Frame Page Tools Window Help
*myfirst.jsp
myfirst.jsp - Customer inquiry *
DIV
Standard
Design
Source
Preview
labxx/webApplication/myfirst.jsp

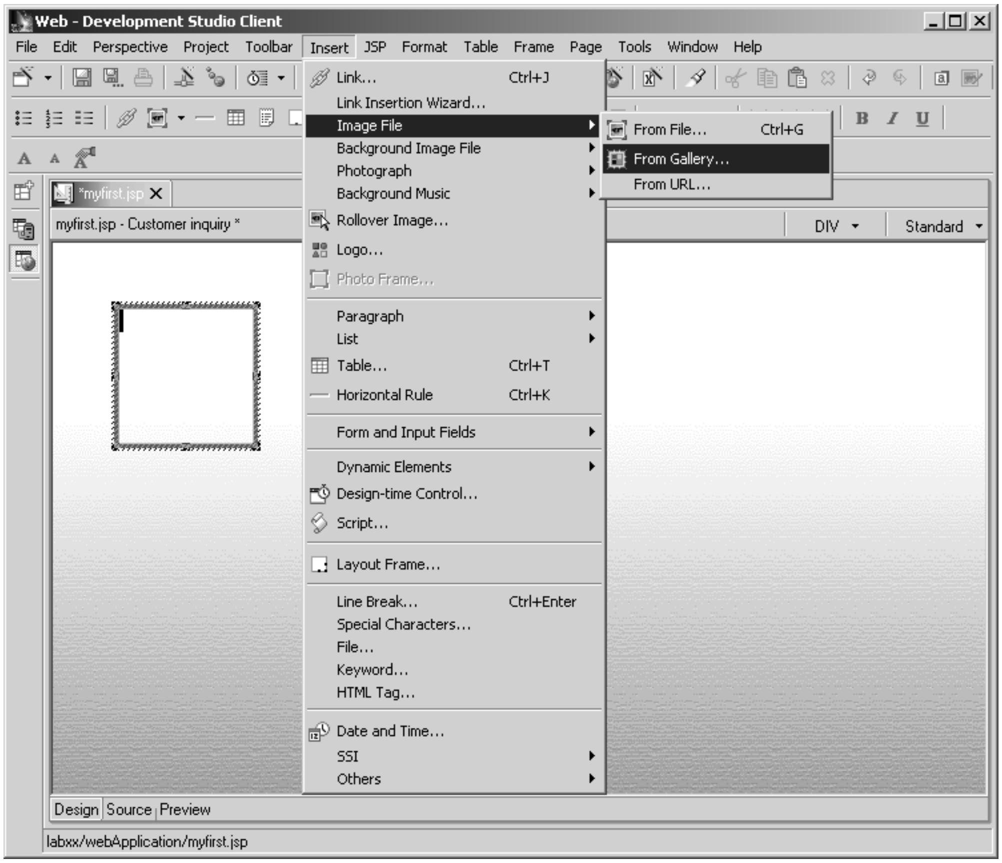
-

Adding an Animated Image to the Page

You will now add an **Animated** picture to the frame.

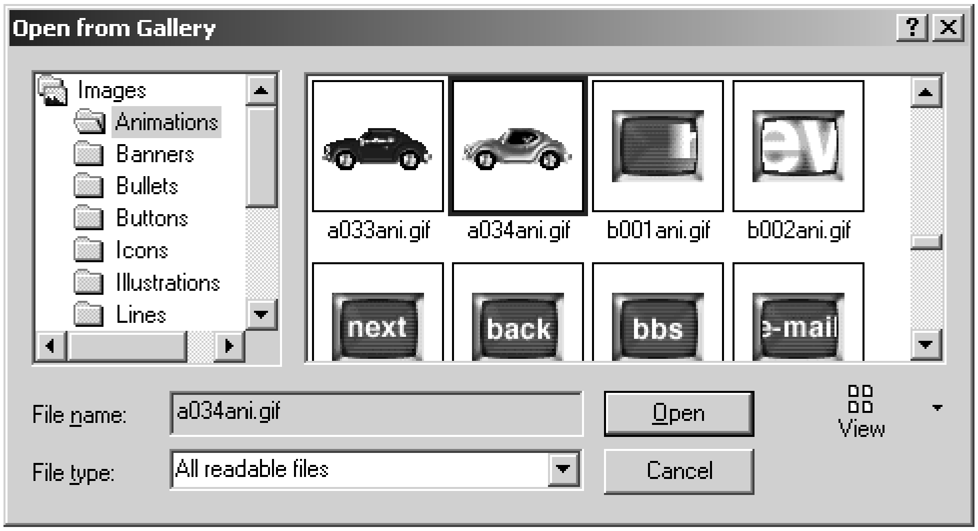
- Mouse click **inside** the **layout** frame in your design window, so the cursor is positioned inside the layout frame

In the workbench



- Select ***Insert -> Image file -> From Gallery*** from the workbench **main** window menu bar

From the **Dialog** in the **Animation folder**

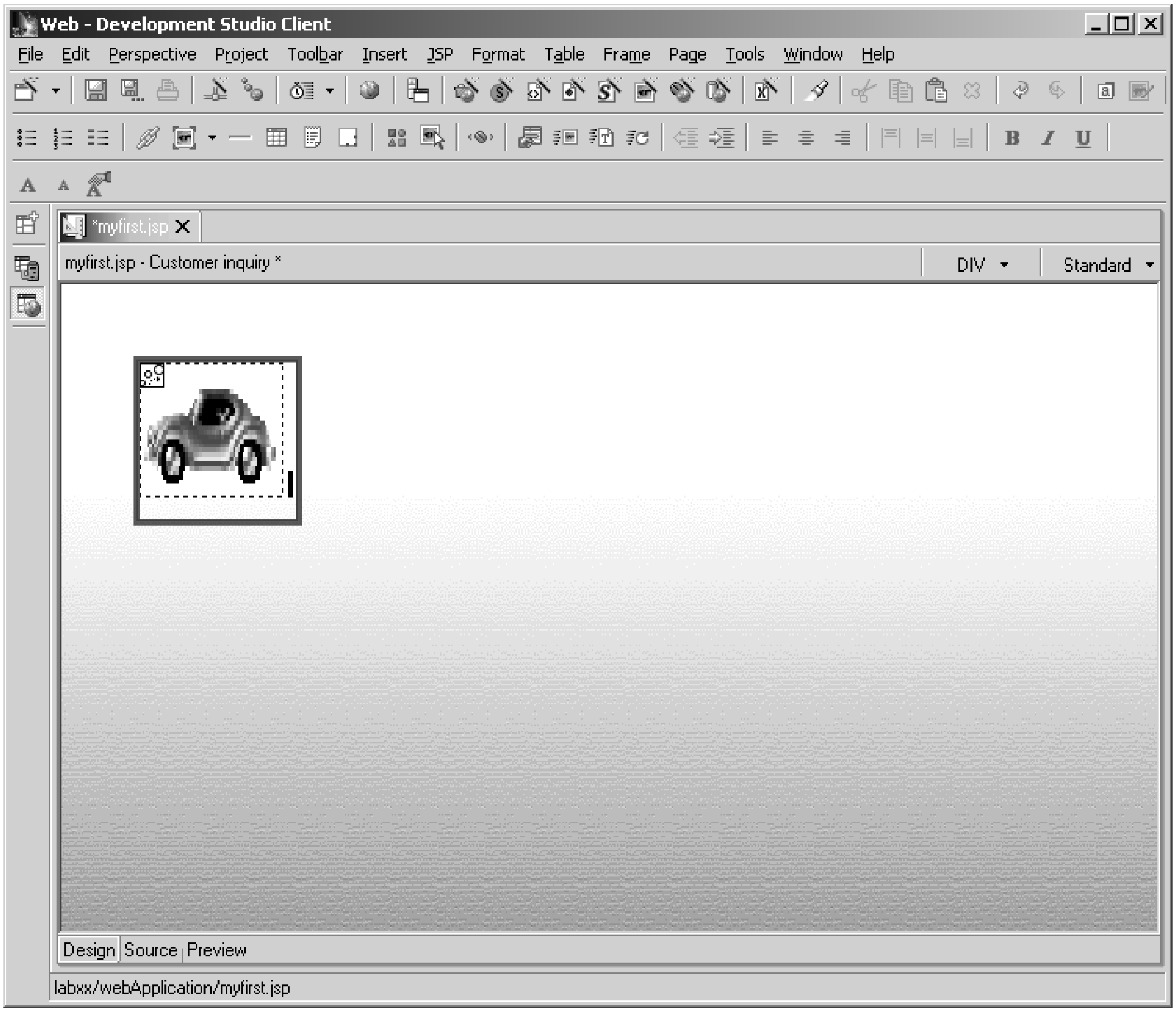


File name: a034ani.gif

File type: All readable files

- Select the **a034ani.gif**, or whatever image you like best
- Press the **Open** push button

The animated image is now part of your page


Web - Development Studio Client
File Edit Perspective Project Toolbar Insert JSP Format Table Frame Page Tools Window Help
*myfirst.jsp
myfirst.jsp - Customer inquiry *
DIV
Standard
Design Source Preview
labxx/webApplication/myfirst.jsp

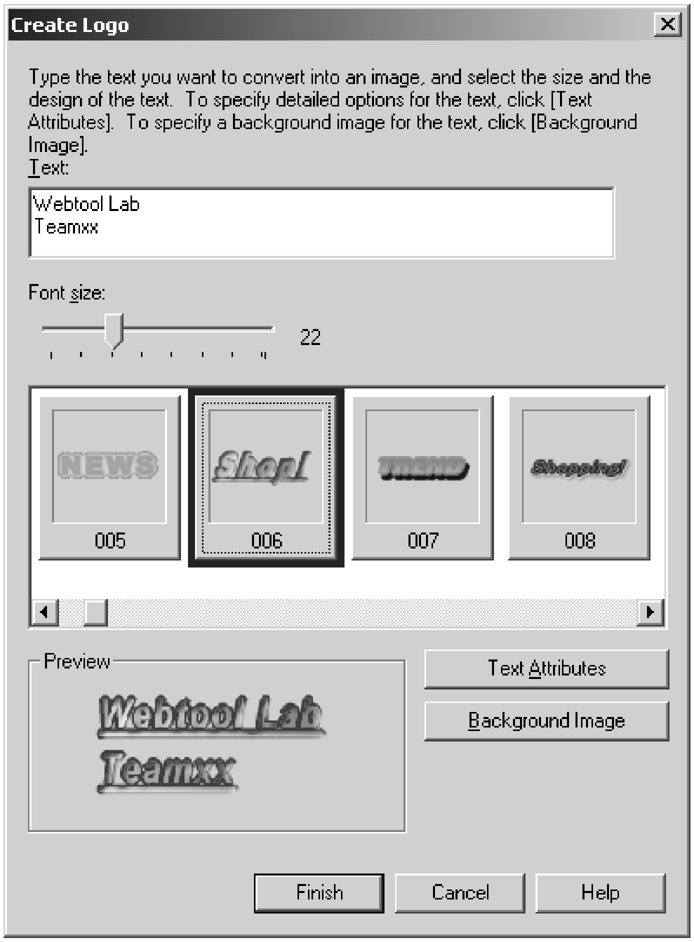
Stretch the picture a bit so it becomes bigger

Adding a Logo to the page using the Logo Wizard

Now you'll add a logo using the **Logo Wizard** that comes with **WDSc**

- Click inside the **layout frame** beside the picture you just added, to position the cursor on the target location for the Logo
- Select ***Insert-> Logo*** from the workbench menu bar

The Logo Wizard dialog appears



Create Logo

Type the text you want to convert into an image, and select the size and the design of the text. To specify detailed options for the text, click [Text Attributes]. To specify a background image for the text, click [Background Image].

Text:

Webtool Lab
Teamxx

Font size: 22

In the text field at the top

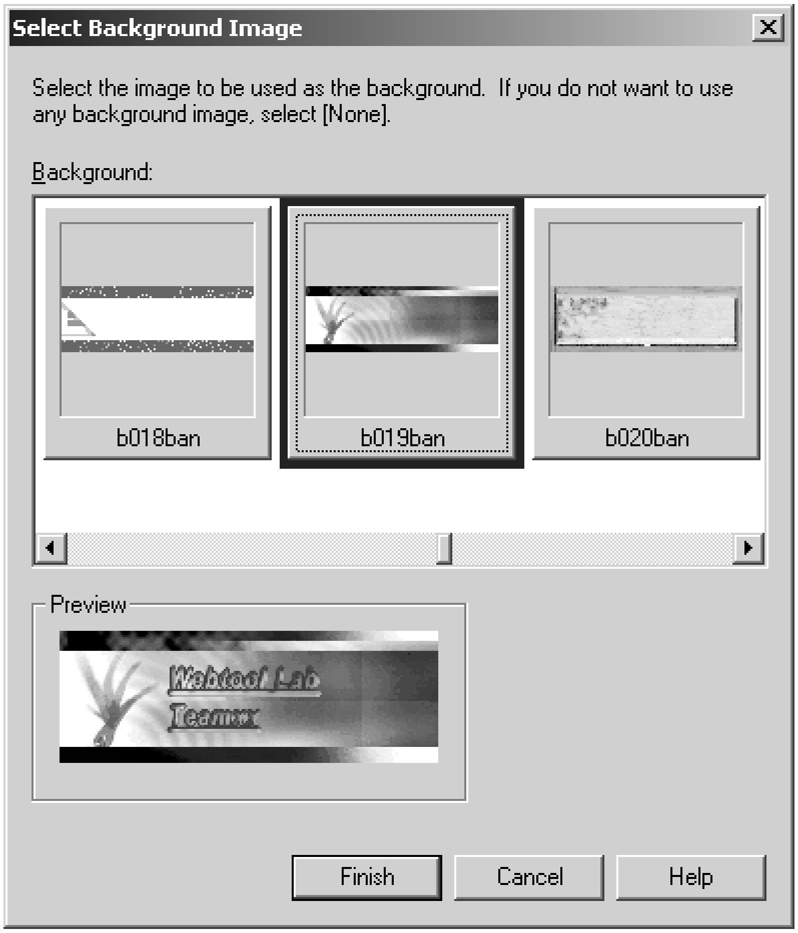
- **Enter**

  Webtool Lab
  Teamxx

> ◆ **xx** being your Lab team number

- Select a **Font** that you like for example **006** as in the figure above, use the slider underneath the font list to view the different fonts available.
- Press the **Background Image** push button

The Select Background Image dialog comes up in the Logo wizard



Select the image to be used as the background. If you do not want to use any background image, select [None].

- Move the slider to the right until you find background **b019ban**
- Select**, b019ban**, as the background of choice, or another one that you like most
- Press the **Finish** push button on the select background page
- Press the **Finish** push button on the Create Logo page

You are done with the Logo design.

- Check your design page

Does it look good? We hope it does.

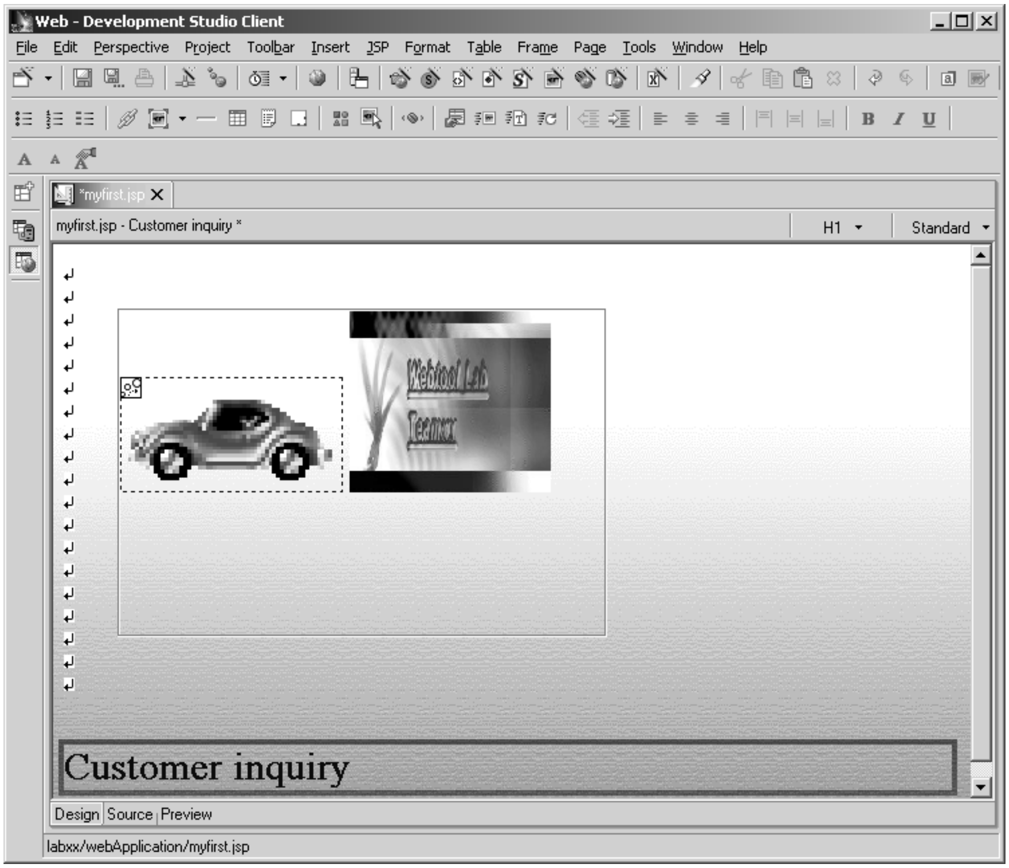
You might want to shrink the Logo a bit.

If the LOGO doesn't appear beside the image don't worry leave it where it is for now.

Creating a heading

Now you will create a heading **underneath** the **LOGO** you just created:

- Click with the left mouse button in the white space above the layout frame in your JSP page in the **page designer** window. The cursor will be located at the top of the page.


Web - Development Studio Client
File Edit Perspective Project Toolbar Insert JSP Format Table Frame Page Tools Window Help
*myfirst.jsp
myfirst.jsp - Customer inquiry *
H1
Standard
Webtool Lab
Teamxx
Customer inquiry
Design Source Preview
labxx/webApplication/myfirst.jsp

- Press the **Enter** key a couple of times until the **cursor is located below** the Layout Frame

On the workbench menu bar

- Select **Insert --> Paragraph --> Heading 1**
- Now in the Design window key in

Customer Inquiry

Now you will create an **entry field** and a **heading** for the data input on your web page. This entry field will be used to enter a customer number. You will send the customer number to an iSeries RPG program which will retrieve the customer record for this customer number and send back the data base information to a Java Server Page (JSP) to show the data in a browser.

Adding a Form

In the **second part** of this Lab you will use a **servlet** (a JAVA program that gets generated by **WDSc**) to invoke an **RPG** program on the iSeries. In order to invoke this **servlet** you need a **Form** on your web page.

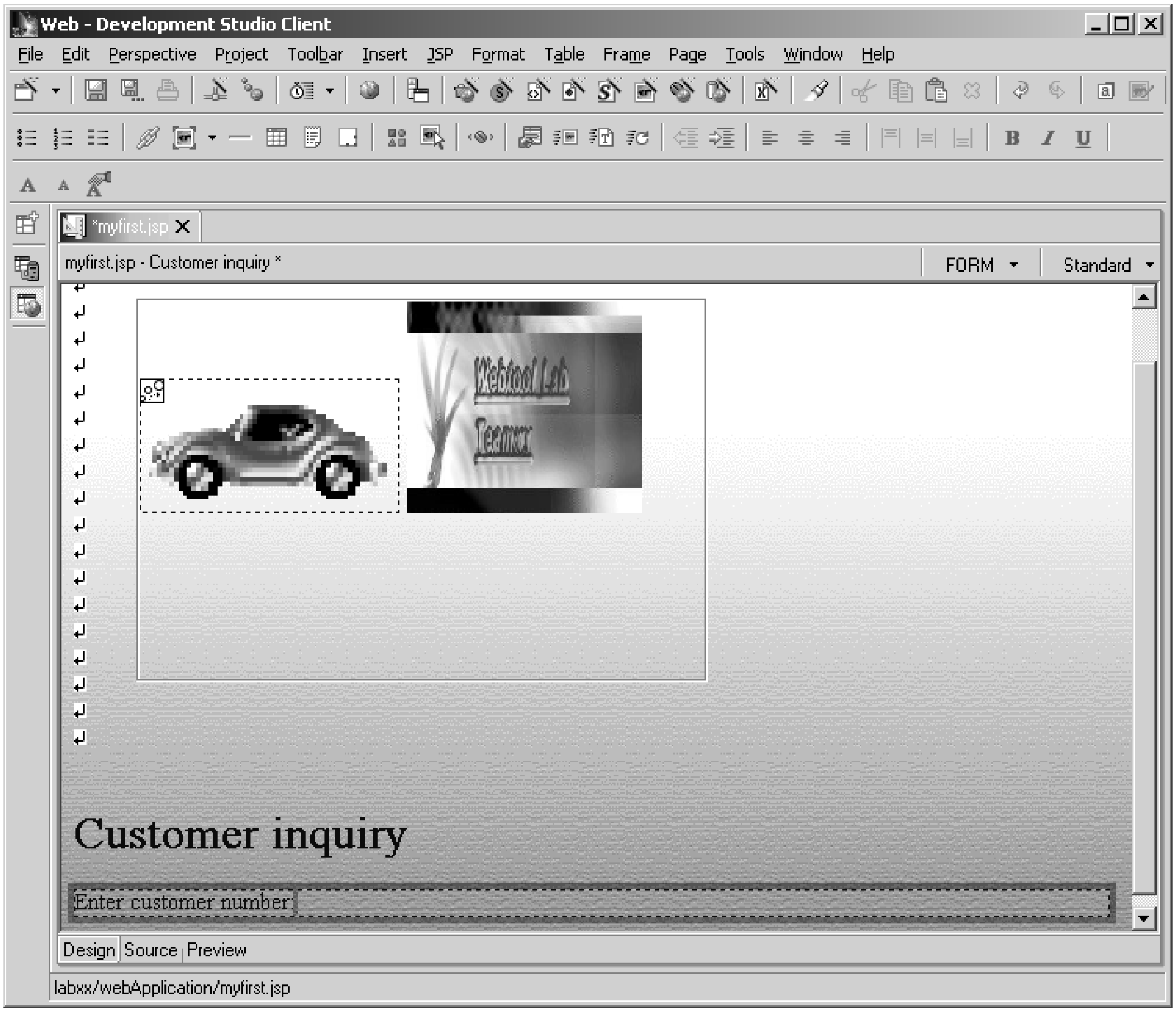
- In the Design window, **click** with your **left** mouse button on the **blue space** underneath the heading that you just inserted.

> - **Note:** If the cursor doesn't position underneath the heading, put it **beside** the heading and **press** the Enter key.

