IBM WebSphere Development Studio Client for iSeries



Build a Web user interface for an RPG application using iSeries Web Tools

Version 5.1.2

IBM WebSphere Development Studio Client for iSeries



Build a Web user interface for an RPG application using iSeries Web Tools

Version 5.1.2

Contents

Chapter 1. Module 1. Introducing the Web customer inquiry application 1 Introducing iSeries Web Application Development 1 Introducing the Web customer inquiry application 2
Chapter 2. Module 2. Creating an RPG
program.7Exercise 2.1: Starting the product8
Exercise 2.2: Opening the Remote System Explorer
perspective
server
Exercise 2.4: Creating the IFS Directory for the Web
project
Exercise 2.6: Opening a source member for edit 15
Exercise 2.7: Reviewing Remote System LPEX Editor
features
Exercise 2.8: Editing an RPG source member and creating a *PGM object
creating a *PGM object
Chapter 3. Module 3. Creating the RPG
service program for your Web
application (optional)
Exercise 3.1: Starting the product
Exercise 3.2: Opening the Remote System Explorer
Exercise 3.2: Opening the Remote System Explorer perspective
Exercise 3.2: Opening the Remote System Explorer perspective27Exercise 3.3: Creating a connection to the iSeries
Exercise 3.2: Opening the Remote System Explorer perspective
Exercise 3.2: Opening the Remote System Explorer perspective
Exercise 3.2: Opening the Remote System Explorer perspective
Exercise 3.2: Opening the Remote System Explorer perspective27Exercise 3.3: Creating a connection to the iSeries server29Exercise 3.4: Creating the IFS Directory for the Web project30Exercise 3.5: Finding an iSeries source member32Exercise 3.6: Open a source member for edit33
Exercise 3.2: Opening the Remote System Explorer perspective27Exercise 3.3: Creating a connection to the iSeries server29Exercise 3.4: Creating the IFS Directory for the Web project30Exercise 3.5: Finding an iSeries source member32Exercise 3.6: Open a source member for edit33Exercise 3.7: Reviewing Remote System LPEX Editor
Exercise 3.2: Opening the Remote System Explorer perspective
Exercise 3.2: Opening the Remote System Explorer perspective
Exercise 3.2: Opening the Remote System Explorer perspective
Exercise 3.2: Opening the Remote System Explorer perspective 27 Exercise 3.3: Creating a connection to the iSeries server 29 Exercise 3.4: Creating the IFS Directory for the Web project 30 Exercise 3.5: Finding an iSeries source member 32 Exercise 3.6: Open a source member for edit 33 Exercise 3.7: Reviewing Remote System LPEX Editor features. 34 Exercise 3.8: Editing a member and creating a service program module 37 Chapter 4. Module 4. Creating a Web
Exercise 3.2: Opening the Remote System Explorer perspective 27 Exercise 3.3: Creating a connection to the iSeries server 29 Exercise 3.4: Creating the IFS Directory for the Web project 30 Exercise 3.5: Finding an iSeries source member 32 Exercise 3.6: Open a source member for edit 33 Exercise 3.7: Reviewing Remote System LPEX Editor features. 34 Exercise 3.8: Editing a member and creating a service program module 37 Chapter 4. Module 4. Creating a Web project. 45
Exercise 3.2: Opening the Remote System Explorer perspective 27 Exercise 3.3: Creating a connection to the iSeries server 29 Exercise 3.4: Creating the IFS Directory for the Web project 30 Exercise 3.5: Finding an iSeries source member 32 Exercise 3.6: Open a source member for edit 33 Exercise 3.7: Reviewing Remote System LPEX Editor features. 34 Exercise 3.8: Editing a member and creating a service program module 37 Chapter 4. Module 4. Creating a Web project. 45 Exercise 4.1: Opening the Web perspective 45
Exercise 3.2: Opening the Remote System Explorer perspective27Exercise 3.3: Creating a connection to the iSeries server29Exercise 3.4: Creating the IFS Directory for the Web project30Exercise 3.5: Finding an iSeries source member32Exercise 3.6: Open a source member for edit33Exercise 3.7: Reviewing Remote System LPEX Editor features.34Exercise 3.8: Editing a member and creating a service program module37Chapter 4. Module 4. Creating a Web project.37Exercise 4.1: Opening the Web perspective45Exercise 4.3: Setting up the iSeries server47
Exercise 3.2: Opening the Remote System Explorer perspective27Exercise 3.3: Creating a connection to the iSeries server29Exercise 3.4: Creating the IFS Directory for the Web project30Exercise 3.5: Finding an iSeries source member32Exercise 3.6: Open a source member for edit33Exercise 3.7: Reviewing Remote System LPEX Editor features.34Exercise 3.8: Editing a member and creating a service program module37Chapter 4. Module 4. Creating a Web project.37Exercise 4.1: Opening the Web perspective45Exercise 4.2: Creating a dynamic Web project47
Exercise 3.2: Opening the Remote System Explorer perspective
Exercise 3.2: Opening the Remote System Explorer perspective
Exercise 3.2: Opening the Remote System Explorer perspective