After the cursor is positioned there, on the workbench menu bar

- **Select Insert --> Form and Input Fields --> Form**

A form, that appears as a frame on the design window, has been added to your design window


Web - Development Studio Client
File Edit Perspective Project Toolbar Insert JSP Format Table Frame Page Tools Window Help
*myfirst.jsp
myfirst.jsp - Customer inquiry *
FORM
Standard
Webtool Lab
Teamxx
Customer inquiry
Enter customer number:
Design Source Preview
labxx/webApplication/myfirst.jsp

- Position the cursor inside the form
- **Key in**

Enter customer number:

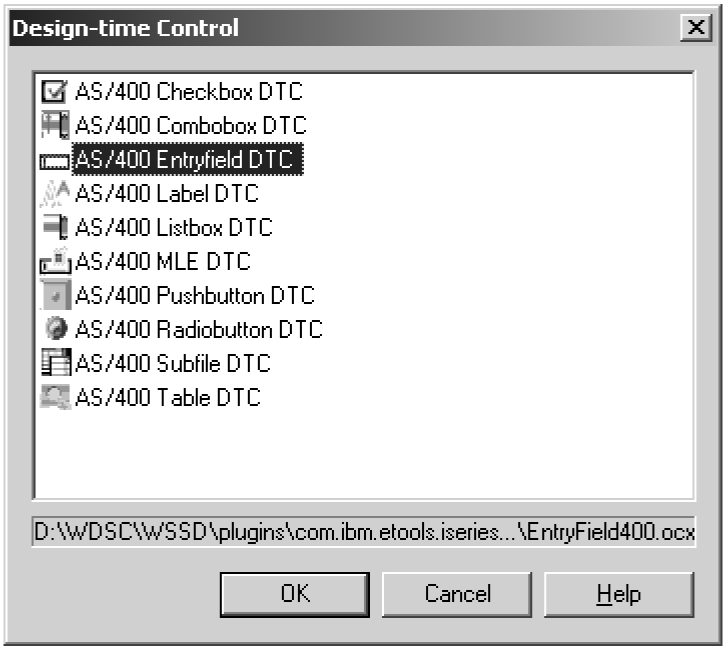
You will now add an **Entry field** to the form on your web page. You will use an **iSeries Entry field** which is implemented as a **Design Time Control** (DTC)

Adding Design Time controls (iSeries entry field)

With the cursor located beside the text you just entered

- Select **Insert --> Design-time Control...** from the workbench menu bar

The figure below shows a list of registered DTC's on a particular PC. It may vary from the list you see on your workstation

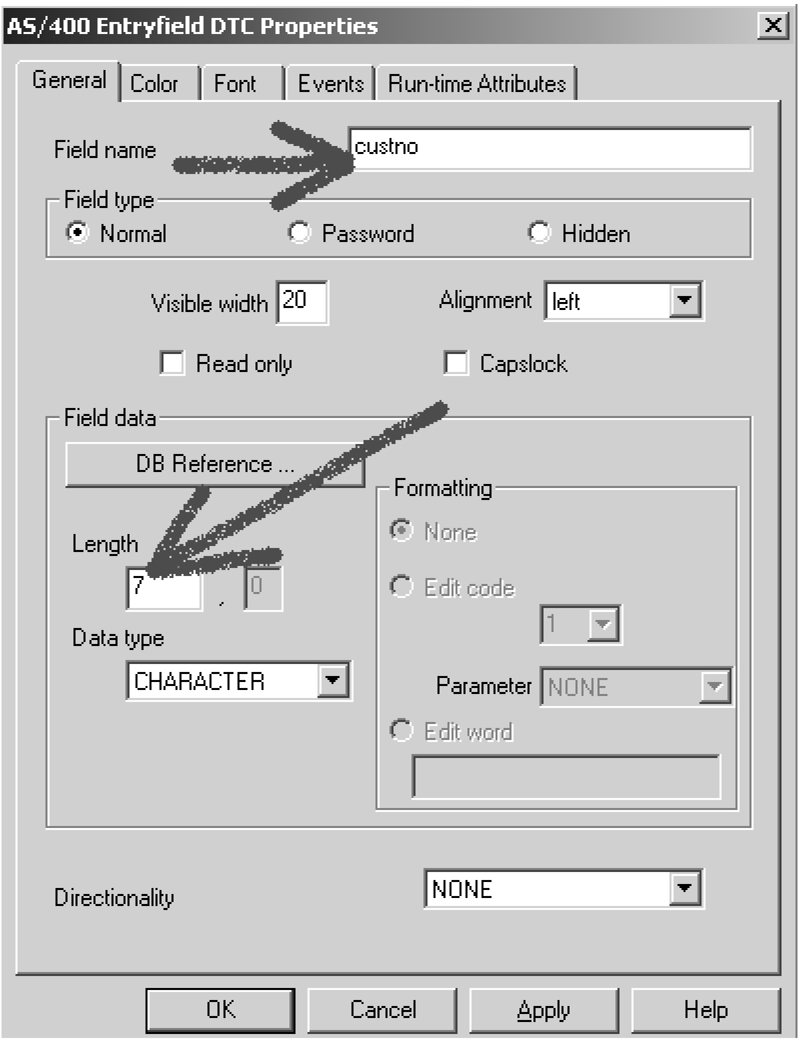


- Select **AS/400 Entry field DTC** from the list in the Design-time Control dialog
- Press the **OK** push button

Back on the design page in the page designer:

- Select the **Entry field** you just added
- Right mouse click on the **Entry field**
- Select **Control properties** from the pop up menu

On the properties dialog that appears



Field name: custno

Field type: ● Normal ○ Password ○ Hidden

Visible width: 20 Alignment: left

□ Read only □ Capslock

Field data

Length: 7 , 0

Data type: CHARACTER

Formatting: ● None ○ Edit code 1 Parameter: NONE ○ Edit word

Directionality: NONE

- Enter ***CUSTNO*** for **Field name**
- Enter **7** for the **Length** of the field
- Press the **OK** push button

The last part you need on this page is a push button to send a **submit** request to the servlet, so it can invoke the iSeries program and return some data.

Adding Design Time controls (iSeries push button)

Position the cursor **to the right of** the entry field you just added and press **Enter twice**

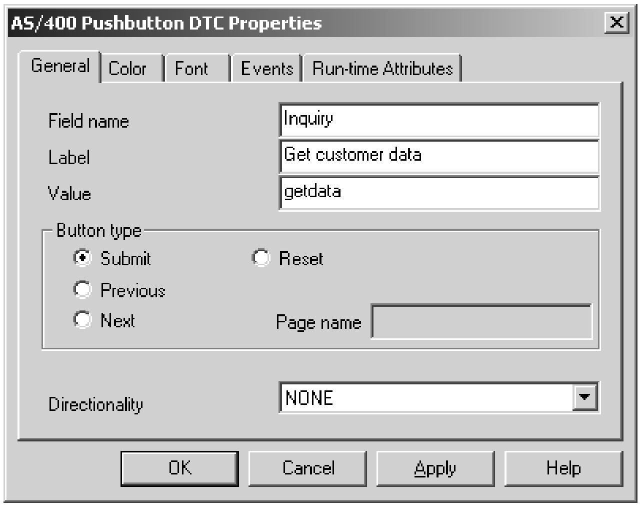
This positions the cursor below our entry field heading inside the **Frame**.

- Select on the workbench menu bar ***Insert --> Design Time Control***
- From the dialog select **AS/400 Pushbutton DTC**
- Press the **OK** push button

In the Page designer window

- Right mouse click on the new **Pushbutton400** DTC you just added
- Select **Control Properties** from the pop up menu

The AS/400 Pushbutton DTC Properties dialog shows



Field name	Inquiry
Label	Get customer data
Value	getdata

Button type

● Submit ○ Reset

○ Previous

○ Next Page name

Directionality NONE

- Change the **Field name** to ***Inquiry***
- Change the **Label** to ***Get customer data***
- Change the **value** to ***getdata***
- Press the **OK** pushbutton on the **properties dialog** to change the AS/400 pushbutton properties

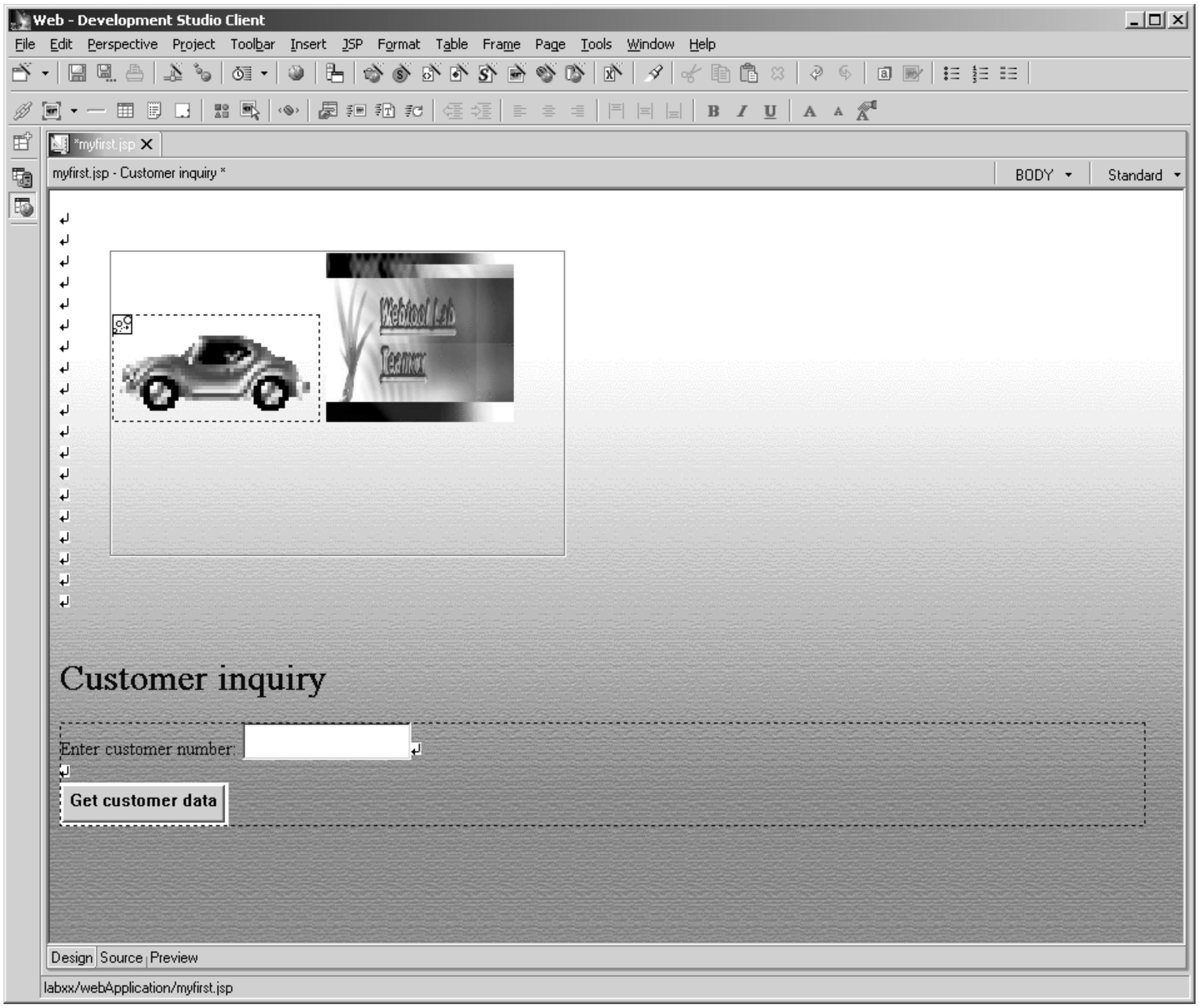
Here are some explanations to the push button attributes you just changed

Field name is the **control name**,

Label is the text that will appear on the push button

Value is data that is sent to the RPG program to determine which push button has been pressed. If you had **multiple** push buttons on a page then this field comes in handy to identify which push button got pressed. This is like an indicator that is set on by different command keys in a **5250** interface.

If your web page looks similar or better than the figure below you did a fantastic job.

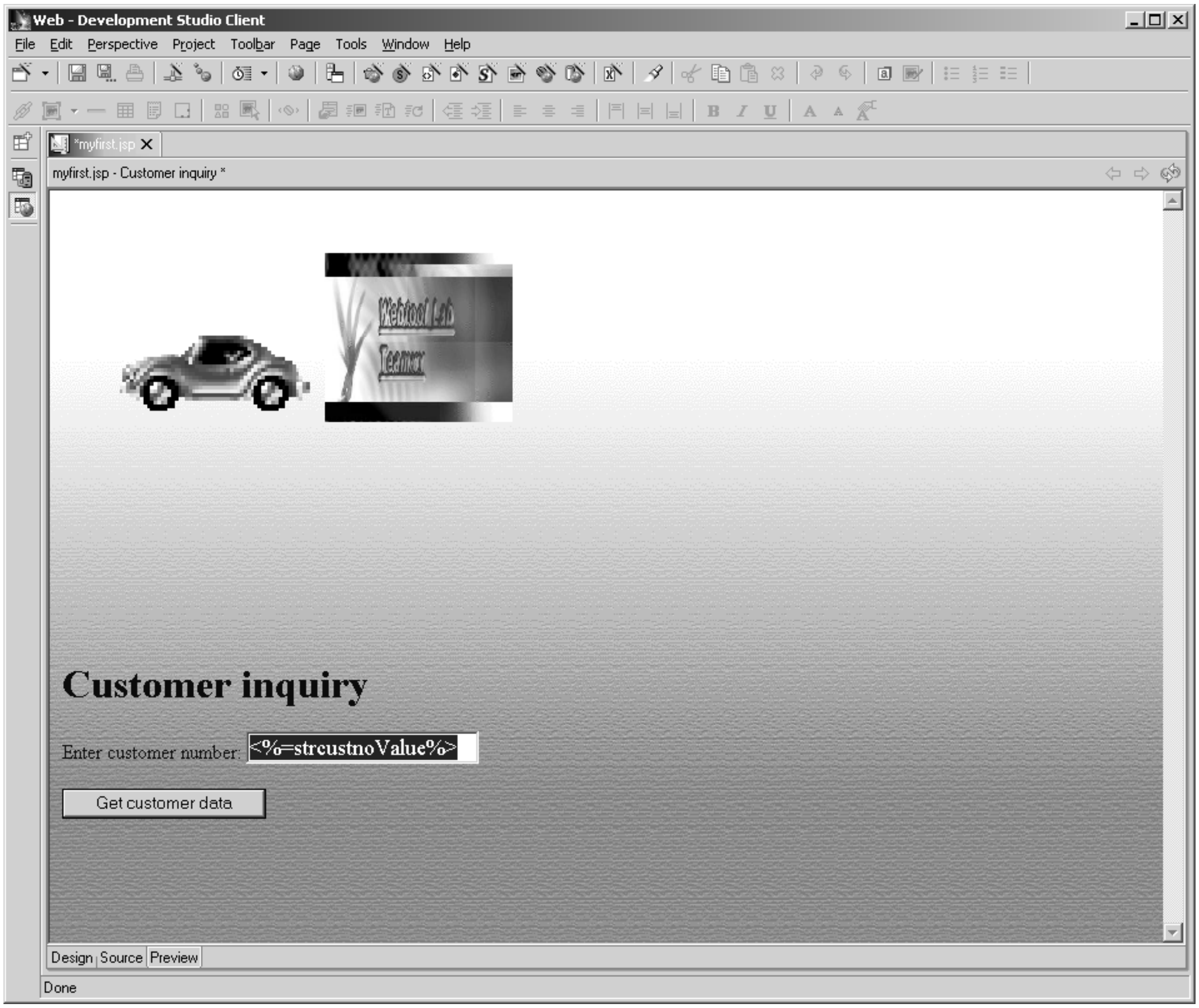

Web - Development Studio Client
File Edit Perspective Project Toolbar Insert JSP Format Table Frame Page Tools Window Help
*myfirst.jsp
myfirst.jsp - Customer inquiry *
BODY
Standard
Webtool Lab
Teamxx
Customer inquiry
Enter customer number:
Get customer data
Design Source Preview
labxx/webApplication/myfirst.jsp

Preview your page

This completes the user interface design

Preview your web page by

- Clicking on the **preview tab** at the bottom of the ***page designer*** pane



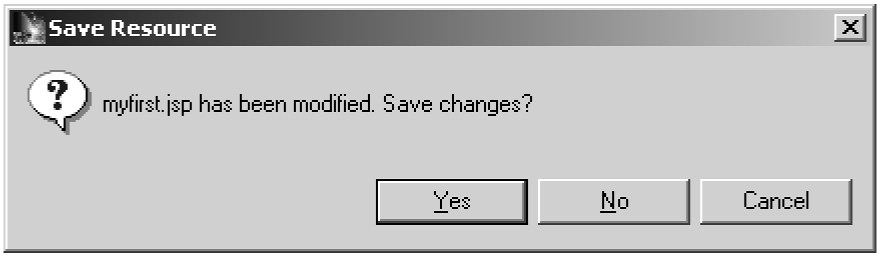
Don't worry about the funny string in your entry field **<%=strcustnoValue%>** this will not show up at runtime.

You might want to check out the additional tab in **Page designer**

- **HTML Source** shows you the **HTML** tags generated for your design page.

Close the Page Designer by:

- Selecting the **File** menu option on the **workbench window**
- Selecting **Close** from the menu pull down



Save Resource

myfirst.jsp has been modified. Save changes?

- Press the **Yes** pushbutton to **Save** the JSP File

Files added

Notice that there have been some files added to your project, these are image files needed to display your animated web page.

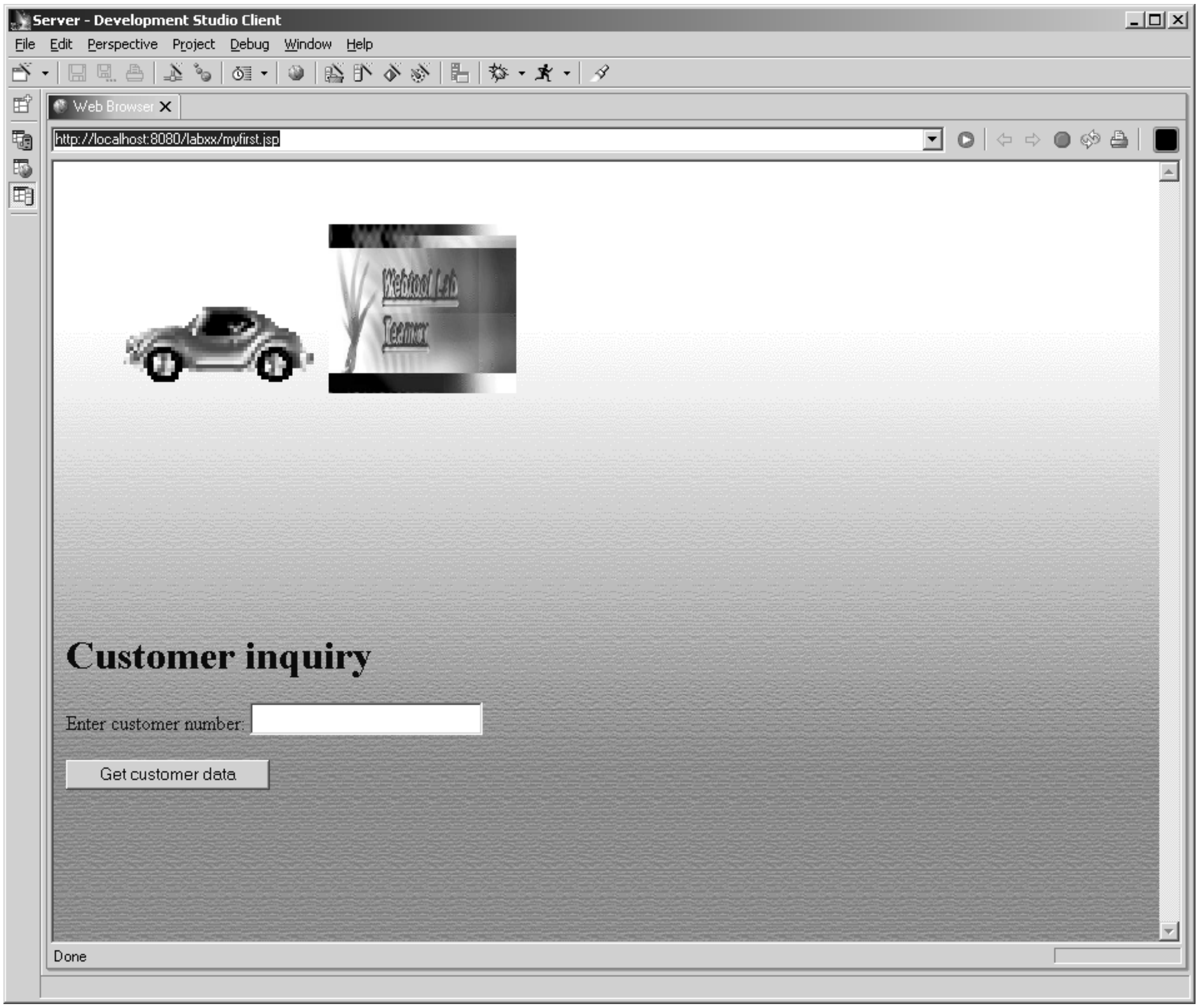
Normally you would now publish (copy) the files to your web server and test your user interface. WDSc comes with a built in WebSphere Application Server Test Environment that is fully integrated into the workbench. That enables you to test your application directly in the workbench.

Testing your web page

In the workbench, in the Navigator view,

- Select the **myfirst.jsp** file in your l**abxx** project under the webApplication folder
- Right mouse click on the **myfirst.jsp** icon and select **Run on Server** from the pop up menu

After a few moments the browser should show your web page


Server - Development Studio Client
File Edit Perspective Project Debug Window Help
Web Browser
http://localhost:8080/labxx/myfirst.jsp
Webtool Lab
Teamxx
Customer inquiry
Enter customer number:
Get customer data
Done

If you get an error that the myfirst.jsp can’t be found, then the automatic publishing has failed.

Only if you get this error go through these steps

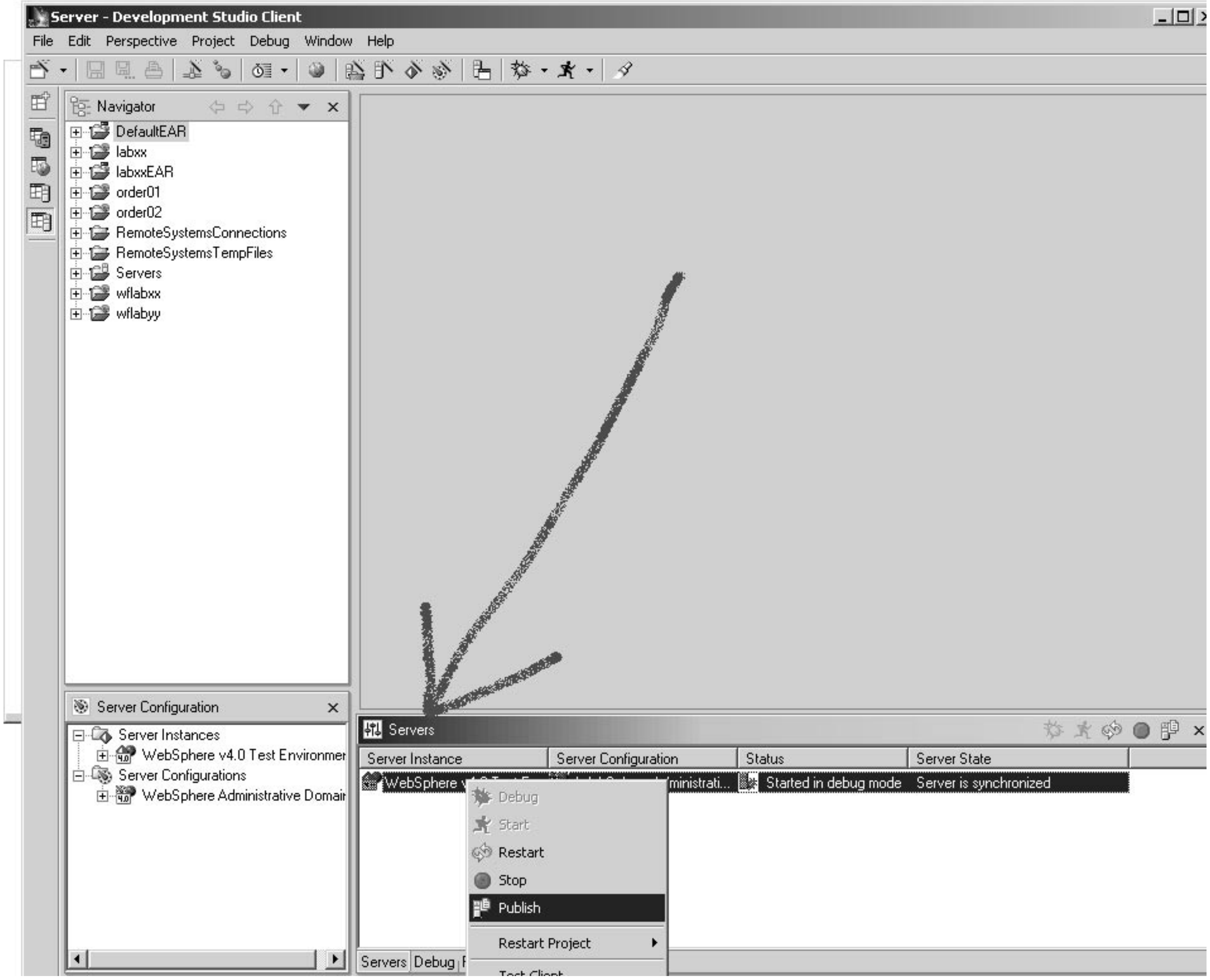
 otherwise proceed to the heading ***Lab Summary***

Publishing manually

In the workbench menu bar

- Select the **Perspective** menu item
- Select **Open** from the sub menu
- Select **Other...**
- Select **Server** from the Select Perspective Dialog
- Press the **OK** push button

You should now have the Server perspective open, one of the views in this perspective, is the Servers view.



For the location of the Server view look at the arrow in the figure above

In the **server view list**, you will find the WAS server instance running in the test environment

- Right mouse click on the **Server instance** in the list
- Select **Publish** from the pop up menu

It will now publish(copy) all your project files to this server environment

If a message dialog shows that the publishing has finished

- Press the **OK** button on the message dialog
- Now go back and run your **myfirst.jsp** on the server.

This concludes the **first** lab exercise.

Before you go ahead, you need to close the browser

- Click on the **X** in the Browsers window bar

This will make sure that your jsp is unloaded from the browser Cache.

Lab Summary

Congratulations, your first **web page** is completed:

You have learned:

- How to create and design a **web page** using the page designer tool in WDSc
- How to create a **dynamic image**
- How to create a **LOGO**
- How to use **AS/400** Design time controls (**DTC's**)
- How to test your web page in the WDSc WAS test environment

Now you are ready to build a web application that accesses data on the iSeries and returns it to a web page.

***Just experience how easy it is*.**

Go ahead to the next exercise

Using the Interaction Wizard

Enhance Your Application with logic to access DB2/400 and return data to your web page

Exploring the WDSc Interaction Wizard and invoking iSeries programs

During this exercise you will evolve the web page you created in the previous lab to be the front end for a customer inquiry application. The following tasks will be covered in this lab:

- Creating an Output (Result) Page for a web application with the Web Interaction Wizard
- Referencing fields in a DB2/400 database to build a PCML data structure
- Creating the Program Call Markup Language (PCML) definition of the parameters being passed to/from the iSeries program
- Write a Service Program or a regular program to interact with web pages
- Generate JAVA code for Servlets using the Interaction Wizard

As a result of this exercise, you will create a web page that is used to display customer information given the input from the web page you created in the previous exercise.

What You Should be Able to Do

As a result of this exercise you can use the **WDSc Interaction Wizard** and **Remote System Explorer** to write the necessary program logic to create a dynamic web environment with RPG. You will be able to:

- Access an iSeries data base from a web browser
- Use existing **DB2/400** field definitions to create the necessary PCML information to be able to communicate with an iSeries program
- Use **externally described iSeries fields** in your web pages

- Get data from the iSeries by invoking an iSeries program that accesses the database and returns the data via parameters
- Read and write data from/to a web page
- Test the application in the WDSc WAS test environment
- Run a web application in the WAS test environment and use it to access iSeries data by calling an RPG program.

Using the Development Studio client for iSeries Web Interaction Wizard to connect with iSeries programs

During this exercise you will use the **interaction wizard** to create the interface description for parameters to be passed to an iSeries program. You will also generate a web page based on the parameters definition used by the program. Some of the parameters will be defined based on existing iSeries database field definitions.

Using the Web Interaction Wizard

If the workbench is still up from the last exercise make sure you are working with the Web Perspective

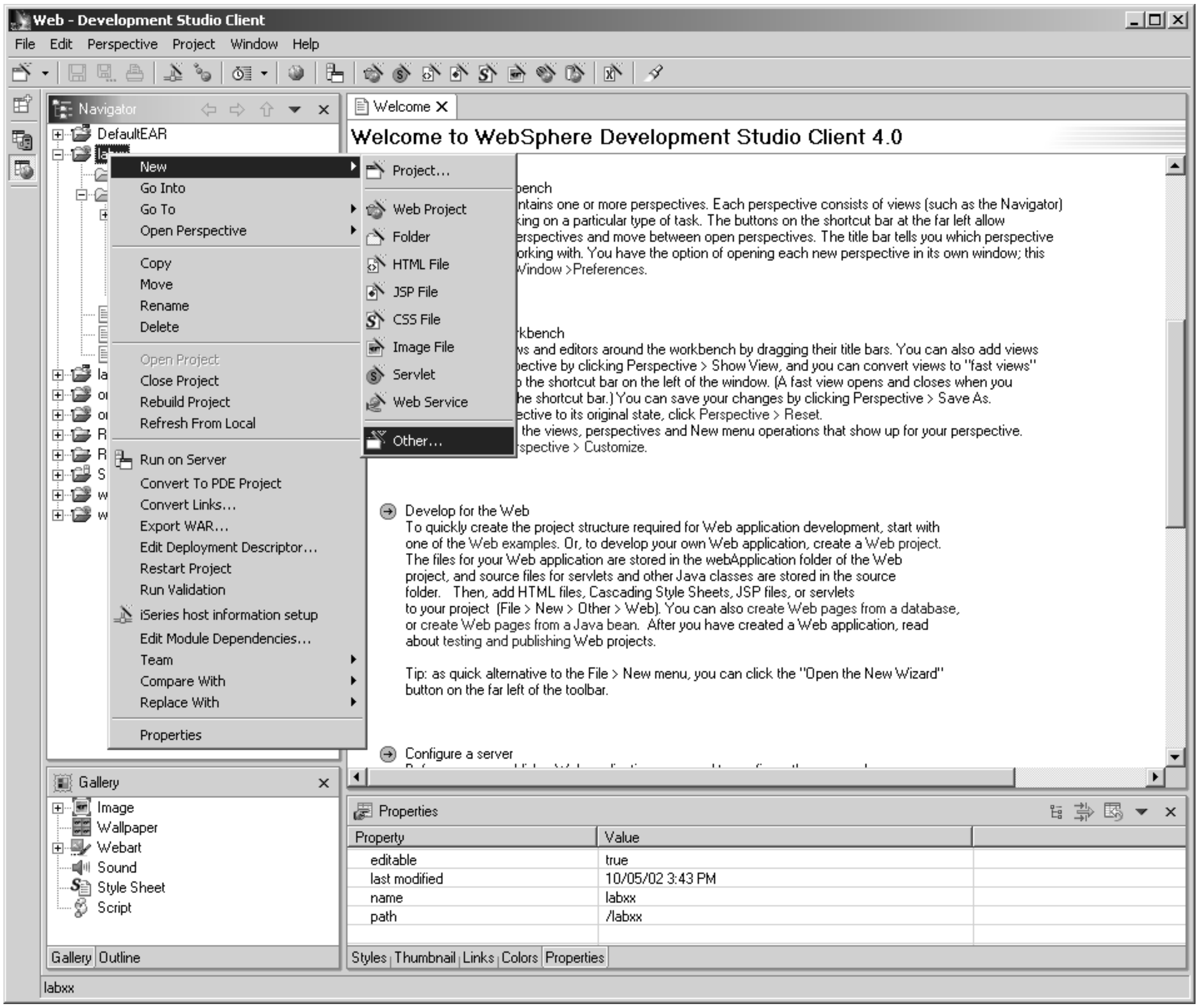
If the **WDSc** workbench is not up,

- Start the **workbench**
- Open the **web perspective**

In the Web Perspective

- Locate your Web project **labxx**, that you created in the previous exercise

To invoke the Web Interaction Wizard:



Welcome to WebSphere Development Studio Client 4.0

bench
ntains one or more perspectives. Each perspective consists of views (such as the Navigator)
king on a particular type of task. The buttons on the shortcut bar at the far left allow
erspectives and move between open perspectives. The title bar tells you which perspective
orking with. You have the option of opening each new perspective in its own window; this
Window >Preferences.

kbench
ws and editors around the workbench by dragging their title bars. You can also add views
pective by clicking Perspective > Show View, and you can convert views to "fast views"
the shortcut bar on the left of the window. (A fast view opens and closes when you
he shortcut bar.) You can save your changes by clicking Perspective > Save As.
ective to its original state, click Perspective > Reset.
the views, perspectives and New menu operations that show up for your perspective.
rspective > Customize.

Develop for the Web
To quickly create the project structure required for Web application development, start with one of the Web examples. Or, to develop your own Web application, create a Web project. The files for your Web application are stored in the webApplication folder of the Web project, and source files for servlets and other Java classes are stored in the source folder. Then, add HTML files, Cascading Style Sheets, JSP files, or servlets to your project (File > New > Other > Web). You can also create Web pages from a database, or create Web pages from a Java bean. After you have created a Web application, read about testing and publishing Web projects.

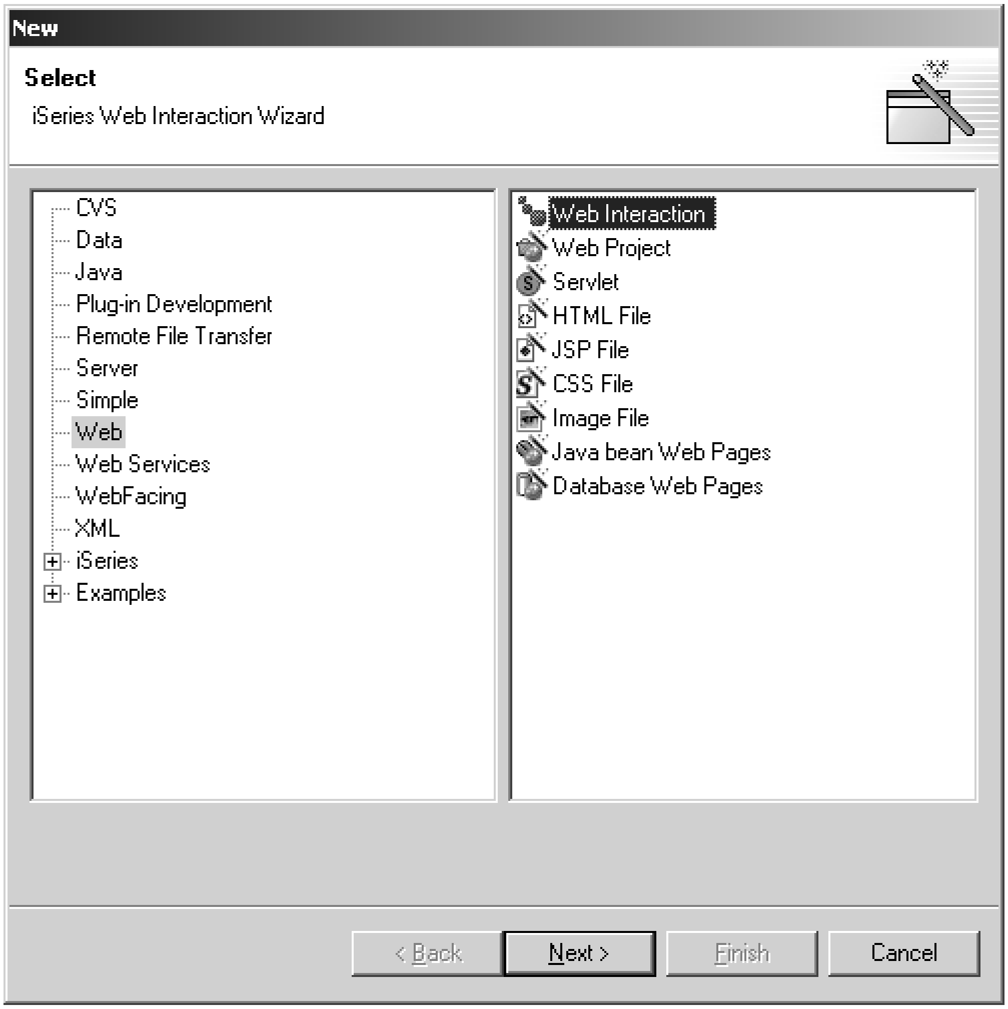
Tip: as quick alternative to the File > New menu, you can click the "Open the New Wizard" button on the far left of the toolbar.

Configure a server

Property	Value
editable	true
last modified	10/05/02 3:43 PM
name	labxx
path	/labxx

- Right mouse click on the **labxx project icon** in the **left hand view**
- Select **New** from the pop up menu
- *Select* ***Other..*** from the cascading menu

The **New Select** dialog will come up



- Select **Web** from the left hand list
- Select **Web Interaction** from the right hand list
- Press the **Next >** push button

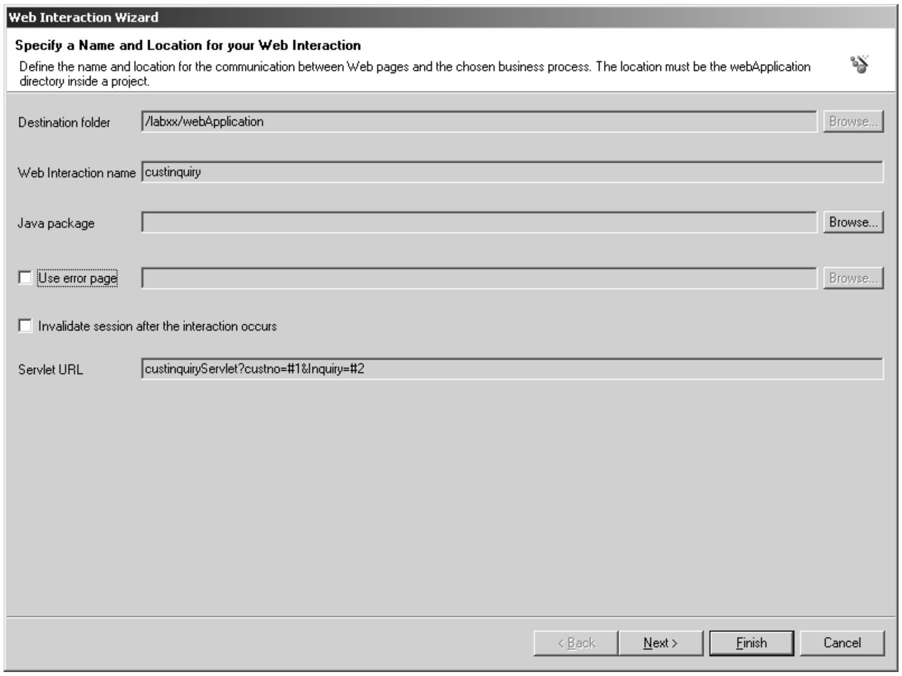
The Web Interaction Wizard dialog appears

Creating a new interaction

An interaction is the **span** between the **submit** of a request from the current web page to the **post** of a new web page. An interaction could be a simple request to the HTTP server to load another page. In this environment an interaction will be made up of a call to an iSeries program, waiting for the return of the program with data and then invoking a **JSP** to show the next web page containing the data returned from the program.

First you have to give the interaction a **name** so you can reference it later on.

You will see the following dialog



Web Interaction Wizard

Specify a Name and Location for your Web Interaction

Define the name and location for the communication between Web pages and the chosen business process. The location must be the webApplication directory inside a project.

Destination folder: /labxx/webApplication

Web Interaction name: custinquiry

Java package:

□ Use error page

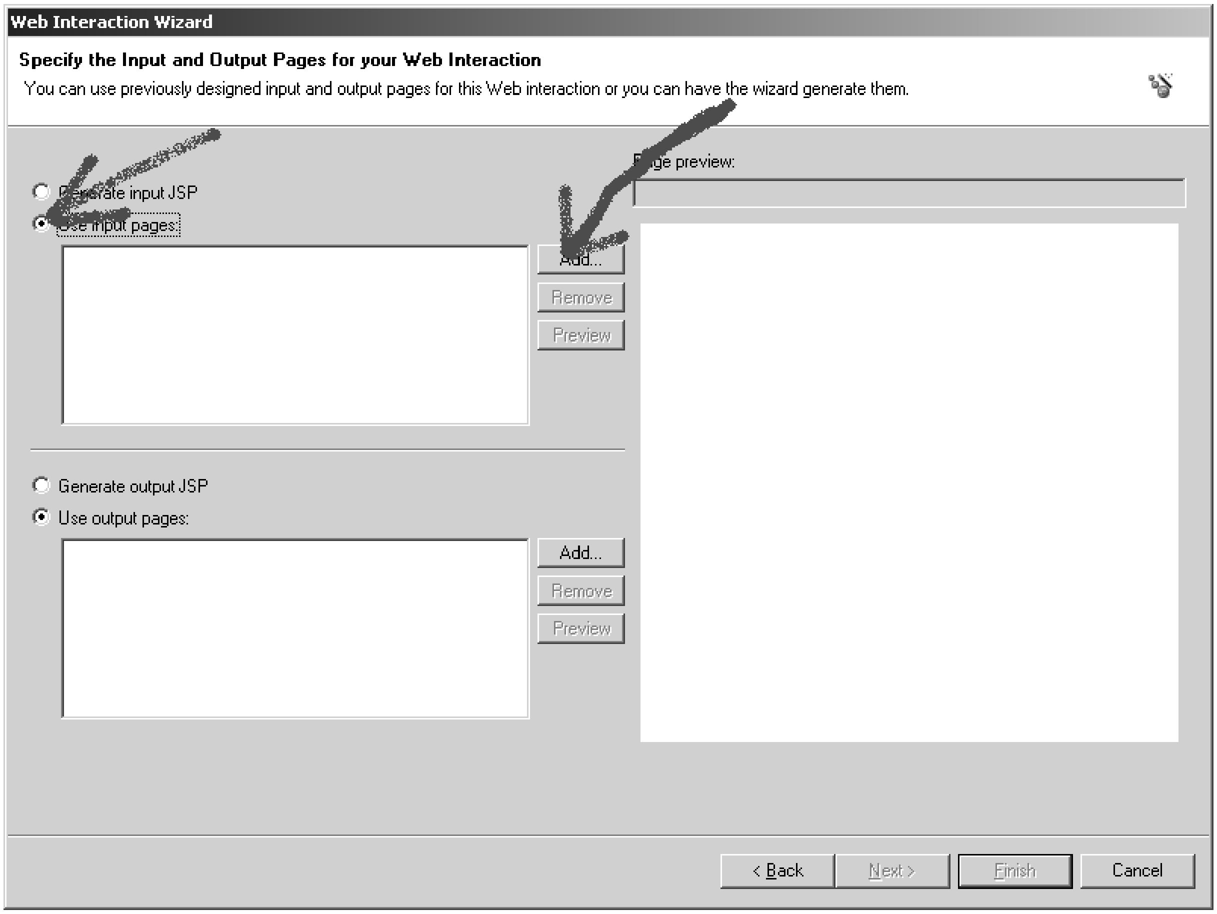
□ Invalidate session after the interaction occurs

Servlet URL: custinquiryServlet?custno=#1&Inquiry=#2

- Press the **Browse..** push button, beside the **Destination folder** entry field
- Select the **webApplication** folder under your project folder **labxx** as your destination folder
- Key in the interaction name: ***custinquiry***
- <u>**Don't**</u> check the check boxes
- **Press** the **Next** > push button

Defining the web pages to work with in this interaction

The next page of the interaction wizard asks for information about the web pages that are involved in this interaction. You have an **input page** that invokes the interaction and an **output page** that completes the interaction it will be shown in the browser at the end of the interaction. Your web page from the **last exercise** will be the input page.



Web Interaction Wizard

Specify the Input and Output Pages for your Web Interaction

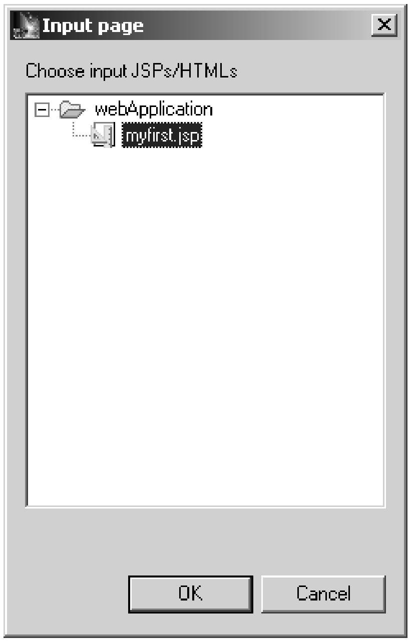
You can use previously designed input and output pages for this Web interaction or you can have the wizard generate them.

○ Generate input JSP
● Use input pages:

○ Generate output JSP
● Use output pages:

- Check the Radio button for: ***<u>Use input page</u>***
- Press the **Add** push button beside the **upper** list box

This brings up a list of pages that can be used in this project, since you only have one page created in this project, only **one page** is shown in this list.

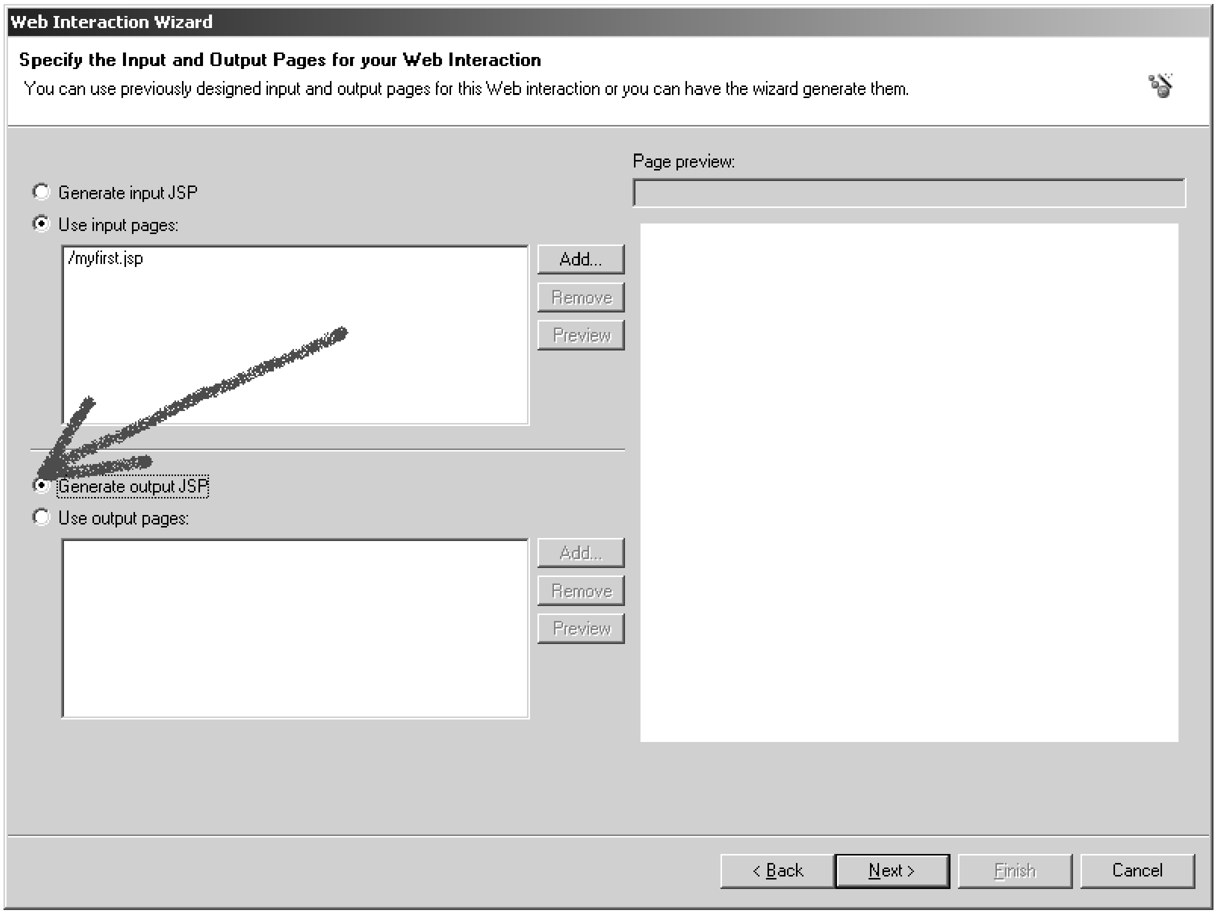


- Expand the **webApplication** folder
- Select your **myfirst.jsp** file, as the input page to be used in this interaction
- Press the **OK** push button

Specifying the output page

You don’t have a result page to be shown when this interaction is completed, so the wizard gives you a choice to generate a page for you. Compare this to **DFU**, where you can select database fields to work with and DFU creates the user interface for you.

Still on the same wizard page:



Web Interaction Wizard

Specify the Input and Output Pages for your Web Interaction

You can use previously designed input and output pages for this Web interaction or you can have the wizard generate them.