Exercise 5.4: Defining the iSeries GETDATAS service program invocation and parameters	
program invocation and parameters	. 72
	. 74
	. 76
	. 78
Exercise 5.9: Ensuring the customer number field	
0	. 82
Exercise 5.10: Visualizing the flow structure of your	
Web application	. 84
* *	
Chapter 6. Module 6. Running the Web	
application	89
Exercise 6.1: Opening the Web perspective	
Exercise 6.2: Finding the Web application and	. 09
	. 90
	. 90
Chapter 7. Module 7. Debugging a Web	
	OF
application	95
Exercise 7.1: Introducing the iSeries Integrated	
00	. 95
Exercise 7.2: Starting a Debug session using service	
	. 96
	. 97
	. 98
0	. 99
Exercise 7.6: Closing the debug session	. 99
Chanten O. Madula O. Adding on amor	
Chapter 8. Module 8. Adding an error	
page 1	01
page 1 Exercise 8.1: Creating the flow control page	1 01 101
page 1 Exercise 8.1: Creating the flow control page . Exercise 8.2: Modifying your Web interaction. .	101 101 103
page 1 Exercise 8.1: Creating the flow control page	101 101 103
page 1 Exercise 8.1: Creating the flow control page . Exercise 8.2: Modifying your Web interaction. . Exercise 8.3: Testing the new error page .	101 101 103
page 1 Exercise 8.1: Creating the flow control page . Exercise 8.2: Modifying your Web interaction. .	101 101 103
page	101 103 110
page	101 103 110
page 1 Exercise 8.1: Creating the flow control page . Exercise 8.2: Modifying your Web interaction. . Exercise 8.3: Testing the new error page . Chapter 9. Module 9. Enhancing the input page using Web tools . Exercise 9.1: Opening Page Designer. .	101 103 110 113 114
page	101 103 110 113 114
page 1 Exercise 8.1: Creating the flow control page . Exercise 8.2: Modifying your Web interaction. . Exercise 8.3: Testing the new error page . Chapter 9. Module 9. Enhancing the input page using Web tools . Exercise 9.1: Opening Page Designer. . Exercise 9.2: Working with page properties . Exercise 9.3: Linking a cascading style sheet to the	101 103 110 113 114 115
page 1 Exercise 8.1: Creating the flow control page . Exercise 8.2: Modifying your Web interaction. . Exercise 8.3: Testing the new error page . Chapter 9. Module 9. Enhancing the input page using Web tools . Exercise 9.1: Opening Page Designer. . Exercise 9.2: Working with page properties . Exercise 9.3: Linking a cascading style sheet to the	101 103 110 113 114 115
page	101 103 110 113 114 115