○ Generate input JSP

● Use input pages:

/myfirst.jsp

● Generate output JSP

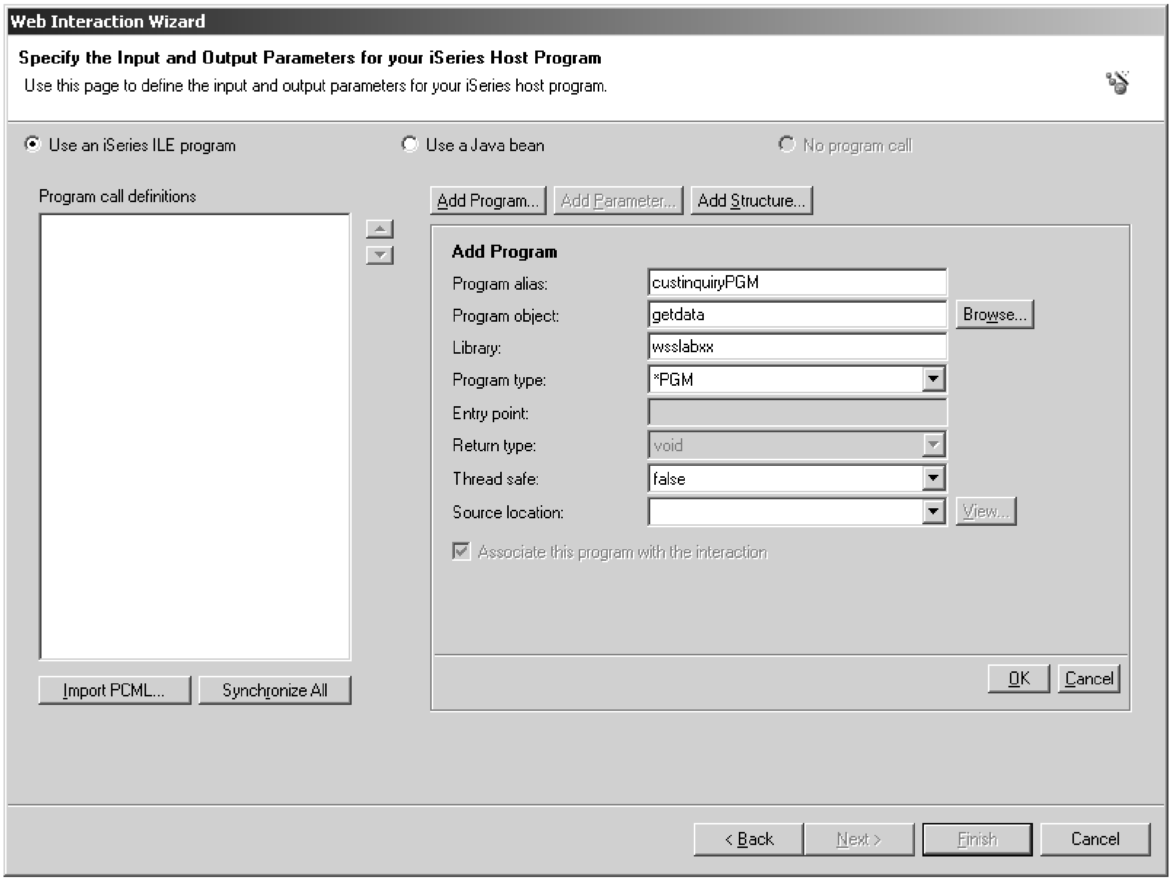
○ Use output pages:

Page preview:

- Check the radio button ***Generate output JSP*** to indicate that the wizard should generate the output (result) web page.
- Press the **Next >** push button

Defining the iSeries program invocation and parameters

The next page will ask you to define the parameters you want to pass between the Web environment and the iSeries program that serves the requests from the web.



Web Interaction Wizard

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

Program alias:	custinquiryPGM
Program object:	getdata
Library:	wsslabxx
Program type:	*PGM
Entry point:	
Return type:	void
Thread safe:	false
Source location:	

☑ Associate this program with the interaction

First define the program that you want to call on the iSeries

Fill in the entry fields in the **right** side of the dialog

- Enter name **custinquiryPGM** in the **Program alias** entry field
- Enter the name **getdata** in the **Program object** entry field
- **Enter** the **library name *WSSLABxx***

xx being your group number

Now you have to make a choice for entry field **program type**

If you feel **comfortable** in working with **Service programs** and **procedures** in Service programs

- **Then follow the steps** on the next page

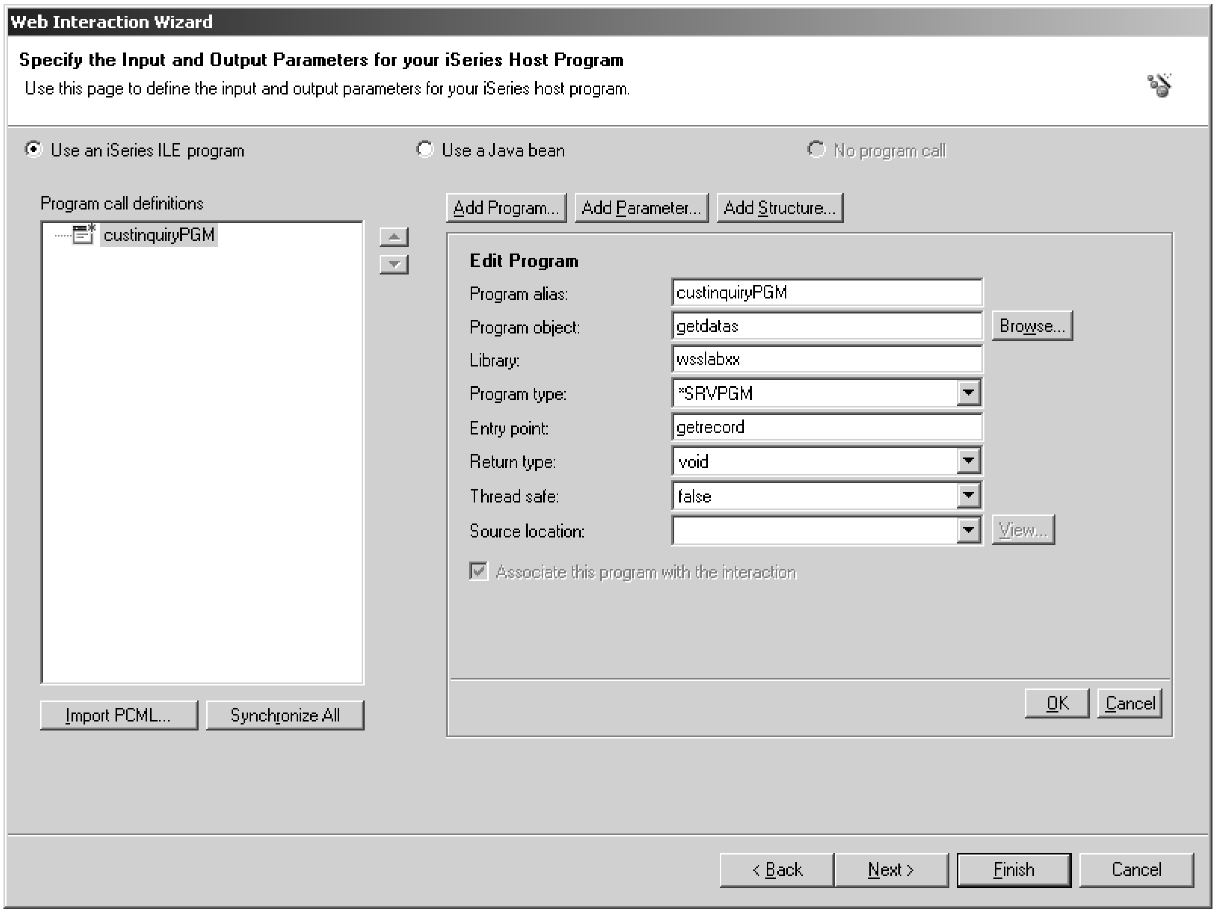
If you would **rather** work with a **regular program**

- Then leave the **Program Type** entry field as is ***PGM**
- Press the **Add program** push button near the top of the dialog
- Press the **Yes** push button when asked to save
- Skip to heading **Proceed here ..** **Approximately 3 pages ahead**.

Extra steps for Service program invocation

<u>Instructions for groups that want to work with procedures in service programs,</u>

<u>if you want to invoke a regular *PGM from your web page just skip to heading *Proceed here..........*</u>



Web Interaction Wizard

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

- custinquiryPGM

Edit Program

Program alias: custinquiryPGM

Program object: getdatas

Library: wsslabxx

Program type: *SRVPGM

Entry point: getrecord

Return type: void

Thread safe: false

Source location:

☑ Associate this program with the interaction

The Service program you call will have a different name, so first change the program name:

- Select the **Program object** entry field

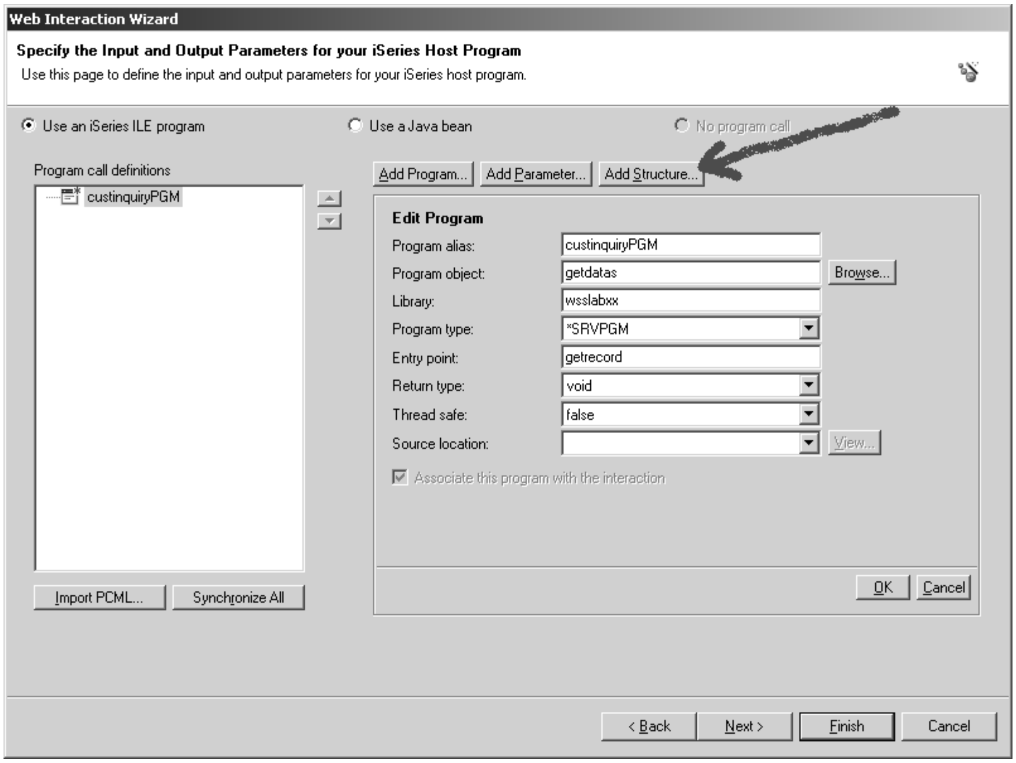
- Change it to ***getdatas***
- In the **Program type** combo box select ***SRVPGM** from the drop down list
- Enter **entry point** name ***getrecord***, this is the procedure name in your service program
- Press the **Add program** push button near the top of this dialog
- Press the **Yes** push button when asked to save

The special instructions for calling Service programs end here.

Proceed here:

_for both *PGM and *SRVPGM invocation

Defining the parameter information



Web Interaction Wizard

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

- custinquiryPGM

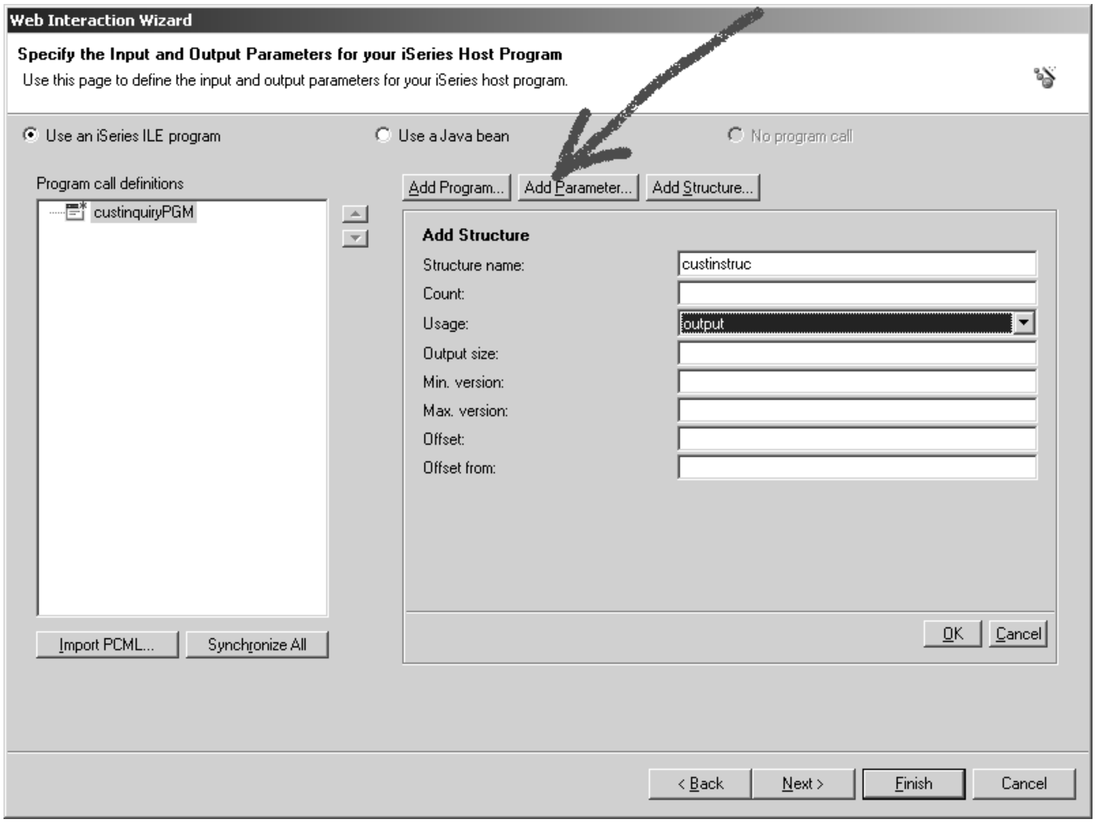
Edit Program

Program alias:	custinquiryPGM
Program object:	getdatas
Library:	wsslabxx
Program type:	*SRVPGM
Entry point:	getrecord
Return type:	void
Thread safe:	false
Source location:	

☑ Associate this program with the interaction

- Press the **Add structure** push button

A new page will appear in the dialog



Web Interaction Wizard

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: custinquiryPGM

Add Structure

Structure name:	custinstruc
Count:	
Usage:	output
Output size:	
Min. version:	
Max. version:	
Offset:	
Offset from:	

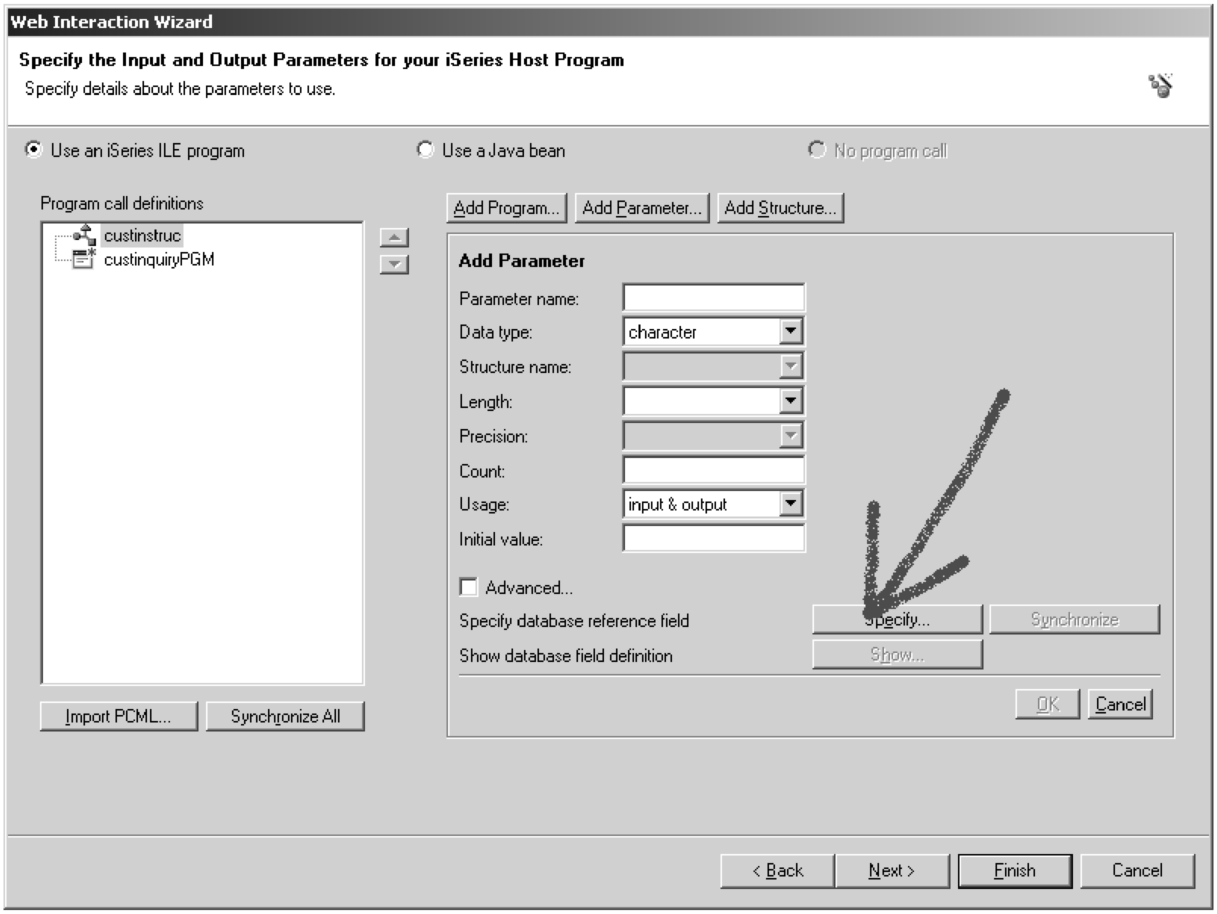
- Enter **custinstruc** as the structure name
- Change the **usage** combo box to **output**

This will tell the wizard that the fields in the structure will only be used to transfer data from the program to the user interface.

- Now press the **Add parameter...** push button on top of the group box
- Press the **Yes** push button when you get asked to Save changes

Add fields to structure

A new dialog will show in the group box of this wizard page, it allows to specify parameters



Web Interaction Wizard

Specify the Input and Output Parameters for your iSeries Host Program

Specify details about the parameters to use.

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

Program call definitions

- custinstruc
- custinquiryPGM

Add Parameter

Parameter name:	
Data type:	character
Structure name:	
Length:	
Precision:	
Count:	
Usage:	input & output
Initial value:	

□ Advanced...

Specify database reference field

Show database field definition

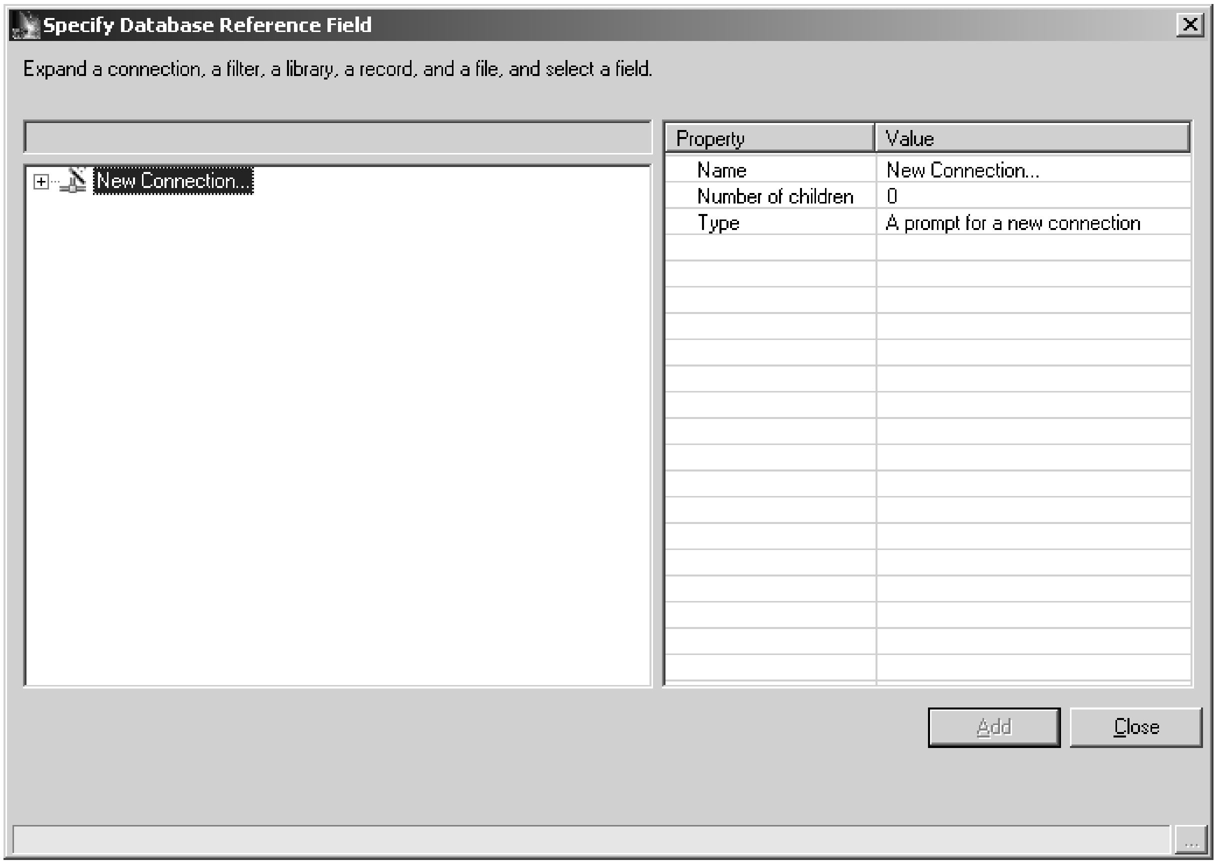
.

You will use a record format on the iSeries to describe the layout of the structure so you will need to connect to the iSeries server to select the record format.

To get the dialog to connect to the iSeries

- Press the **Specify....** push button

The specify reference field dialog shows up



Specify Database Reference Field

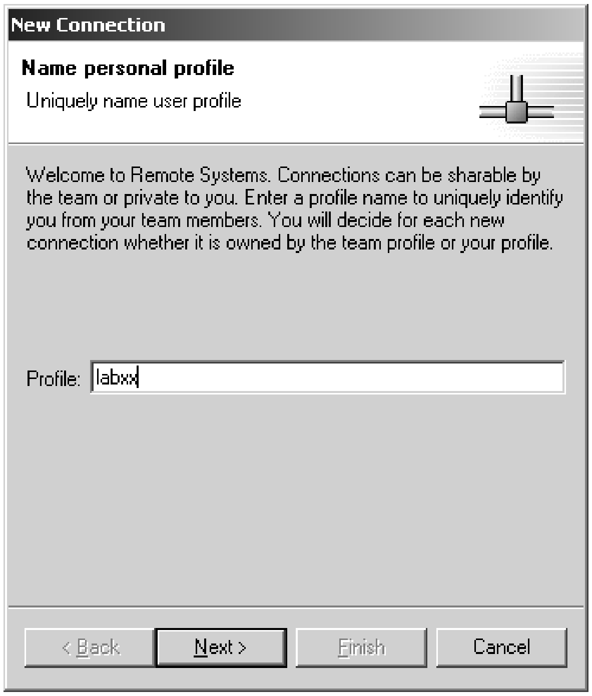
Expand a connection, a filter, a library, a record, and a file, and select a field.

Property	Value
Name	New Connection...
Number of children	0
Type	A prompt for a new connection

In the list on the left hand side

- Expand the **New Connection...** list item by clicking on the ✚ beside its icon

- The new Connection dialog shows



New Connection

Name personal profile

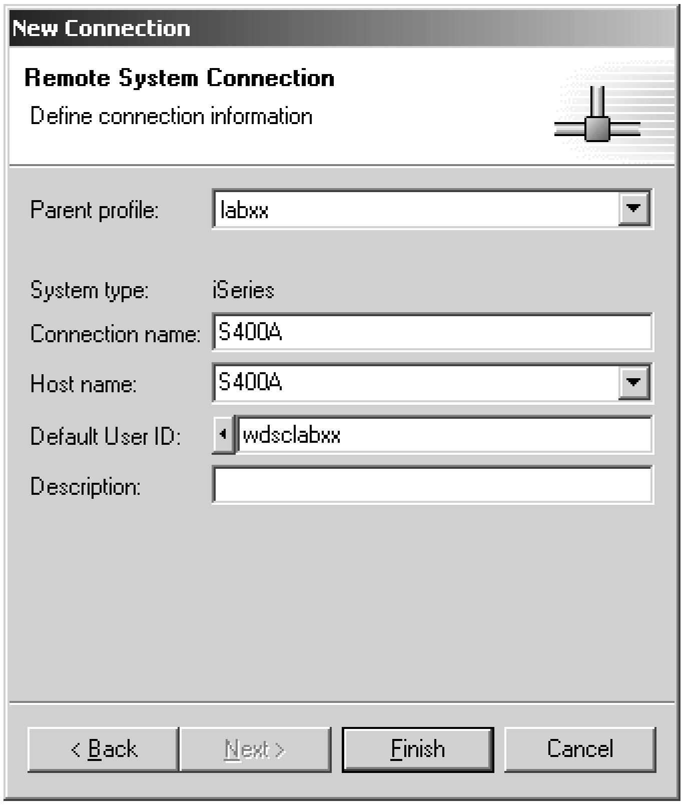
Uniquely name user profile

Welcome to Remote Systems. Connections can be sharable by the team or private to you. Enter a profile name to uniquely identify you from your team members. You will decide for each new connection whether it is owned by the team profile or your profile.