page 1 Exercise 8.1: Creating the flow control page 2 Exercise 8.2: Modifying your Web interaction. 2 Exercise 8.3: Testing the new error page 2 Chapter 9. Module 9. Enhancing the input page using Web tools 2 Exercise 9.1: Opening Page Designer. 2 Exercise 9.2: Working with page properties 3 Exercise 9.3: Linking a cascading style sheet to the Web page. 4 Exercise 9.4: Designing and adding a logo. 4 Exercise 9.5: Adding a heading 1 tag to the page 4	101 101 103 110 113 114 115 116 118 127
page	101 101 103 110 113 114 115 116 118 127 128
page	101 101 103 110 113 114 115 116 118 127 128 130
page	101 101 103 110 113 114 115 116 118 127 128 130
page	101 101 103 110 113 114 115 116 118 127 128 130 134
page	101 101 103 110 113 114 115 116 118 127 128 130 134
page	101 101 103 110 113 114 115 116 118 127 128 130 134 137
page 1 Exercise 8.1: Creating the flow control page 1 Exercise 8.2: Modifying your Web interaction. 1 Exercise 8.3: Testing the new error page 1 Chapter 9. Module 9. Enhancing the input page using Web tools 1 Exercise 9.1: Opening Page Designer. 1 Exercise 9.2: Working with page properties 1 Exercise 9.3: Linking a cascading style sheet to the Web page. 1 Exercise 9.4: Designing and adding a logo. 1 Exercise 9.5: Adding a heading 1 tag to the page 1 Exercise 9.7: Adding moving text to the page. 1 Exercise 9.8: Changing the text color. 1 Chapter 10. Summary. 1	101 101 103 110 113 114 115 116 118 127 128 130 134 137
page 1 Exercise 8.1: Creating the flow control page 1 Exercise 8.2: Modifying your Web interaction. 1 Exercise 8.3: Testing the new error page 1 Chapter 9. Module 9. Enhancing the input page using Web tools 1 Exercise 9.1: Opening Page Designer. 1 Exercise 9.2: Working with page properties 1 Exercise 9.3: Linking a cascading style sheet to the Web page. 1 Exercise 9.4: Designing and adding a logo. 1 Exercise 9.5: Adding a heading 1 tag to the page 1 Exercise 9.7: Adding moving text to the page. 1 Exercise 9.8: Changing the text color. 1 Chapter 10. Summary. 1 Appendix. Notices 1	101 101 103 110 113 114 115 116 118 127 128 130 134 137
page 1 Exercise 8.1: Creating the flow control page 1 Exercise 8.2: Modifying your Web interaction. 1 Exercise 8.3: Testing the new error page 1 Chapter 9. Module 9. Enhancing the input page using Web tools 1 Exercise 9.1: Opening Page Designer. 1 Exercise 9.2: Working with page properties 1 Exercise 9.3: Linking a cascading style sheet to the Web page. 1 Exercise 9.4: Designing and adding a logo. 1 Exercise 9.5: Adding a heading 1 tag to the page 1 Exercise 9.7: Adding moving text to the page. 1 Exercise 9.8: Changing the text color. 1 Chapter 10. Summary. 1	101 101 103 110 113 114 115 116 118 127 128 130 134 137