Profile: labxx

- Specify **labxx** in the **Profile** entry field
- Press the **Next** > push button

The second page of the new connection dialog shows



Parent profile: labxx

System type: iSeries

Connection name: S400A

Host name: S400A

Default User ID: wdsclabxx

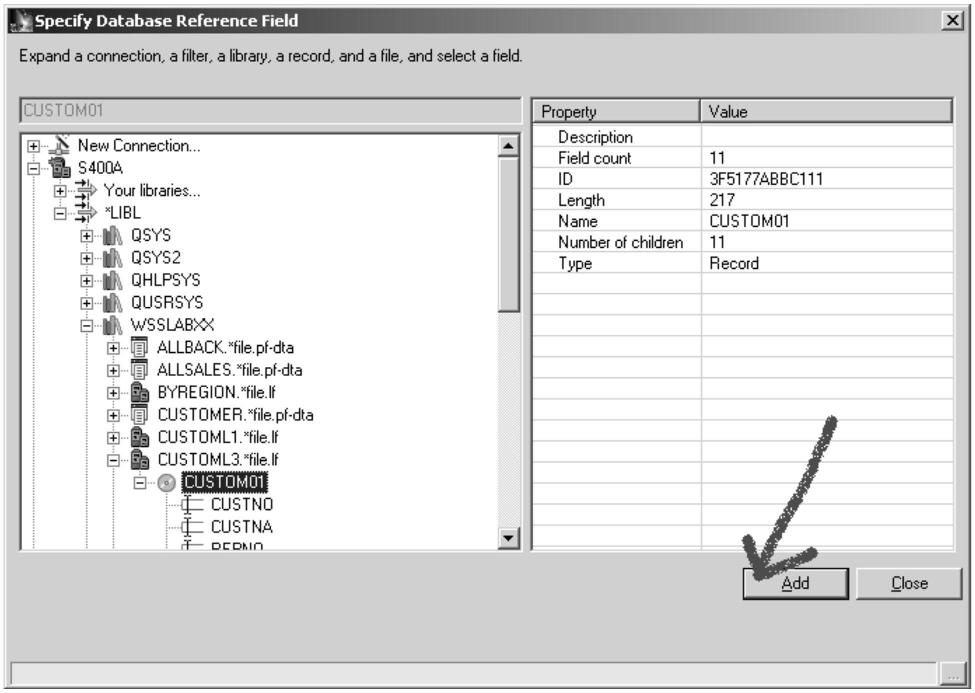
Description:

- Enter the connection name, same as your **(Host)server name**
- In the **Host name** entry field, enter your iSeries server name
- In the **Default User ID** entry field, enter your team userid   **WDSCLABxx**
- Press the **Finish** push button

You are now back on the Specify Database Reference fields dialog.

Back on the *Specify Database Reference fields* dialog

You need to expand a couple of nodes in the connection tree



Specify Database Reference Field

Expand a connection, a filter, a library, a record, and a file, and select a field.

CUSTOM01

Property	Value
Description	
Field count	11
ID	3F5177ABBC111
Length	217
Name	CUSTOM01
Number of children	11
Type	Record

If you get a **Signon dialog** during these steps use:

Userid: WDSCLAB**xx** , **xx** being your team number

Password:WDSCLAB**xx** , **xx** being your team number

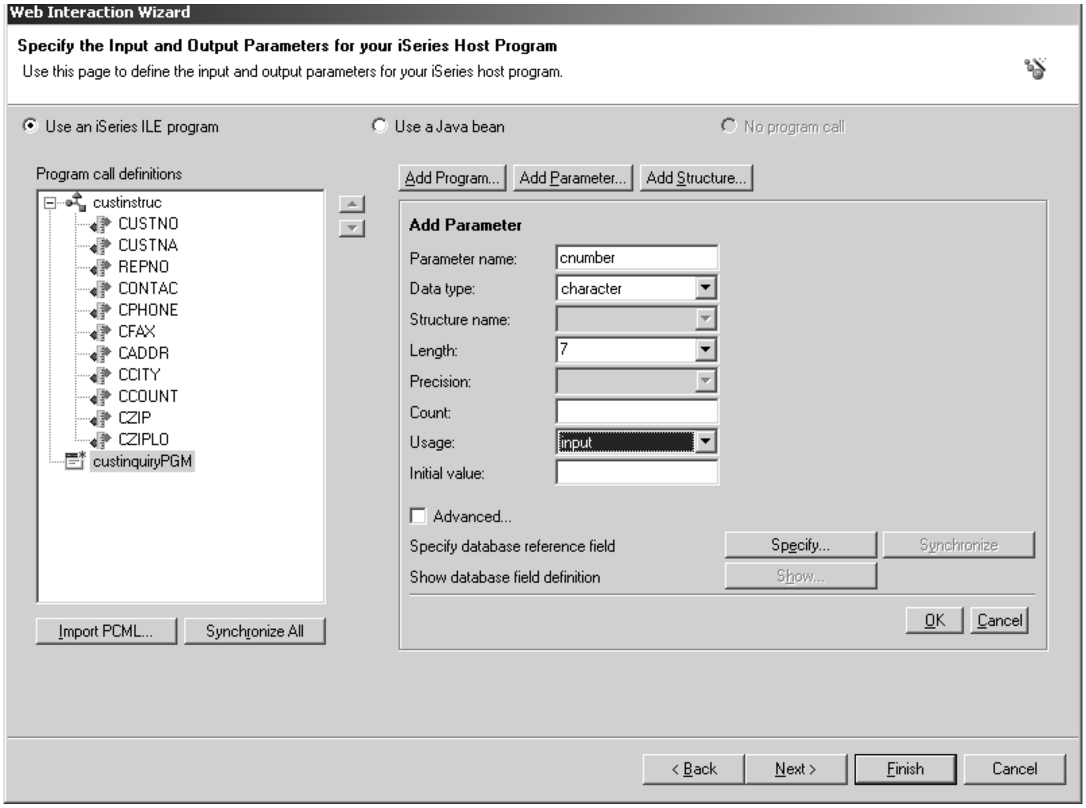
- Expand your **server node,** by pressing the ✚ beside the icon
- Expand the ***LIBL node** in the expanded branch, by pressing the ✚
- Expand the library **WSSLABxx node** by pressing the ✚ beside the icon
- Expand the logical file **CUSTOML3 node** by pressing the ✚ beside the icon
- Expand your record format **CUSTOM01 node** , by pressing the ✚ beside the icon
- Select the record format **CUSTOM01 node** so it is highlighted
- Press the **Add** push button

You should see a message on the bottom of the dialog that fields have been added

- Press the **Close** push button

Adding parameters to the program

Back on the wizard page



Web Interaction Wizard

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

- custinstruc
  - CUSTNO
  - CUSTNA
  - REPNO
  - CONTAC
  - CPHONE
  - CFAX
  - CADDR
  - CCITY
  - CCOUNT
  - CZIP
  - CZIPLO
- custinquiryPGM

Add Parameter

Parameter name:	cnumber
Data type:	character
Structure name:	
Length:	7
Precision:	
Count:	
Usage:	input
Initial value:	

□ Advanced...

- Select the **program icon** of the program you added 2 steps ago, in the list at the left hand side of the dialog
- Press the **Add parameter** pushbutton, near the top of the dialog
- Enter **cnumber** in the **Parameter name** entry field
- Enter **7** in the **Length** entry field
- Select **input** in the **Usage** combo box

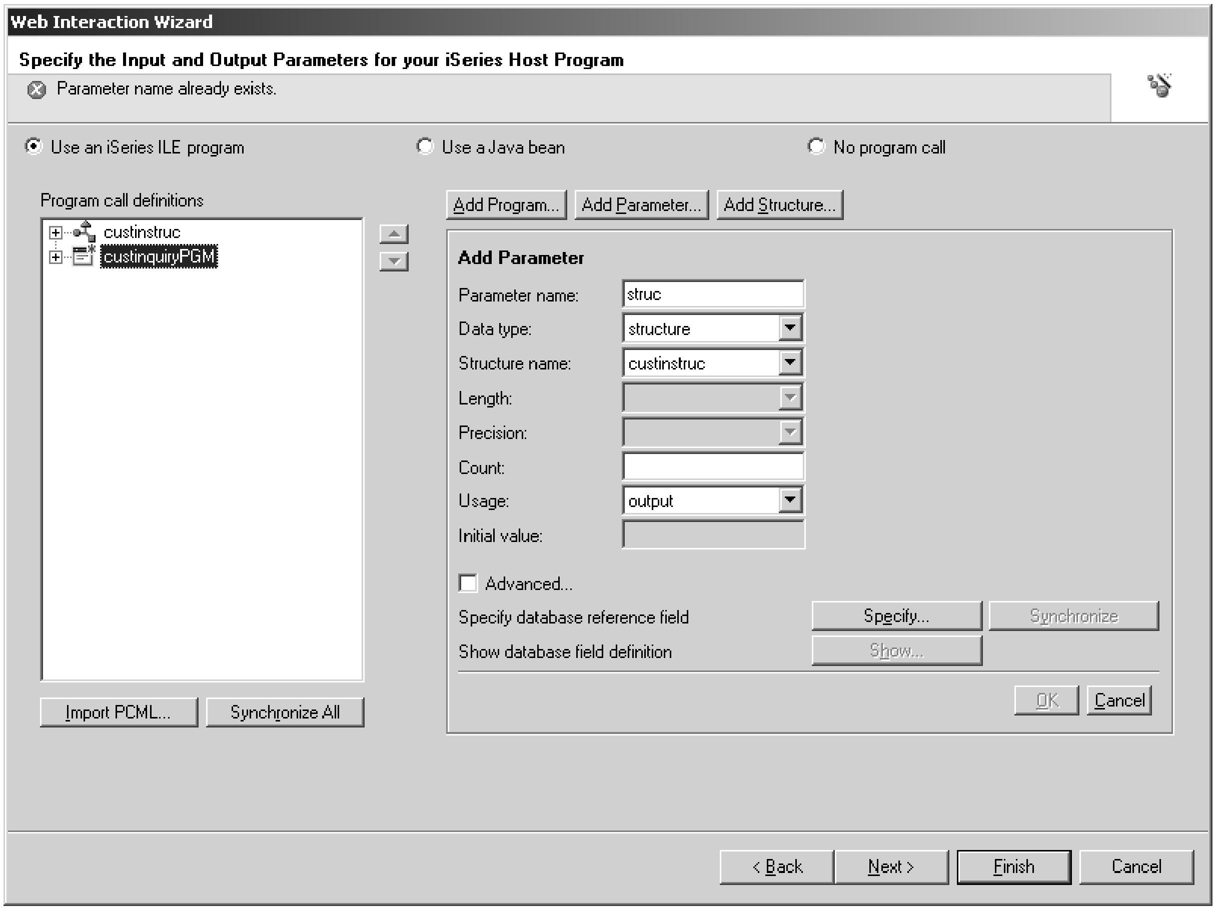
  This is the Input parameter that contains the customer number that will be send from your web page to the program
- Press the **Ok** push button on the right hand side of the dialog

If getting asked whether you want to, press the **Yes** push button

You defined the first parameter

Defining the structure as a parameter

Now you will add the structure as a parameter to the program call definition



Web Interaction Wizard

Specify the Input and Output Parameters for your iSeries Host Program

Parameter name already exists.

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

Program call definitions

- custinstruc
- custinquiryPGM

Add Parameter

Parameter name: struc

Data type: structure

Structure name: custinstruc

Length:

Precision:

Count:

Usage: output

Initial value:

□ Advanced...

Specify database reference field

Show database field definition

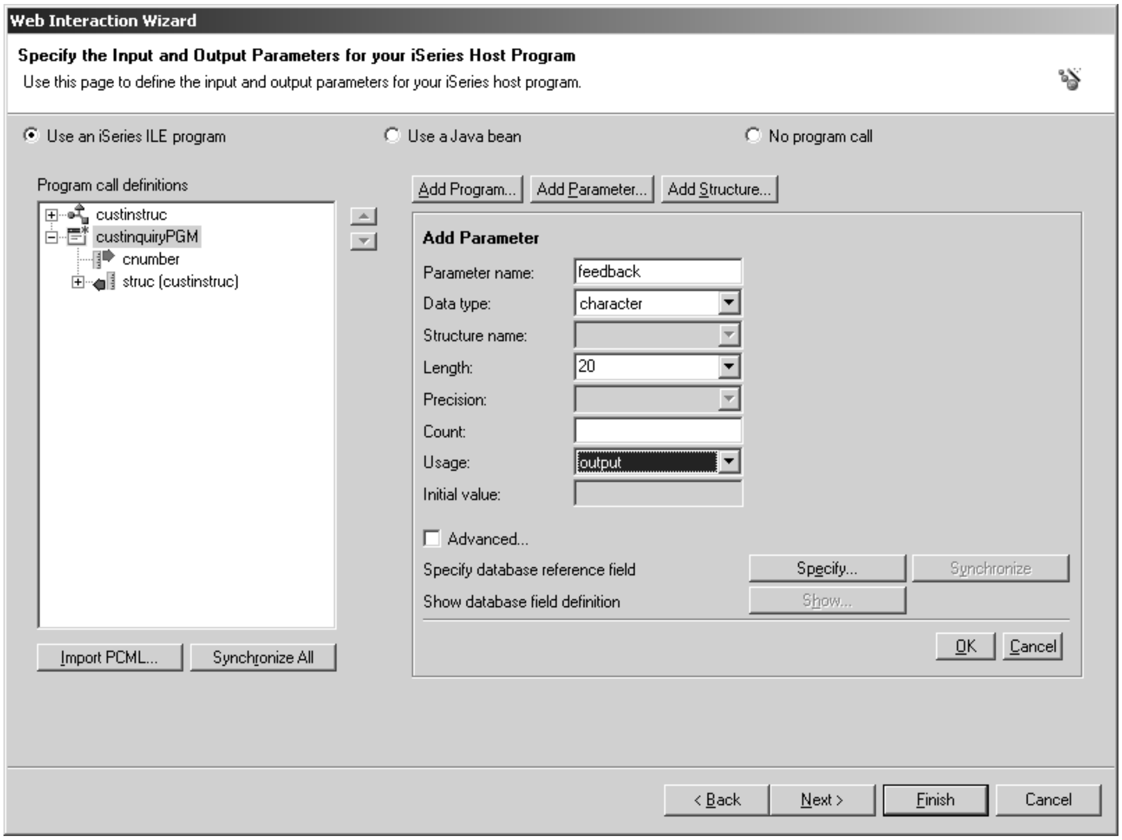
- Press the **Add Parameter** push button on top of the group box
- Key in the Parameter name entry field: **struc**
- Select **structure** in the Datatype combo box
- Select **output** in the Usage combo box
- Press the **OK** push button in the middle of the dialog

If getting asked, whether you want to save, press the **Yes** push button

Your structure is now an output parameter for the program call, meaning it will pass data back from the program to the servlet running in WAS.

Adding a feedback parameter

The program on the iSeries is written in a way that it will return a value to indicate whether the action (accessing a customer record) on iSeries was successful or not. If it was not successful the return value will contain a message number and a value for a substitution variable. You need to add a third parameter to the program call.



Web Interaction Wizard

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

- custinstruc
- custinquiryPGM
  - cnumber
  - struc (custinstruc)

Add Parameter

Parameter name: feedback
Data type: character
Structure name:
Length: 20
Precision:
Count:
Usage: output
Initial value:

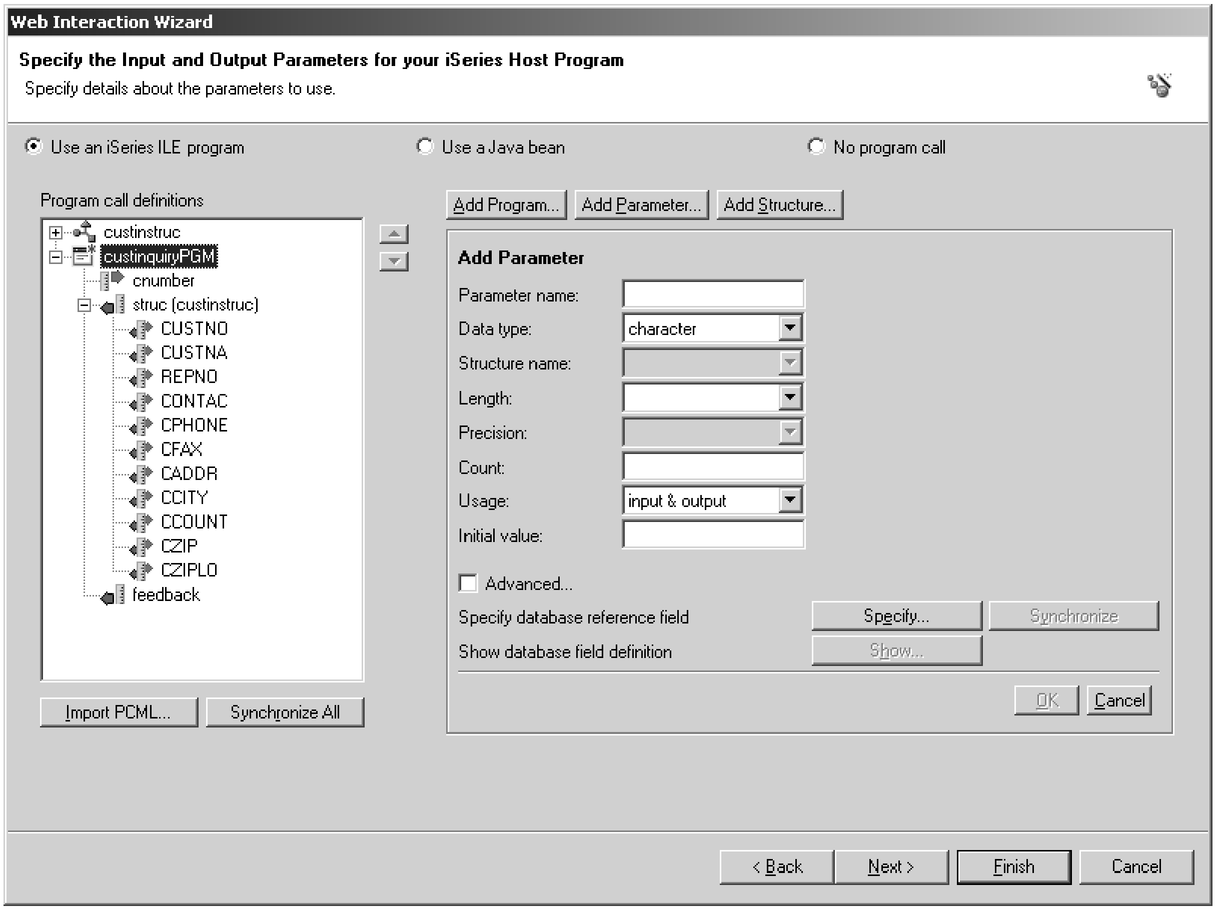
□ Advanced...

Specify database reference field

Show database field definition

- Press the **Add parameter** push button
- Enter **feedback** as a name in the **Parameter name** entry field
- Enter **20** into the **Length** entry field
- Select **output** in the **Usage** combo box
- Press the **OK** push button

Your are done all parameters have been specified



Web Interaction Wizard

Specify the Input and Output Parameters for your iSeries Host Program

Specify details about the parameters to use.

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

Program call definitions

```
custinstruc
custinquiryPGM
    cnumber
    struc (custinstruc)
        CUSTNO
        CUSTNA
        REPNO
        CONTAC
        CPHONE
        CFAX
        CADDR
        CCITY
        CCOUNT
        CZIP
        CZIPLO
    feedback
```

Add Parameter

Parameter name:

Data type: character

Structure name:

Length:

Precision:

Count:

Usage: input & output

Initial value:

☐ Advanced...

Specify database reference field

Show database field definition

You can expand the Program icon in the list on the left side this dialog to get a detailed look at the parameter definitions.

Ready to go to the next page

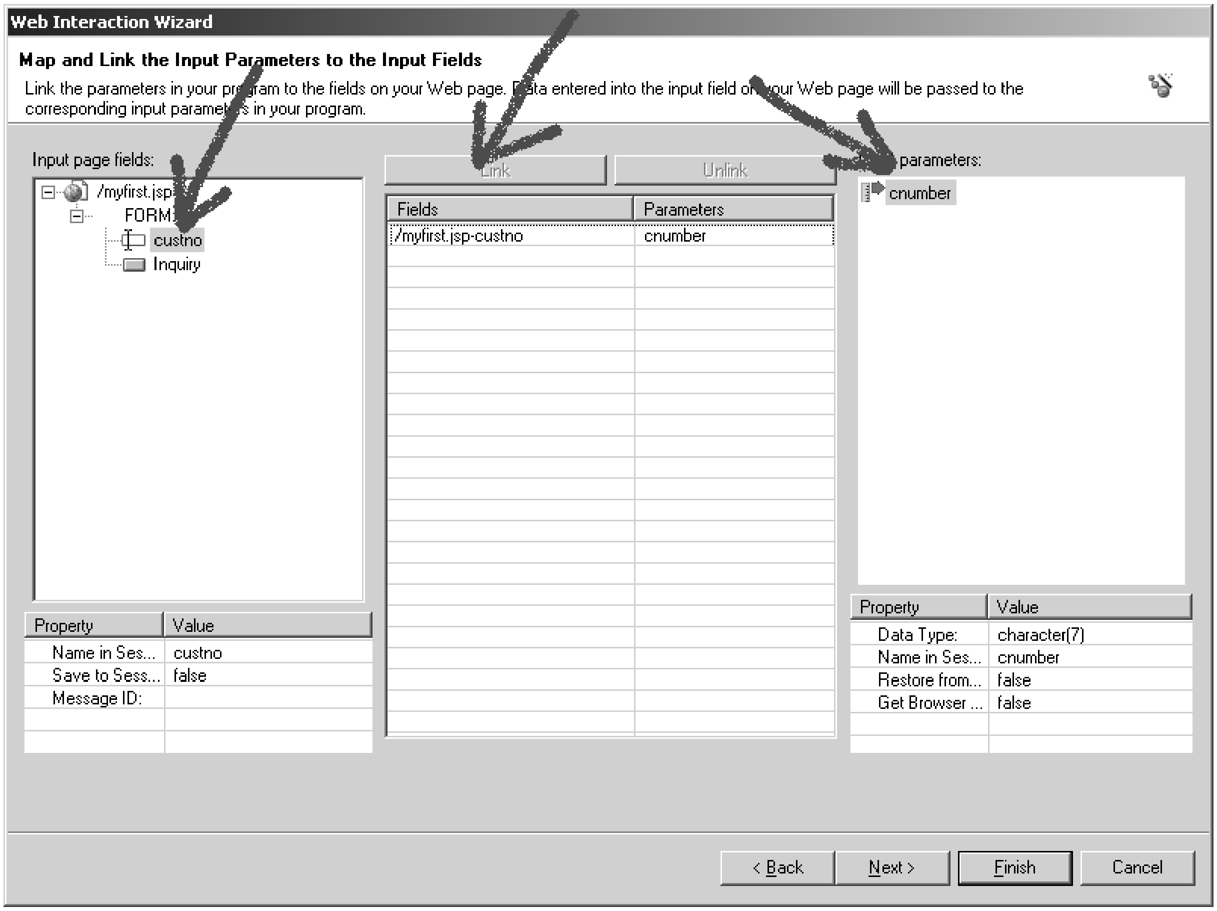
- Press the **Next >** push button at the bottom of the wizard page

The interface between the **servlet** and the **iSeries program** that you will use is now **defined.**

Handle input parameters

Now you need to specify which fields on your Input web page get mapped to the input parameters of the program call, or in other words which parameter field does get the data from your input page.

This is easy in this application since you only have the customer number input field. Also if the field on the web page and the parameter field have the same name they get mapped automatically by the wizard. In this example we named them differently so you need to do the mapping per hand. Here is the wizard page.



Web Interaction Wizard

Map and Link the Input Parameters to the Input Fields

Link the parameters in your program to the fields on your Web page. Data entered into the input field on your Web page will be passed to the corresponding input parameters in your program.

Input page fields:

- /myfirst.jsp
  - FORM:
    - custno
    - Inquiry

Property	Value
Name in Ses...	custno
Save to Sess...	false
Message ID:	

Fields	Parameters
./myfirst.jsp-custno	cnumber

Input parameters:

- cnumber

Property	Value
Data Type:	character(7)
Name in Ses...	cnumber
Restore from...	false
Get Browser ...	false

- Expand the **myfirst.jsp** node in the left hand list
- Expand the **FORM:** node under the **myfirst.jsp** node

This list shows all input capable controls in your jsp, like input fields in a DDS record format.

- Select the **custno** field in the expanded tree in the left list

  In the list on the right hand side you only have one choice, since this is the only input parameter defined for your program call. If you have more choices in your right hand list you didn’t select the correct usage on the previous page, but don't worry it will not matter when you run the program.

- Select the **cnumber** field in the right list

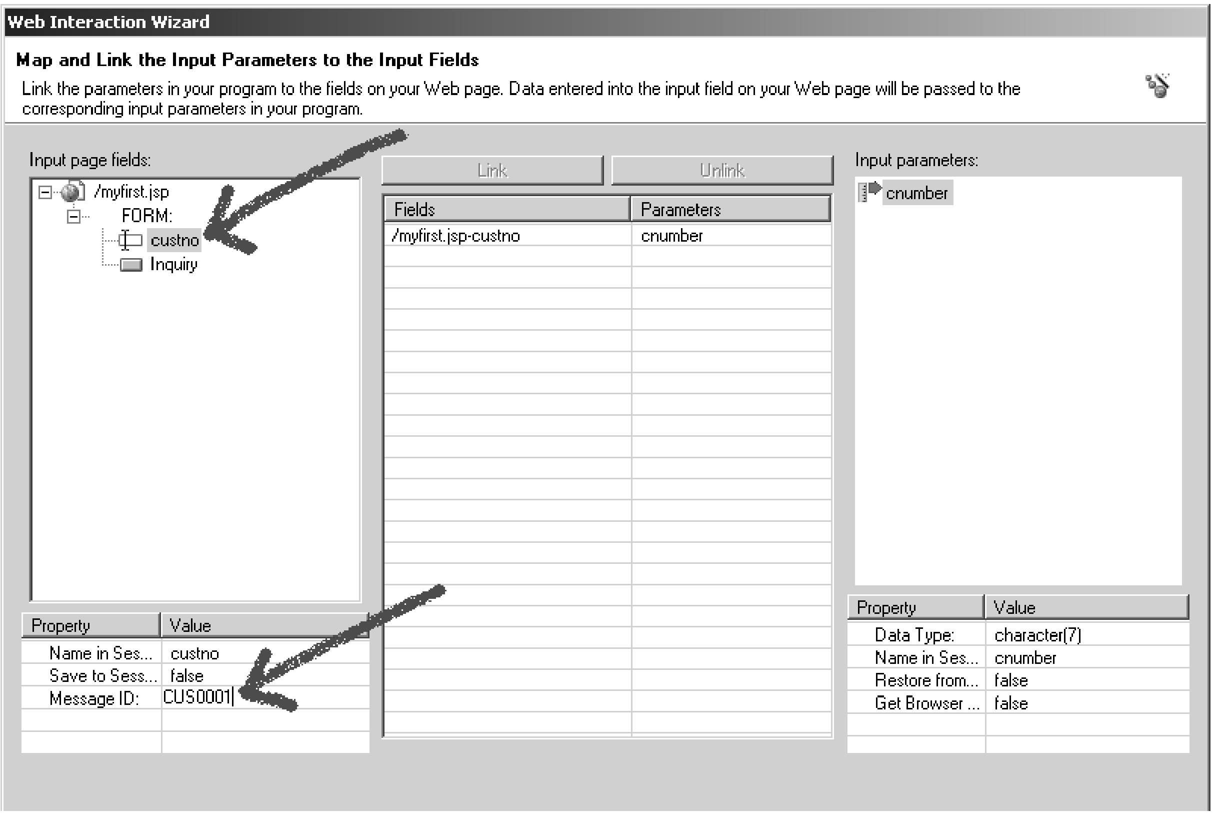
- Press the **Link** push button on top of the dialog

You will see both fields are now listed in the middle table, this table shows the linked input fields.

We also want to be able to tell the end user if there is a problem with the input data he submitted. If the program fails to find a customer record for the customer number that got keyed into the Input page, the program will send back a message id in the feedback parameter. This error message text contains a substitution variable. We want to fill this substitution variable with the data from the custno input field. To enable the wizard to generate the code for this function we need to tell it that variable **custno** contains data for the substitution variable in our message.

You will do this on this wizard page as well

Identifying variable to be used in message substitution



Web Interaction Wizard

Map and Link the Input Parameters to the Input Fields

Link the parameters in your program to the fields on your Web page. Data entered into the input field on your Web page will be passed to the corresponding input parameters in your program.

Fields	Parameters
/myfirst.jsp-custno	cnumber

Property	Value
Name in Ses...	custno
Save to Sess...	false
Message ID:	CUS0001

Property	Value
Data Type:	character(7)
Name in Ses...	cnumber
Restore from...	false
Get Browser ...	false

- Select the **custno** field in the list on the left side of the wizard page

You will see the properties window at the bottom, below the list, will be filled with the properties for field **custno**. In this properties pane:

- Select the **entry field** beside the **Message ID:** heading and enter **CUS0001**

- **Important make sure to press the Enter key after making this entry**

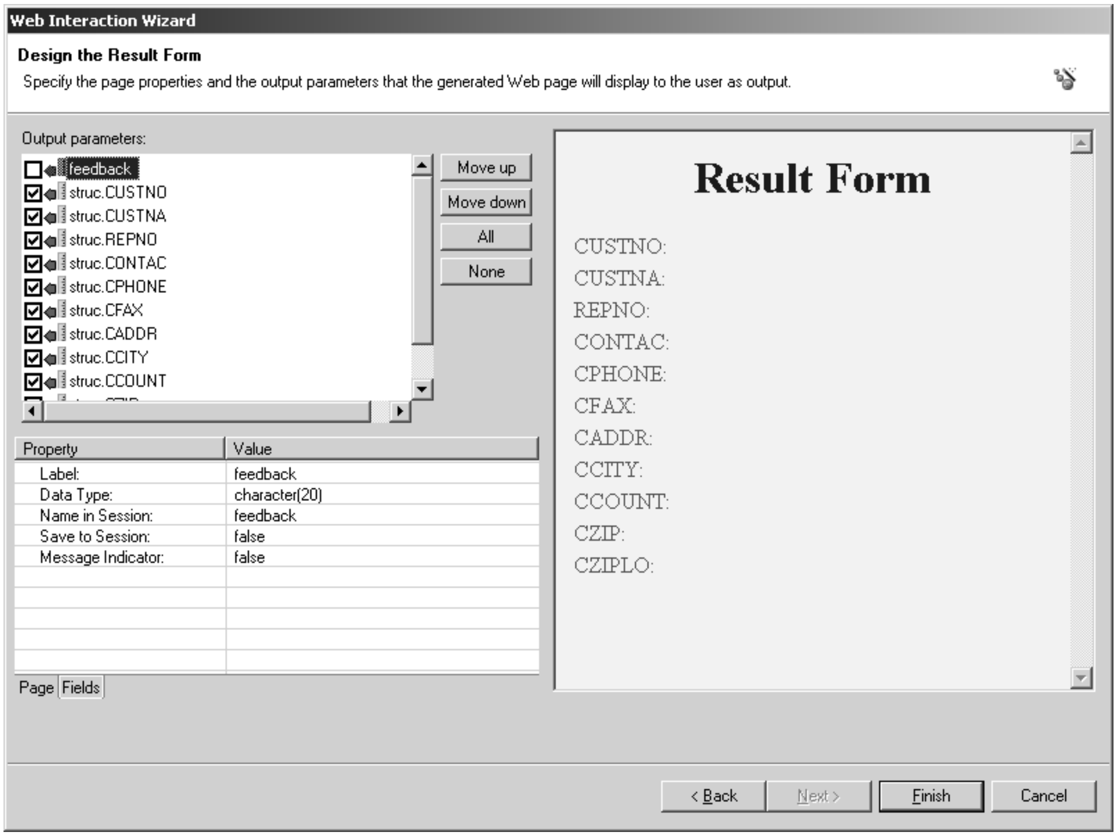
CUS0001 is the message id that contains a substitution variable that will be filled with the content of **custno** in case of an exception during the database retrieval

- Press the **Next >** push button to advance to the next wizard page

Defining the Output page content

Now you can specify what data should be displayed on the web page generated for you. On the right hand side you see a sample of the page that will be generated for you.

By default all output parameters will be added to the web page, but we don't want the data in the **feedback** parameter to be displayed. Lets go through the steps to erase the feedback parameter from the output page.



Web Interaction Wizard

Design the Result Form

Specify the page properties and the output parameters that the generated Web page will display to the user as output.

Output parameters:

- □ feedback
- ☑ struc.CUSTNO
- ☑ struc.CUSTNA
- ☑ struc.REPNO
- ☑ struc.CONTAC
- ☑ struc.CPHONE
- ☑ struc.CFAX
- ☑ struc.CADDR
- ☑ struc.CCITY
- ☑ struc.CCOUNT

Property	Value
Label:	feedback
Data Type:	character(20)
Name in Session:	feedback
Save to Session:	false
Message Indicator:	false

Result Form

CUSTNO:
CUSTNA:
REPNO:
CONTAC:
CPHONE:
CFAX:
CADDR:
CCITY:
CCOUNT:
CZIP:
CZIPLO:

- Uncheck the check box beside the **feedback** parameter in the left hand list.

  This will remove it from the web page

You will stay on this page to specify more error handling

Specifying error handling

Now you can specify what should happen in case a customer number didn't match the customer numbers in our data base.

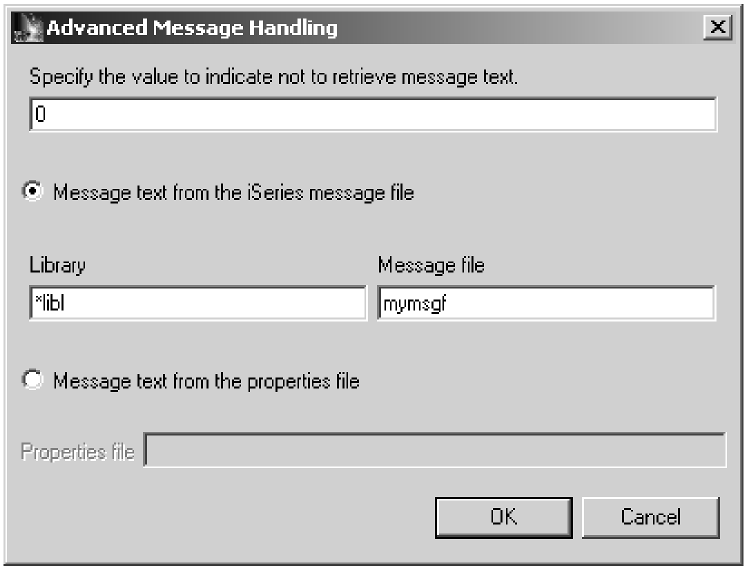
- Select the **feedback** variable in the left hand list

You will see the properties for this variable in the window below the left hand list.

- Locate the **Message Indicator** heading in this properties table
- Click in the field beside the **Message Indicator** and set it to true

  This just tells the wizard that this parameter will contain an indication whether the program call was successful or.

  A dialog as in the figure below will show up .



Specify the value to indicate not to retrieve message text.
0

● Message text from the iSeries message file

Library: *libl
Message file: mymsgf

○ Message text from the properties file

Properties file

- Specify **0** in the upper entry field in the Advanced Message handling dialog
- Check the radio button **Message text from the iSeries message file**

- Enter ***LIBL** in the **Library** entry field
- Enter **MYMSGF** in the **Message file** entry field
- Press the **OK** push button on this dialog
- Press the **Finish** push button on the **Design the Result Form** page in the wizard dialog

Creating the Interaction

The interaction wizard will now create the necessary files for your application

- Press the **Yes To all** push button, when you get a warning message that some files already exist and will be overwritten.

Before you can test your web application you have to create the iSeries program containing the business logic to return data to your result page.

You can write the logic with SEU if you wish to. These Lab instructions will guide you through using the **CODE** environment to **edit** and **compile** your iSeries program.

Adding the iSeries program to your Web Application

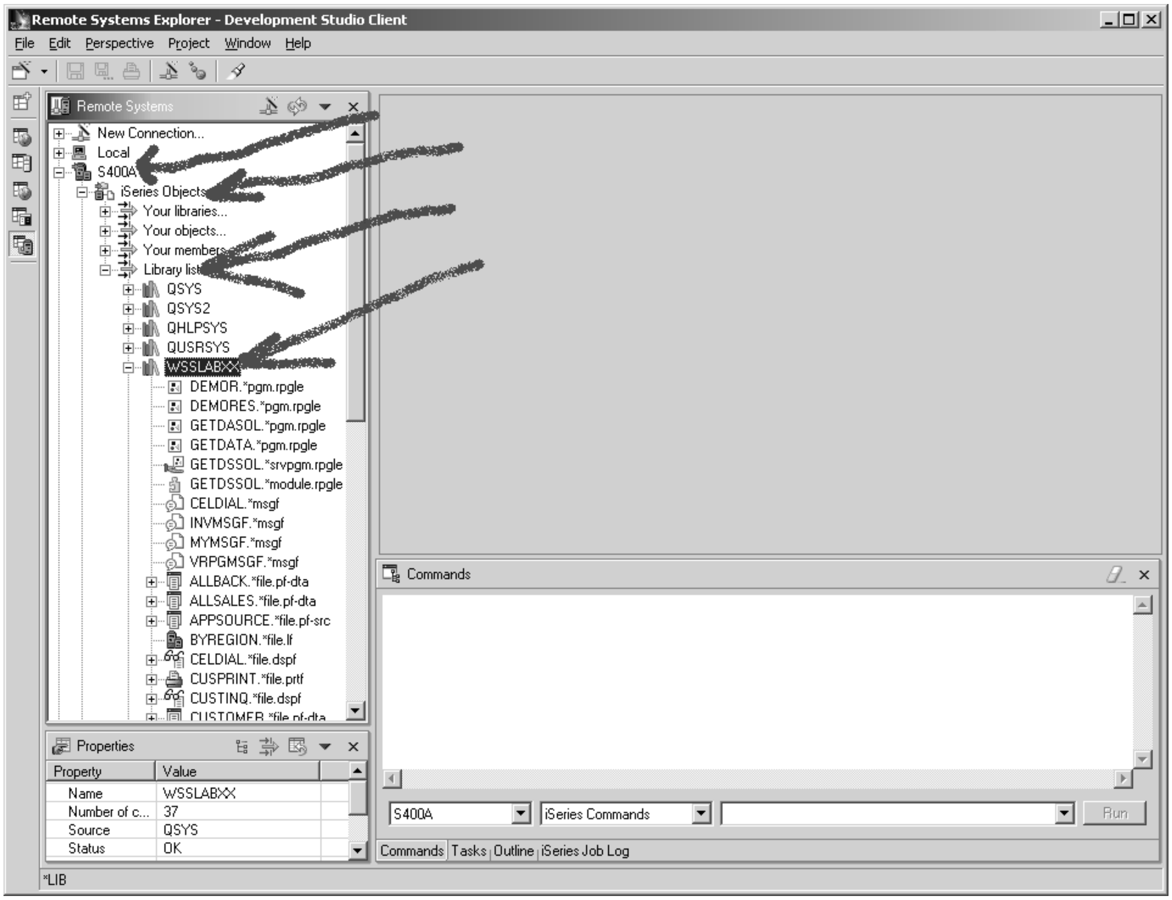
Invocation of the **CODE** editor can be accomplished in a couple of ways, the traditional way before the **WDSc** workbench was available, would have been through the Windows **Start menu**. Now since you are already working inside the WDSc workbench you can use the **Remote Explorer Perspective** (the perspective for iSeries programmers) to access the program source member inside the workbench.

The **Remote Explorer Perspective** provides a **PDM like** environment but with a GUI interface.

In the workbench

Select the ***Perspective --> Open ---> Remote System Explorer*** menu options

That will open the Remote Explorer



Property	Value
Name	WSSLABXX
Number of c...	37
Source	QSYS
Status	OK

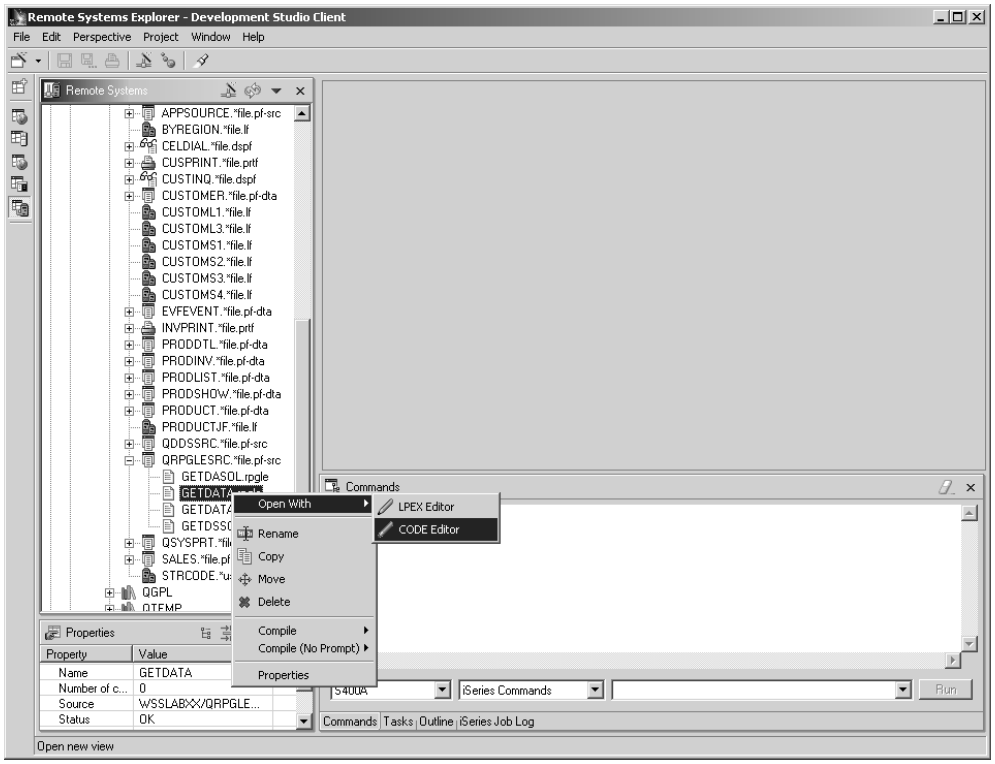
You will see a list of servers in the Remote Systems list box.

- Expand **your iSeries server** icon, use the same server that you have used to specify the reference fields in the data structure
- Expand the **iSeries Objects** icon
- Expand the **Library List** Icon in the expanded iSeries Objects branch

  Normally you would see a UserId/Password dialog being displayed, but since you already used the reference field dialog this connection is already active and no new sign on to the iSeries is needed

- Expand the library **WSSLABxx**

You will see all objects in this library,



Properties

Property	Value
Name	GETDATA
Number of c...	0
Source	WSSLABXX/QRPGLE...
Status	OK

- Scroll down to the **source files** in the expanded list

We have prepared 2 shell source members for you.

Depending on whether you specified that you wanted to work with a **program,** or a **procedure** in a **service program,** you need to select a different member:

For the program call:

- Right mouse click on member **GETDATA**

For the service program call

- Right mouse click on member **GETDATAS**

- Select **Open with** from the pop up menu

- Select **CODE editor** from the sub menu

That brings up the **editor window** and you are ready to write your program.

For the 2 different choices we have given you, writing a regular **RPGIV** program, or an **RPGIV Service** Program, you need to go to the correct section in this script.

Before you do that, if you have not worked with the CODE editor you might want to check out the section: **Some Code Editor Features** just after this paragraph.

Then proceed to the correct section in this hand out,

They are titled:

-- Creating the *PGM object using RPGIV

-- Creating the *SRVPGM using RPGIV

The next section explains some CODE editor features.

If you are already familiar with CODE skip the section on CODE editor features and go directly to your topic of choice.

Some CODE editor Features

Before we start, here are some useful tips on using the **CODE** workstation editor

Let's look at some of the features of the CODE editor, so you can later easily find your way around and utilize them.

- **ALT+L** will mark a line
- **ALT+U** will unmark a line
- **ALT+D** will delete a marked line
- **ALT+C** will copy a marked line
- **ALT+M** will move a marked line Press Enter to insert a new line

Token Highlighting

The CODE editor highlights the **Tokens** (specification fields) of your RPG program source, providing a separate color for each improving readability.

When you make changes to a line, the token colors updated only after you move your cursor off the line.

To see how token highlighting works:

1. Move the cursor to the **Calculation** statement you just created to **set on LR**.
2. Position the cursor to **column 7** (right next to the 'C').
3. Type an asterisk (*****).
4. Move the cursor off the line and watch what happens. The line where the '*****' was typed has become a comment line and its color changes accordingly.
5. Move the cursor back to column 7 and **remove** the '*****'. Move the cursor off that line of code and the statement is tokenized.
6. The token highlighting changes to reflect that this is a non-commented '**C**' specification.

Displaying Types of Lines

Using the CODE editor. it is possible to have only particular types of source lines displayed at a time. To do this:

1. Choose ***View→Show*** from the CODE editor menu bar. The options available on the pull down menu lists the types of line selections that can be made.

2. Choose ***Comments***. The CODE editor now contains only those RPG statements that are comments.

3. Choose ***View→ Show all***. All statements types are displayed.

In addition, the choices in the **View** pull-down can be used to include only control specifications, user subroutines, and action subroutines. The **Include** choice will allow the selection of only those lines containing a particular character string. As the CODE editor is fully-programmable, it is possible to create macros to include/exclude all types of source statements, based on specific criteria.

Syntax Checking

Syntax checking is available for RPG code, and by default will be active. The syntax of the RPG code is checked automatically when a change is made to a line, and the cursor is moved off the line.

When errors are found, they are displayed following the statement with the error.

Using Format Lines

The format line is at the top of the CODE editor window, just above the first statement. A format line is used to help keep track of the columns in a particular specification line. The content of the format line can vary to reflect the particular type of specification being keyed such as **F** specs, **C** specs, **D** specs and so on.

To display a format line:

Make a line active by positioning the cursor to it by clicking on the line with the left mouse button, or by using the arrow keys and clicking the left mouse button. The format line gets updated as you leave a line, to refresh the format line without leaving the current statement

- Choose ***View→Refresh format line***.
- The format line changes to reflect an RPG specification, since the cursor was on a C-specification when the format line was requested.

Tip *The format line can also be refreshed by pressing* ***Ctrl+R****. with the cursor on a statement.*

- You can move the cursor right or left with the arrow keys to go from character to character, or with the Tab key to go from field to field. An indicator on the format line moves with the cursor to show in which column the cursor is positioned.
- To select a format line for any specification you want, choose the ***View*** menu bar choice, then choose ***Select format line***. The **RPG Format Line Selection** window appears

Now you can started and create the RPG **program** or **service program**.

Creating a *PGM object with RPGIV

To make life easier for you we already prepared most of the **RPGIV** source for you.

You have already opened the correct source member **GETDATA** in the previous step using the RSE object list.

- Go through the source which is pretty simple

It contains an **F spec** to access file **CUSTOML3** which contains the customer data sorted by customer number

The **D spec**s defines the data structure **CSTRUC** that you pass back to your web page and the ***CUSTNOI*** **variable** that gets passed from the web page to this program.

As well the **Return** variable is defined as a 20 length character field.