Exercise 5.3: Defining the iSeries GETDATA

Introduction

This tutorial teaches you how to create a simple e-business customer inquiry application that uses a Web-based front end to communicate with the business logic written in ILE RPG residing on an iSeries[™] server. While creating a browser user interface, you will learn how to use the iSeries Web Interaction wizard to generate input and output JSP files as well as Page Designer to enhance the input and output Web pages. You will also add iSeries Web components to your pages, for example, Web equivalents of iSeries command keys, input fields that accept only particular types of data, or output fields such as subfile names. You will then learn how to run the application in the WebSphere[®] Test Environment that is part of the product.

Prerequisites

In order to complete this tutorial end to end, you should already have working knowledge of the following:

- Basic Microsoft[®] Windows[®] operations such as working with the desktop and basic mouse operations such as opening folders and performing drag-and-drop operations.
- How Web applications work.
- How to use a browser to navigate the Internet.

It is also useful, but not necessary, for you to have basic knowledge of the following:

- DDS
- Servlets and Java[™] Server Pages (to understand the generated output of the Web Interaction wizard)

The file(s) required for this tutorial are available for download at http://ibm.com/software/awdtools/wdt400/library.

Time required

To complete the modules of this tutorial, you will need approximately **2 hours and 20 minutes**.

Learning objectives

The tutorial is broken into 8 modules, each with its own learning objectives. You can choose to complete one or all of the modules. Each module contains several exercises that must be completed in order for the tutorial to work properly.

Chapter 1, "Module 1. Introducing the Web customer inquiry application," on page 1 teaches you about the completed e-business customer inquiry application that you will build in this tutorial. In this module, you will:

- · View a concept diagram of the Web development tools process
- · Recognize the capabilities of the iSeries Web tools
- View the finished Web customer inquiry application

Chapter 2, "Module 2. Creating an RPG program," on page 7 teaches you how to use the Remote System Explorer perspective to edit and compile an iSeries program that will be invoked by the Web application to get data from the iSeries. In this module, you will:

- · Start the product
- Set the default workspace
- Check the Remote System Explorer perspective is open
- · Set up a new connection to an iSeries server
- Set up an IFS folder to contain the compiled iSeries source member
- Locate and RPG source file you want to edit
- · Edit an RPG source member with the Remote Systems LPEX Editor
- Highlight the tokens of your RPG source
- Show only comment lines in the RPG source
- Show all lines in the RPG source
- Display a format line
- Compile a program object
- Create a Program Call Markup Language (PCML) program interface description

Chapter 3, "Module 3. Creating the RPG service program for your Web application (optional)," on page 25 teaches you how to use the Remote System Explorer perspective to edit and compile an iSeries service program that will be invoked by the Web application to get data from the iSeries. In this module, you will:

- Start the product
- Set the default workspace
- Check the Remote System Explorer perspective is open
- · Set up a new connection to an iSeries server
- Set up an IFS folder to contain the compiled iSeries source member
- · Locate and RPG source file you want to edit
- · Edit an RPG source member with the Remote Systems LPEX Editor
- Highlight the tokens of your RPG source
- Show only comment lines in the RPG source
- Show all lines in the RPG source
- Display a format line
- Compile a service program object
- Create a Program Call Markup Language (PCML) program interface description

Chapter 4, "Module 4. Creating a Web project," on page 45 teaches you about the Web perspective and the tools in this perspective to help you create a Web project to contain your Web application files and to supply the information which the iSeries server will use for serving business information for this Web application. In this module, you will:

- · Access tools and views for iSeries Web application development
- Know the difference between a static Web project and a dynamic Web project
- Set up a dynamic Web project
- · Define where the RPG program or service program resides on an iSeries server

Chapter 5, "Module 5. Creating a Web application," on page 57 teaches you how to create a Web interaction to use an input and output page, and to create the servlet

to invoke an RPG program to get data from the iSeries database. You will also learn how to visualize the flow structure of your Web application using the Web Diagram editor. In this module, you will:

- Start the Web Interaction wizard
- Create an input page that invokes the interaction
- Create an output page that displays the results of the interaction
- Get the program interface definitions
- Specify the input and output parameters
- Specify the input parameters to show on the input page
- Specify what data to show on the output page
- · Specify what will happen when an incorrect customer number is entered
- Change the background color of the output page
- Make sure the customer number field is read only
- Know what Struts is all about
- View the flow structure of a Struts-based Web application

Chapter 6, "Module 6. Running the Web application," on page 89 teaches you how to run the Web application in the WebSphere Test Environment. You also learn about the WebSphere Test Environment. In this module, you will:

- · Check the Web perspective is open
- Locate input page of the Web application
- Invoke the Web application to run in the WebSphere Test Environment
- Test the Web application

Chapter 7, "Module 7. Debugging a Web application," on page 95 teaches you about the iSeries integrated debugger, helps you learn how to start a debug session using service entry breakpoints, and helps you learn how to add and delete breakpoints, display a variable, view the call stack, run, step and step into a program, set watch breakpoints and close the debug session. In this module, you will:

- · Recognize the features of the iSeries Integrated Debugger
- Set breakpoints
- Monitor an expression
- Expand a thread to show every program, module, procedure and method on the call stack at the current execution point
- Run the program to termination

Chapter 8, "Module 8. Adding an error page," on page 101 teaches you how to create an informative error page for customers when an incorrect customer number is entered.