The ***Entry** parameter list is defined as the

- **CUSTNOI** field (this is the input parameter)
- **CSTRUC** structure (this the data structure for the customer data)
- **RETURN** field to indicate no success for file access

The code you have to write here fetches a customer record by chaining to the file with the **CUSTNOI** that gets passed into the program

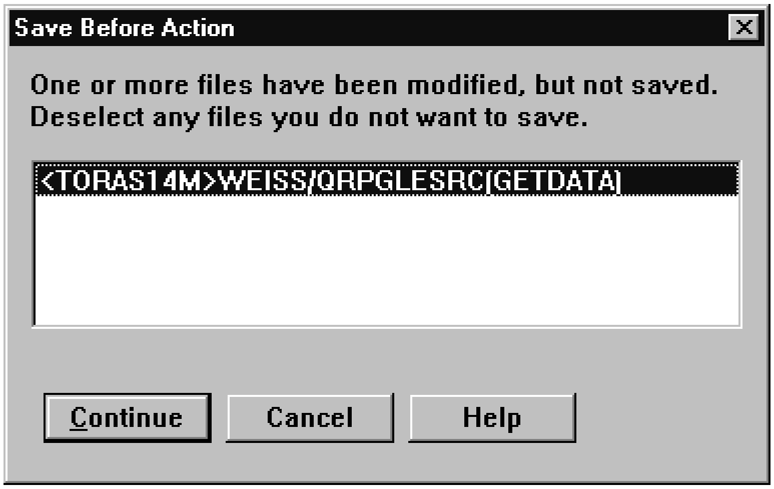
Add the code to get the customer record and check whether the Chain was successful

```
CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq
C     CUSTNOI       CHAIN     CUSTOML3                           5050
C
C                   If        *IN50
C                   EVAL      RETURN=’CUS0001‘ + CUSTNOI
C                   ELSE
C                   EVAL      RETURN=’0’
C                   ENDIF
```

Done with coding so far, **lets Create** the program

- Select ***Actions* --> *Compile* --> *Prompt*** from the Edit window menu bar

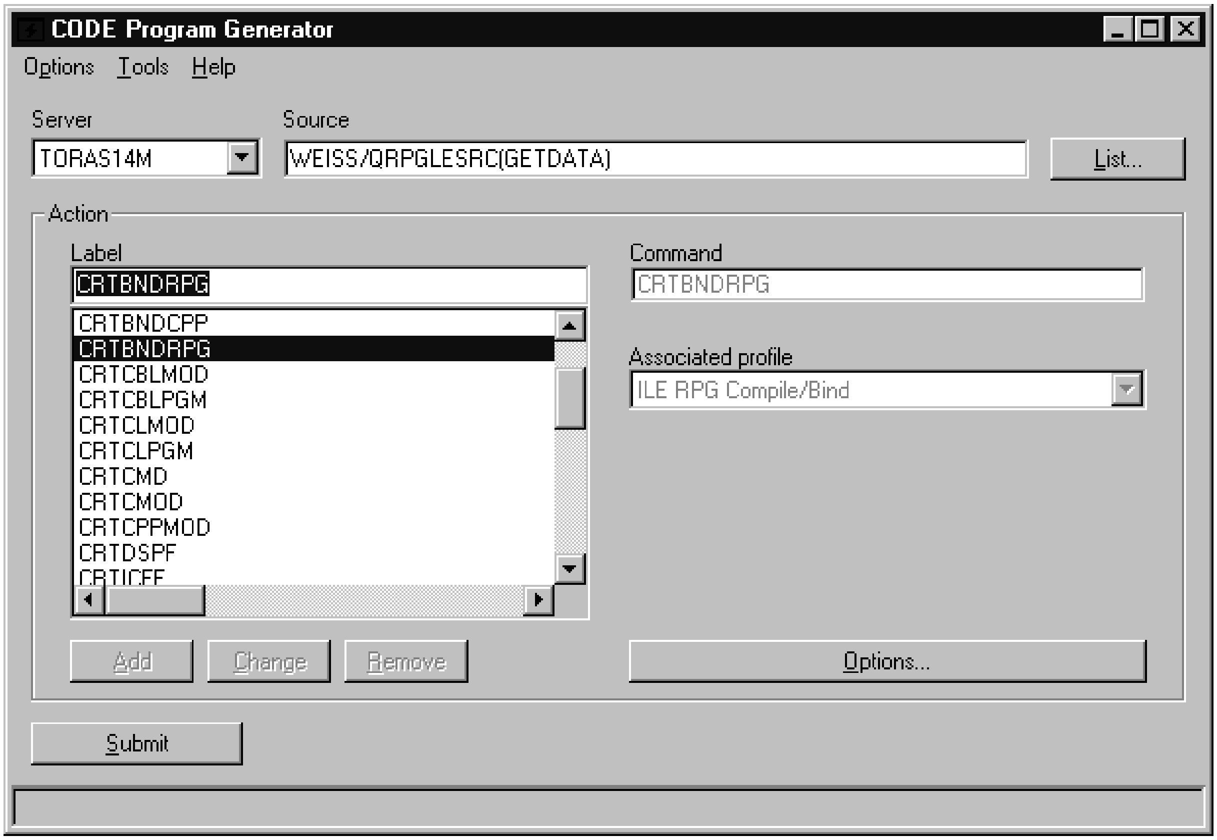
If you haven’t saved before you will get a dialog asking you whether you want to save



One or more files have been modified, but not saved.
Deselect any files you do not want to save.

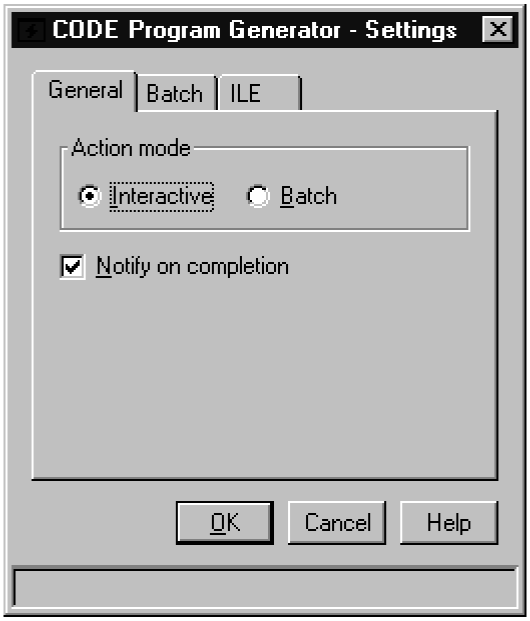
<TORAS14M>WEISS/QRPGLESRC(GETDATA)

Press the **Continue** push button



In the **CODE Program Generator** dialog,

- Select **CRTBNDRPG** from the list of choices
- Select **the Options** --> ***Settings* menu option** from the CODE Program Generator dialog



CODE Program Generator - Settings

Action mode

● Interactive ○ Batch

☑ Notify on completion

- Select the **Interactive** Radio button
- Press the **Ok** push button

On the **CODE Program Generator dialog**

- Press the **Submit** push button

If you are lucky you get a message that states your **compile has been completed** and the program object has been created.

Now you are **ready to run** your web application.

Proceed to the section

Setting up the iSeries server information

and skip the creating a ***SRVPGM** part.

If you get a message that there have been **errors found** in your program,

- **go through the edit compile fix cycle**.

Edit compile fix cycle

you will see a dialog that asks you whether to proceed with the command window.

- Press the **No** pushbutton from this dialog

An error list window will appear, **check the errors by:**

- Double clicking **on the error**

That will position you to the Edit window to the statement that is wrong

Fix all errors until the error list only contains error messages with a **check mark**

- Select **Actions --> Compile -- no prompt** from the edit window menu bar

The CODE editor saves the compile options for each member, so you don't have to re-specify that you want to use the CRTBNDRPG command.

- **Continue this cycle until you get a clean compile**
- Then proceed to

Setting up the iSeries server information

Creating a *SRVPGM with RPGIV

To make life easier for you we already prepared most of the **RPGIV** source for you.

You have already opened the correct source member **GETDATAS** in the previous step using the RSE object list.

- **Read** through the source which is pretty simple

It contains an **F spec** to access file **CUSTOML3** which contains the customer data sorted by customer number

The **D spec**s define the data structure **CSTRUC** that gets passed back to your web page and the **CUSTNOI** variable that gets passed from the web page to this program.

Also variable **RETURN** is defined as 20 character, to return message id and customer number in case the customer record was not found.

The prototype for procedure getrecord with the parameters defined as the

- **CUSTNOI** field (this is the input parameter)
- **CSTRUC** structure (this is the data structure for customer data)
- **RETURN** field (returns error or success)

> **Note**: The external name is case sensitive, this has to exactly match what **you specified in the Inter Action wizard**

The procedure interface defines the parameters the same way the prototype did.

Inside the procedure after the procedure interface you need to add the RPG code

You have to add the following code

```
CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq
C     CUSTNOI       CHAIN     CUSTOML3                           5050
C
C                   If        not *IN50
C                   EVAL      cust1=cstruc
C                   EVAL      RETURN=’0’
C                   ELSE
C                   EVAL      RETURN=’CUS0001 ‘ + CUSTNOI
C                   ENDIF
```

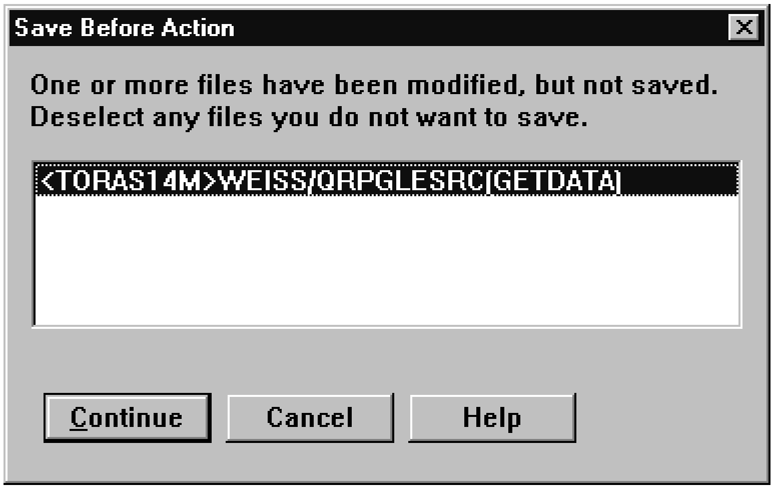
You do a CHAIN to the data base and move the data base fields to the local procedure structure. If the database access fails move the MSGID and Customer number to the RETURN variable.

Create an RPG module

Done with coding so far, lets Create the program

- Select **Actions --> Compile --> Prompt** from the Edit window menu bar

If you haven’t Saved before you will get a dialog asking you whether you want to save

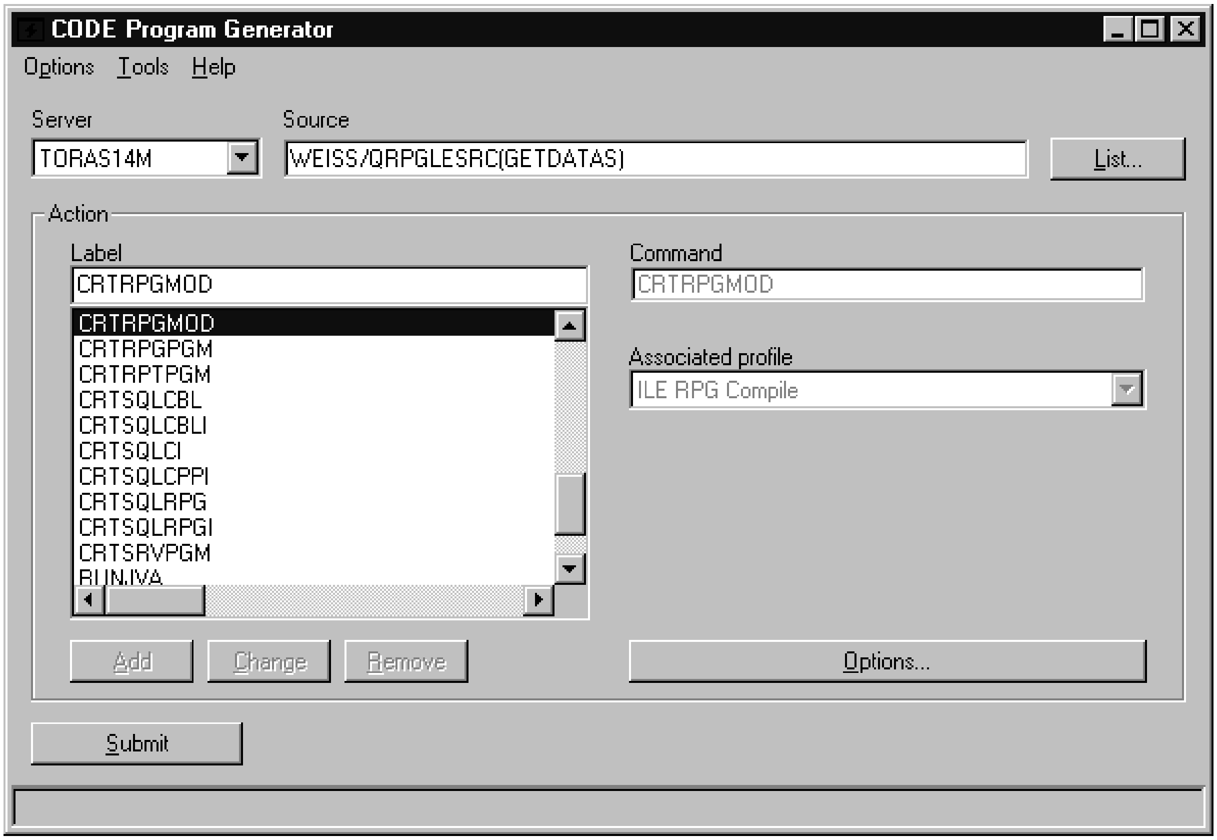


Save Before Action

One or more files have been modified, but not saved.
Deselect any files you do not want to save.

<TORAS14M>WEISS/QRPGLESRC(GETDATA)

- Press the **Continue** pushbutton



Server: TORAS14M

Source: WEISS/QRPGLESRC(GETDATAS)

Action

Label: CRTRPGMOD

CRTRPGMOD
CRTRPGPGM
CRTRPTPGM
CRTSQLCBL
CRTSQLCBLI
CRTSQLCI
CRTSQLCPPI
CRTSQLRPG
CRTSQLRPGI
CRTSRVPGM
RUNJVA

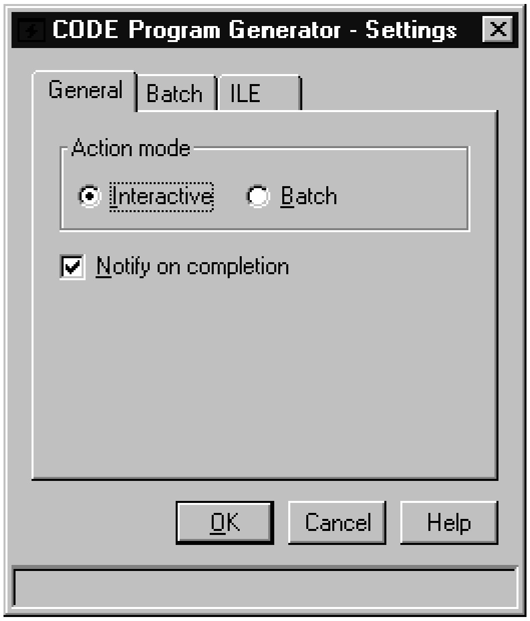
Command: CRTRPGMOD

Associated profile: ILE RPG Compile

In the CODE Program Generator dialog,

Select **CRTRPGMOD** from the list of choices

- Select the **Options --> Settings** menu option from the CODE Program Generator **dialog menu bar**, don't use the **Options pushbutton** at this time



Action mode

● Interactive ○ Batch

☑ Notify on completion

- Select the **Interactive Radio button**
- Press the **Ok** pushbutton

On the **CODE Program Generator dialog**

- Press the **Submit** push button

If you are **lucky** you get a message that states that your **compile has been completed** and the program object has been created.

Now you are ready to **create the service program** for your web application go the section Create service program.

If you get a message that there have been **errors found** in your program, go through the edit compile fix cycle.

Edit compile fix cycle

you will see a dialog that asks you whether to proceed with the command window.

- Press the **No** push button from this dialog

An error list window will appear, check the errors by:

- Double clicking on the error

That will position you to the Edit window to the statement that is wrong

Fix all errors until the error list only contains error messages with a check mark on the side, errors that have been fixed get a check mark in the error list

The CODE editor saves the compile options for each member, so you **don’t** have to re-specify that you want to use the **CRTRPGMOD** command.

Go **back up** to **Edit Compile fix cycle**

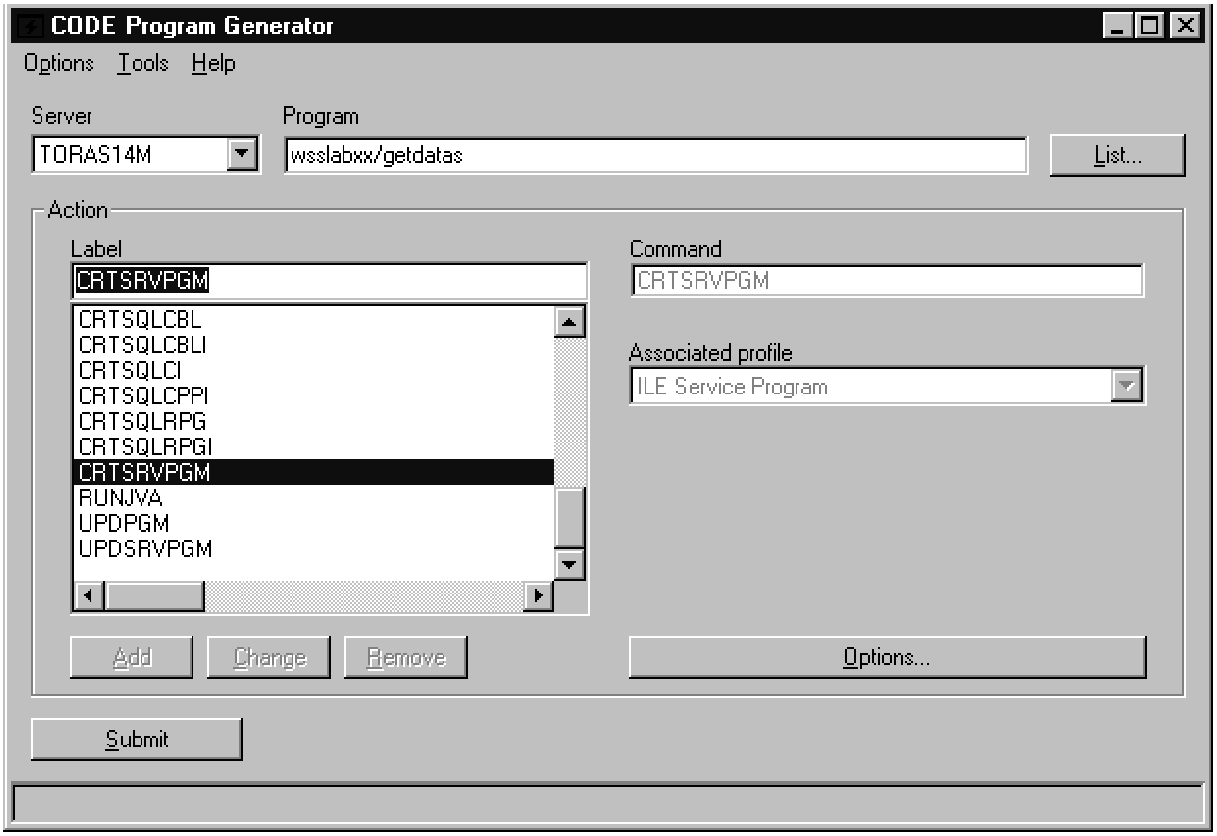
Continue this cycle until you get a clean compile

- then proceed **down to** the next step

Create a service program

Create a Service program by

- Select **Actions --> Compile -- no prompt** from the edit window menu bar .



Server: TORAS14M

Program: wsslabxx/getdatas

Action

Label: CRTSRVPGM

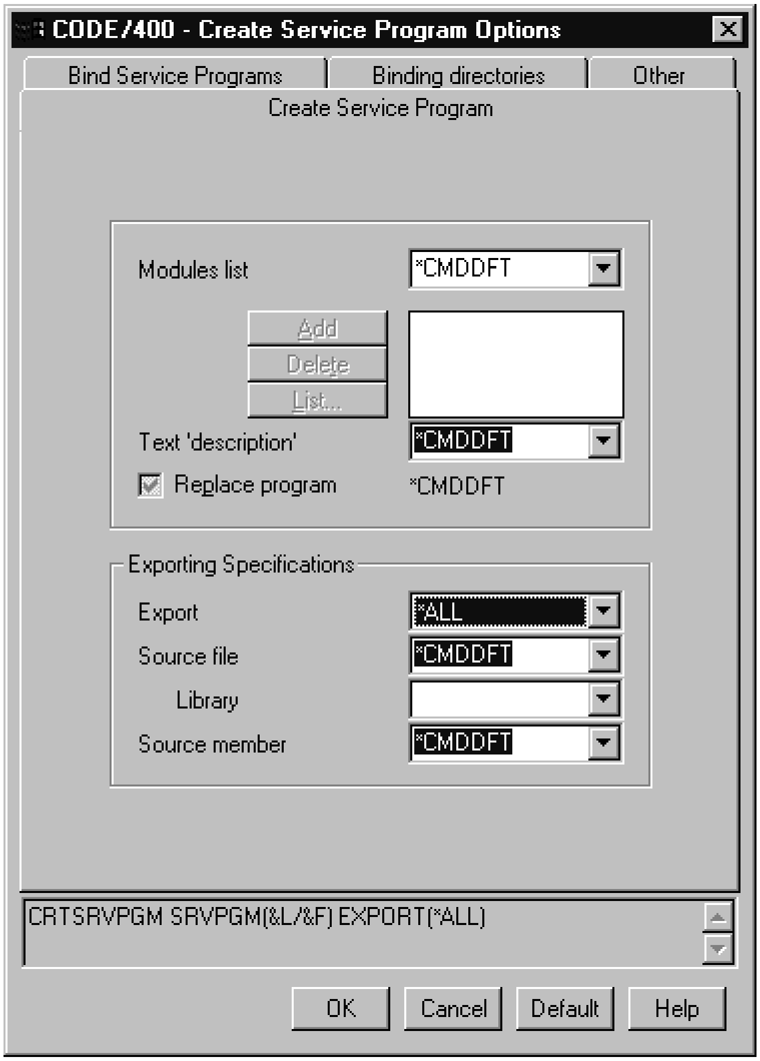
CRTSQLCBL
CRTSQLCBLI
CRTSQLCI
CRTSQLCPPI
CRTSQLRPG
CRTSQLRPGI
CRTSRVPGM
RUNJVA
UPDPGM
UPDSRVPGM

Command: CRTSRVPGM

Associated profile: ILE Service Program

- Select the **CRTSRVPGM** command from the command list

- Enter **wsslabxx/getdatas** in the Entry field for the program name on top of the dialog
- Press **the options** push button (**now it is the pushbutton**)



Modules list: *CMDDFT

Text 'description': *CMDDFT

☑ Replace program: *CMDDFT

Export: *ALL

Source file: *CMDDFT

Library:

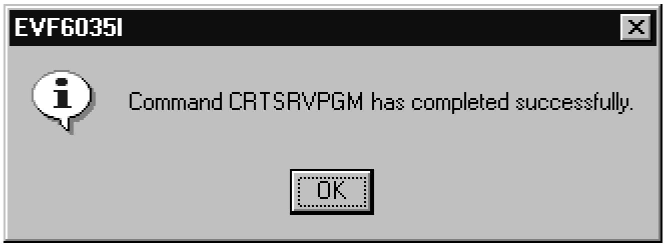
Source member: *CMDDFT

```
CRTSRVPGM SRVPGM(&L/&F) EXPORT(*ALL)
```

On the CODE Service Program Options dialog, on the **Create Service Program page**

- Click on the **little arrow** in the Export **combo box**
- Select ***ALL** from the drop down list
- Press the **OK** push button
- Press the **Submit** push button on the CODE Program Generator Dialog

Hopefully you get a message that your Service program has been generated



EVF6035I

Command CRTSRVPGM has completed successfully.

- Press the **OK** pushbutton in the confirmation message box

Finally you are ready to run your web application

By the way, you could have created the Service program outside of the CODE editor using the RSE iSeries command capabilities

Proceed to the section

Setting up the iSeries server information

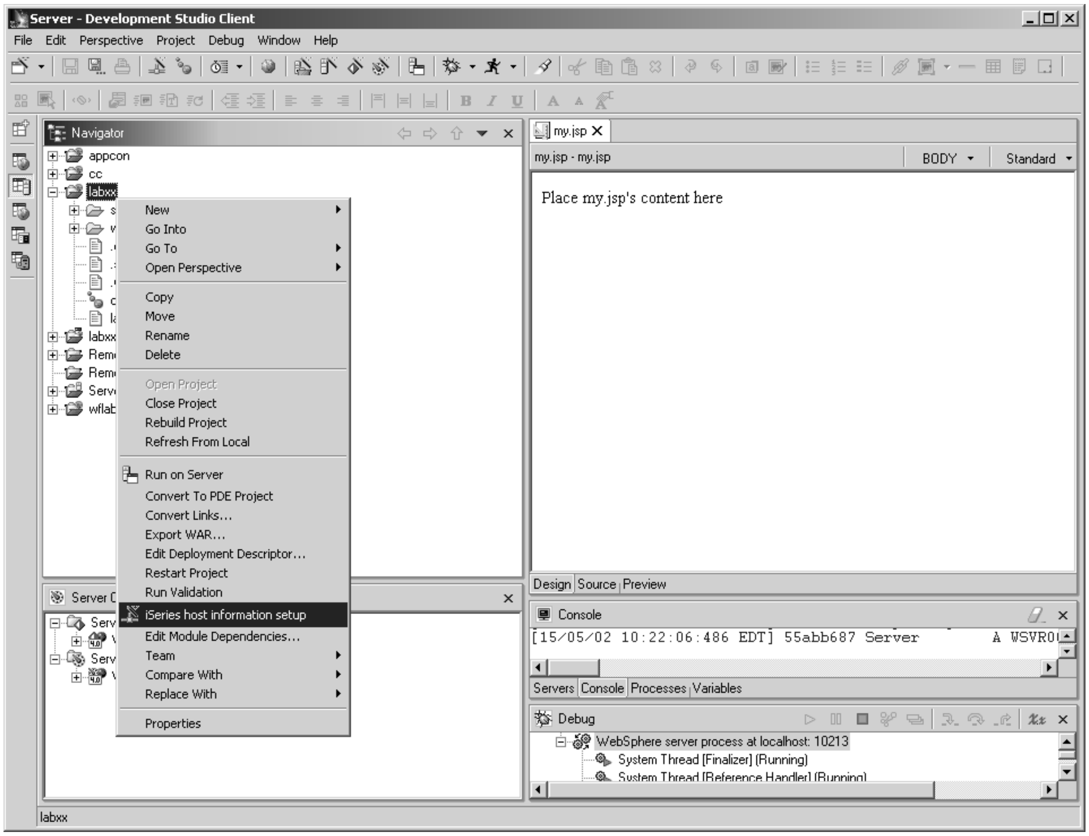
Setting up the iSeries server information

Before you can start testing your application, you will need to define where the program you just created, is located at. You need to specify the **iSeries server name** and the **userid/password** to be used when starting the job on the iSeries to run your program.
In the workbench:

- Go back to the web perspective,

  If there is **no** web perspective open,

  - Open a web perspective.