- Create an error page using Page Designer
- Change the Web interaction to use the new error page
- View the result of adding flow control to the Web application

Chapter 9, "Module 9. Enhancing the input page using Web tools," on page 113 explains how to add color and some pictures to make the input page of the Web application more attractive. You will learn how to use the Page Designer tool and some other related Web tools. In this module, you will:

• Locate the Web application input page and start Page Designer

- View the title of the input page
- Add a style to the input page
- Preview the new style for the input page
- Start WebArt Designer
- Create a logo
- Resize the logo
- Save the object as a WebArt object
- Save the object for a Web page
- Place the object on the Design page
- Add a company name to the input page
- Add an image to the input page
- Add a marquee to the input page
- Apply color to certain areas of text

When you are ready, begin Chapter 1, "Module 1. Introducing the Web customer inquiry application," on page 1.

Chapter 1. Module 1. Introducing the Web customer inquiry application

This module teaches you about the Web customer inquiry application that you will build in this tutorial. You will also learn about the iSeries Web tools included with the product.

In this module, you will:

- · View a concept diagram of the Web development tools process
- Recognize the capabilities of the iSeries Web tools
- · View the finished Web customer inquiry application

Exercises

The exercises within this module must be completed in order:

- "Introducing iSeries Web Application Development"
- "Introducing the Web customer inquiry application" on page 2

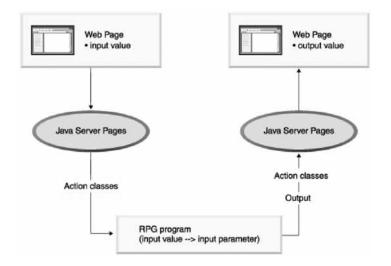
Time required

This module will take approximately 5 minutes to complete.

Introducing iSeries Web Application Development

With the product you can make your iSeries applications and data accessible beyond the green-screen interface. You can create a new Web interface that connects directly to your program's input and output parameters.

You can create interactive Web pages using Web development tools. You use the Web Interaction wizard to define how your pages interact with one or more ILE or non-ILE applications. This wizard generates Java Action classes and JSP files for use with data from HTML forms. When the end user enters data in a form, the input becomes data to your ILE and non-ILE programs, and the output from the programs is formatted for the Web. Your ILE and non-ILE logic can be separated into different programs for each stage of input and output (known as a Web interaction), or can be a single service program with entry points to handle each Web interaction. The following diagram illustrates the Web development tools process:



You can easily customize your HTML and JSP files using the editing tools in Web development tools. You can use these tools to create and update input forms, change the appearance or placement of blocks of text, and add backgrounds and images to your pages. The iSeries-specific Web components help you create Web versions of your input and output pages with the same kinds of input validation, output formatting, and subfile controls that native DDS screens provide. You do not need a detailed knowledge of HTML or Java to accomplish these tasks.

You have an easy test mechanism for your Web-enabled applications. You can easily run your program in the WebSphere test environment of the workbench, quickly make changes, and retest, rather than redeploy your application every time you want to verify functions. When you are finished, you can package and deploy your Web applications as J2EE-based Web archive (WAR) and Enterprise Archive (EAR) files, and then install them in a WebSphere Application Server.

You have looked at the concept diagram of the Web development tools process and can recognize the capabilities of the iSeries Web tools and now you are ready to begin "Introducing the Web customer inquiry application."

Introducing the Web customer inquiry application

Before you begin, you must complete "Introducing iSeries Web Application Development" on page 1.

By the end of this tutorial you will have created a simple e-business customer inquiry application that uses a Web-based front-end to communicate with RPG business logic residing on an iSeries server. You will first create the RPG program or service program for your Web application. You will then create a Web project and invoke the Web Interaction wizard to create the input and output pages and to create the servlet to invoke the RPG program or service program to get data from the iSeries database. Next, you will run the application in the local test environment on your workstation. You will also create an informative error page for when a customer enters an incorrect customer number.