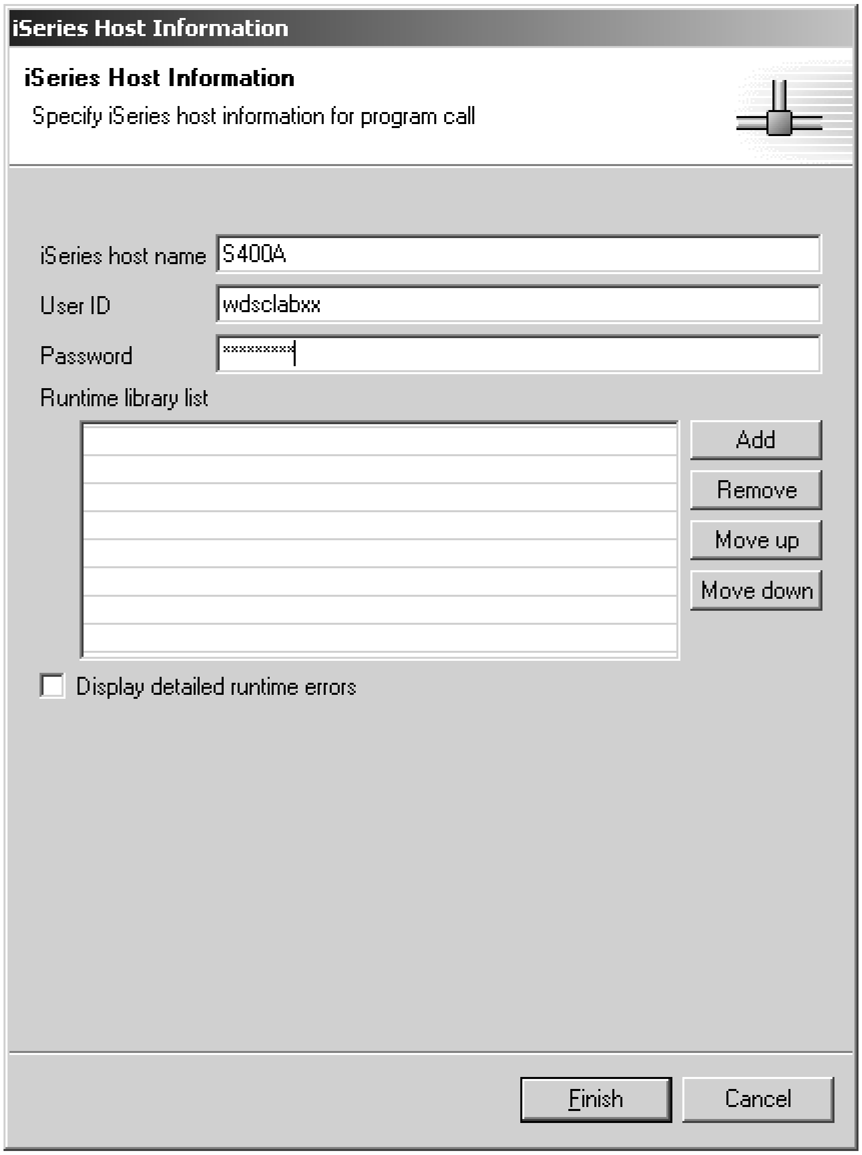
Place my.jsp's content here

```
[15/05/02 10:22:06:486 EDT] 55abb687 Server       A WSVR0
```

```
WebSphere server process at localhost: 10213
  System Thread [Finalizer] (Running)
  System Thread [Reference Handler] (Running)
```

- Select your project **labxx**
- Right mouse click on **labxx**
- Select **iSeries Host Information** from the pop up menu

The following dialog will show:



iSeries Host Information

iSeries Host Information

Specify iSeries host information for program call

iSeries host name: S400A

User ID: wdsclabxx

Password: *********

Runtime library list

□ Display detailed runtime errors

- Enter your **iSeries server name** into the **iSeries host name** entry field, this is the same server you used to get the source from.
  In real life you could specify a different server, but here in this Lab we use the same server for development and production
- Enter your userid in the **User ID** entry field, same as the one you used before
- Enter your password in the **Password** entry field, same as before

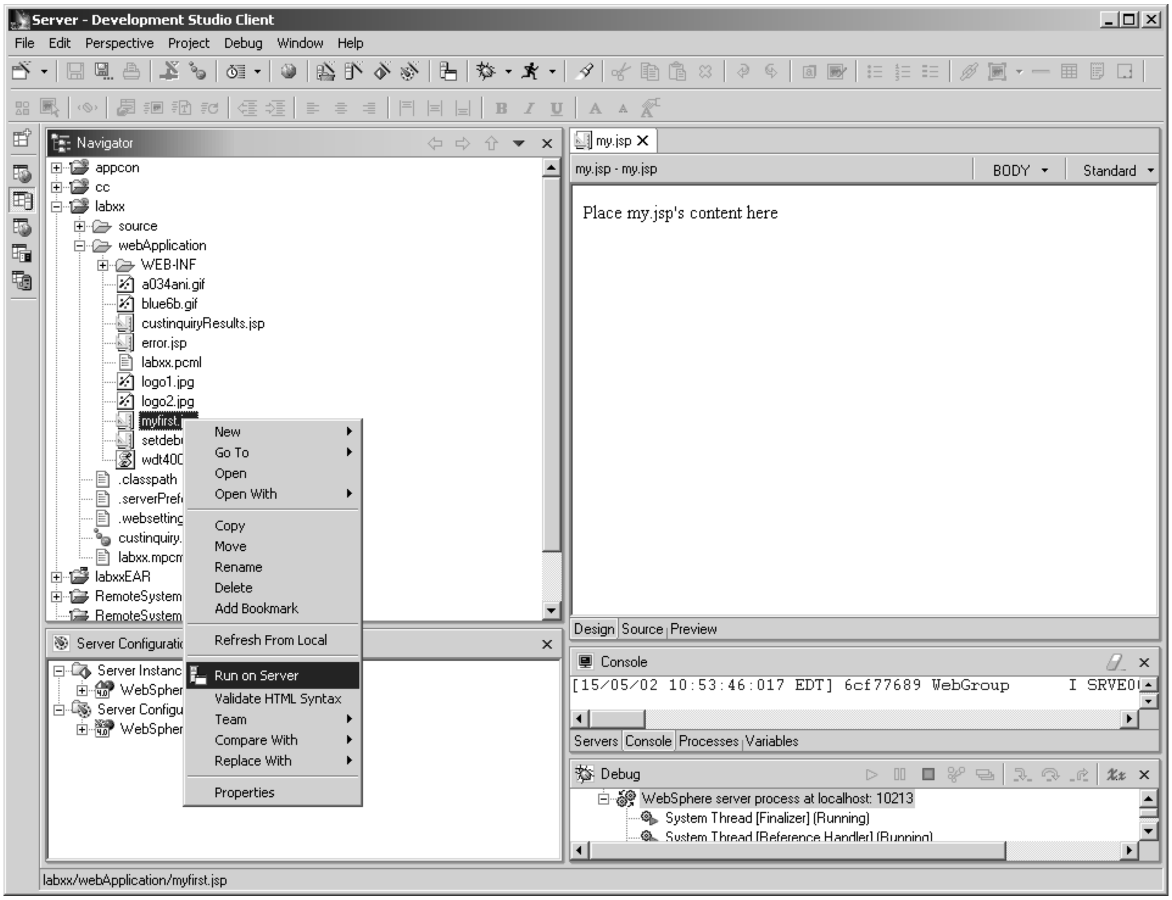
You don’t need to specify anything in the library list table, your userid has been set up in a way that it contains the correct library list. In your own environment you might have to use this table to add a library to the library list when the job gets started to run your program.

- Press the **Finish** push button on this dialog.

You are ready to test your application

Running the web application

Let's see if the application runs. In the workbench in the web perspective that you used in the step before to setup your host environment information.



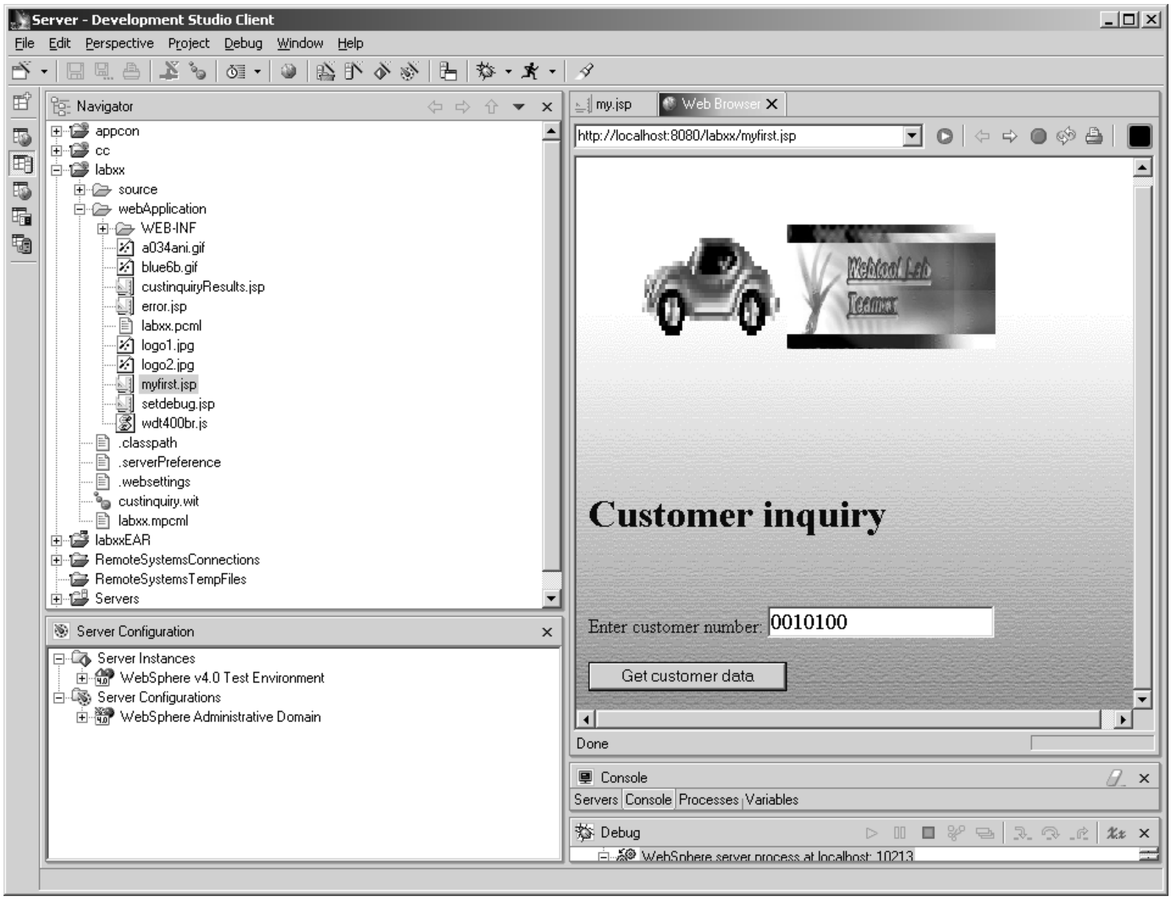
Place my.jsp's content here

```
[15/05/02 10:53:46:017 EDT] 6cf77689 WebGroup      I SRVE0(
```

```
WebSphere server process at localhost: 10213
    System Thread [Finalizer] (Running)
    System Thread [Reference Handler] (Running)
```

- Expand your project **labxx**
- Expand the **webApplication** folder
- Right mouse click on the **myfirst.jsp** file
- Select **Run on server** from the pop up menu

Your first page will be displayed in the workbench browser



.

- **Type**

0010100

as the **customer number**.

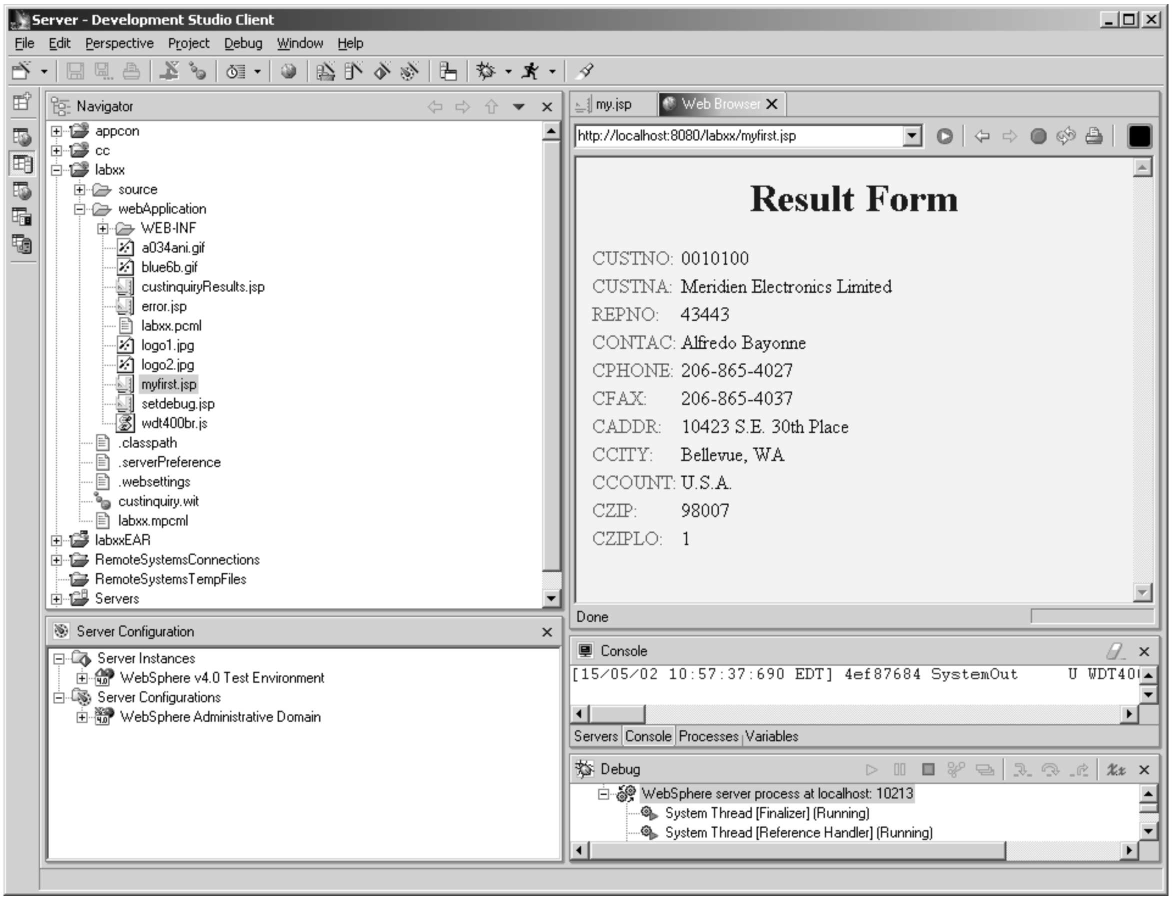
- Press the **Get customer data** push button.

This will bring up the generated **Result** page.

If you get a **Page not found error,** go to the Server perspective and **Restart** the Server instance.

Then go back and try the **Run on Server** again, the Server sometimes needs to be forced to pick up the changes.

You will see the **Result page** with customer data coming up in the workbench browser.



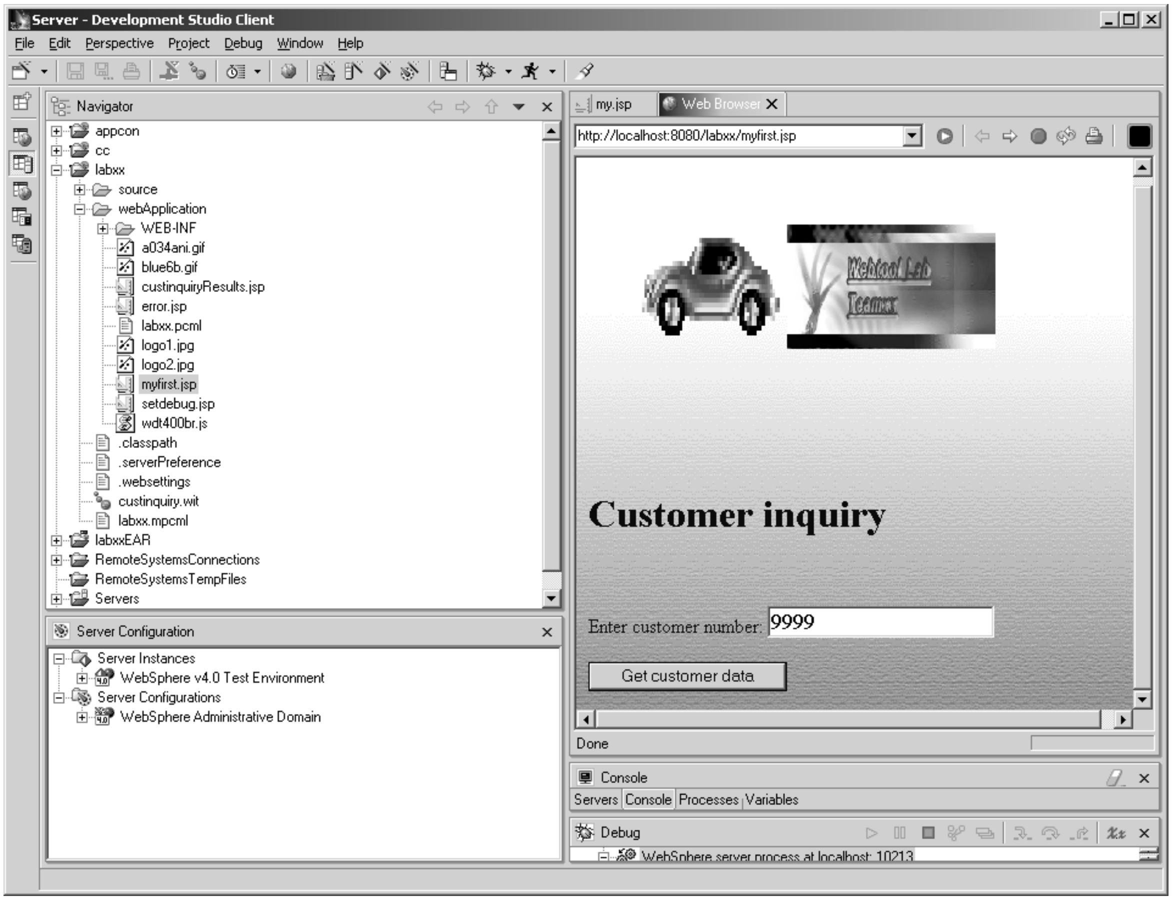
Result Form

CUSTNO:	0010100
CUSTNA:	Meridien Electronics Limited
REPNO:	43443
CONTAC:	Alfredo Bayonne
CPHONE:	206-865-4027
CFAX:	206-865-4037
CADDR:	10423 S.E. 30th Place
CCITY:	Bellevue, WA
CCOUNT:	U.S.A.
CZIP:	98007
CZIPLO:	1

```
[15/05/02 10:57:37:690 EDT] 4ef87684 SystemOut     U WDT40
```

We want you to test the message handling

- Press the **Back** button in the browser to get back to the first page



Customer inquiry

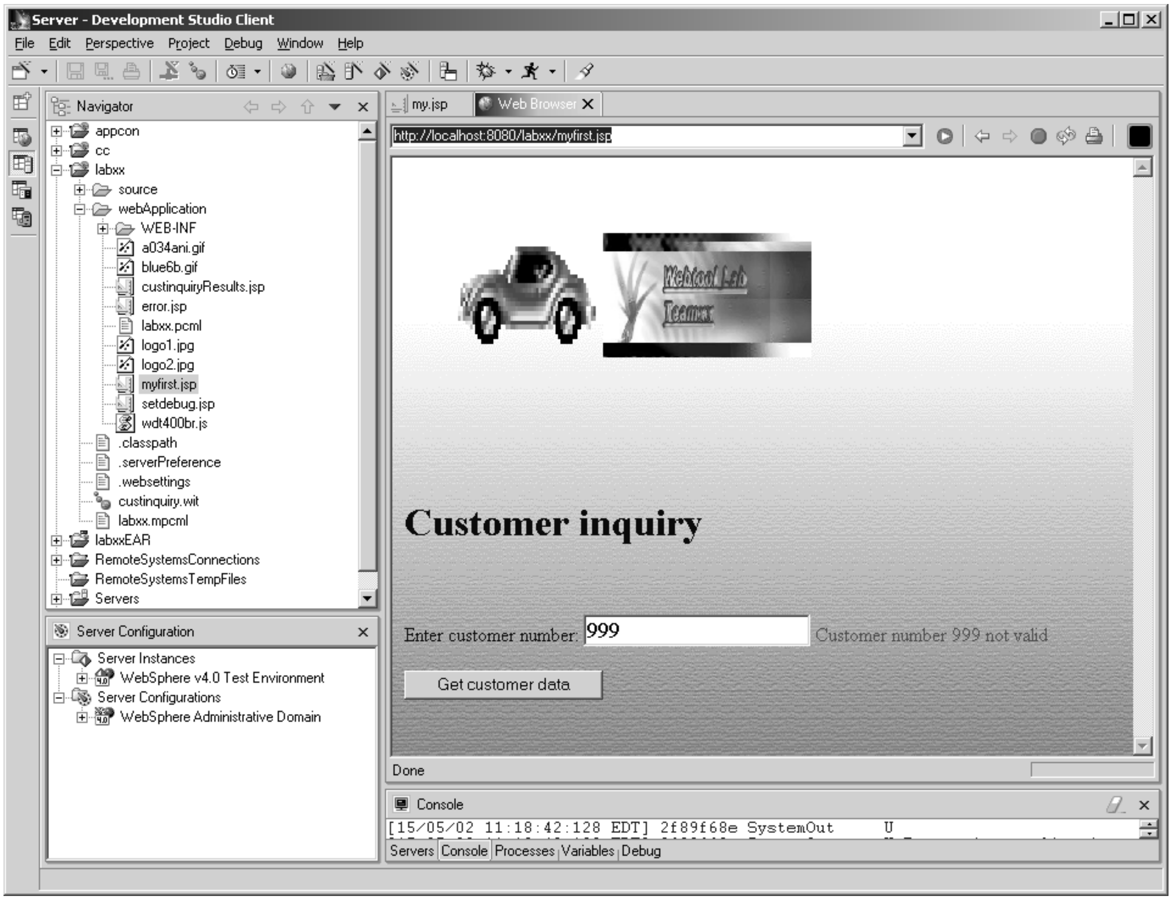
Enter customer number: 9999

Get customer data

- Key in a **wrong** customer number, like **9999**

- Press the **Get customer data** push button on this page

- An error message will appear



http://localhost:8080/labxx/myfirst.jsp

Customer inquiry

Enter customer number: 999 Customer number 999 not valid

```
[15/05/02 11:18:42:128 EDT] 2f89f68e SystemOut     U
```

This functionality is generated by the Interaction wizard. Remember that you specified something for the feedback field, you specified the message file to be accessed if the feedback field contains a **non 0** value, in this case the feedback parameter will contain the message ID to be used for the error message.

WDSc for iSeries Lab Summary

***See how easy it is to create a web application by using your RPG skills. If you need to produce browser based applications the WDSc for iSeries Wizards help you getting the job done*.**

This concludes the WDSc for Series hands on lab. We hope you **enjoyed** the experience. Using what you have learned from this lab you can build on this experience and create your own **RPG web** applications by exploring many of the other features in **WDSc for iSeries** that were not covered here.

Remember WDSc for iSeries includes these many tools to help you build your applications.

For more details, tips and techniques, and service and support information, be sure to visit the WDSc for iSeries web site at

www.ibm.com/software/ad/iSeries.

Have fun with WDSc for iSeries!