Here is what the finished e-business customer inquiry application looks like.

After the WebSphere server has started, your Web application Customer Inquiry input page (inquiry.jsp) will display in the workbench browser:

Welcome Web Browser 🗙
http://localhost:9080/WSSLA8xx/inquiry.jsp 💽 🖸 😓 🔿 🚳 😂 📔
Customer Inquiry
Enter customer number: 0010100
Submit Reset
None

When you enter a customer number and click the **Submit** push button, the Customer Details output page (result.jsp) appears.

🕒 Welcome 🛛 🚳 Web Brows	er 🗙
http://localhost:9080/WSSLABxx/	custInquiry.do 🗾 🖸 🗇 🖘 🔘 🆃 📕
_	
Cus	tomer Details
Customer Number:	0010100
Customer Name:	Meridien Electronics Limited
Representative Numb	er: 43443
Contact:	Alfredo Bayonne
Phone Number:	206-865-4027
Fax Number:	206-865-4037
Address:	10423 S.E. 30th Place
City:	Bellevue, WA
Country:	U.S.A.
Zip:	98007
Submit Reset	
	×
Done	

Next, you click the 🔄 Back button in the browser to return to the Customer Inquiry input page. Here you enter an incorrect customer number and click **Submit**.

	💽 Web Browser 🗙		
http://localhos	t:9080/WSSLABxx/inquiry.jsp	• • • • • • •	
Enter cus	Customer	r Inquiry	×
Submit	Reset		Ŧ

An error message appears.

U Welcome	Web Browser 🗙	
http://localhost:908	0/WSSLABxx/custInquiry.d	• ▼ 0 ⇔ ⇒ ● ∲ ≙ ■
	Custome	er Inquiry
Enter custon	ner number:	
Submit Re	set	
Customer nu	mber 9999 not valio	t
-		<u>*</u>
Done		

You then add a new output error page to improve the message information so that if a wrong customer number is entered,



an error page opens instead of a message.



Module recap

You have completed Chapter 1, "Module 1. Introducing the Web customer inquiry application," on page 1. You have learned how to:

- · View a concept diagram of the Web development tools process
- Recognize the capabilities of the iSeries Web tools
- View the finished Web customer inquiry application

Now that you have this introductory knowledge of the product iSeries Web application development, you can continue with Chapter 2, "Module 2. Creating an RPG program," on page 7 if you want to work with an RPG program or Chapter 3, "Module 3. Creating the RPG service program for your Web application (optional)," on page 25 if you want to work with an RPG service program.

Chapter 2. Module 2. Creating an RPG program

This module teaches you how to create the iSeries program that you will be using for your Web application using the Remote System Explorer perspective. Remote System Explorer provides a PDM like environment but with a GUI interface, and full integration with all iSeries programmer tools in the workbench.

In this module, you will:

- · Start the product
- Set the default workspace
- Check the Remote System Explorer perspective is open
- Set up a new connection to an iSeries server
- Set up an IFS folder to contain the compiled iSeries source member
- · Locate an RPG source file you want to edit
- Open an RPG source member with the Remote Systems LPEX Editor
- · Highlight the tokens of your RPG source
- Show only comment lines in the RPG source
- Show all lines in the RPG source
- Display a format line
- Edit an RPG source member with the Remote Systems LPEX Editor
- · Compile a program object
- Create a Program Call Markup Language (PCML) program interface description

If you want to work with a procedure in a service program instead, go to Chapter 3, "Module 3. Creating the RPG service program for your Web application (optional)," on page 25.

If you haven't used the LPEX editor before, you may want to spend some time trying some LPEX editor features.

Exercises

The exercises in this module must be completed in order:

- "Exercise 2.1: Starting the product" on page 8
- "Exercise 2.2: Opening the Remote System Explorer perspective" on page 9
- "Exercise 2.3: Creating a connection to the iSeries server" on page 11
- "Exercise 2.4: Creating the IFS Directory for the Web project" on page 12
- "Exercise 2.5: Finding an iSeries source member" on page 14
- "Exercise 2.6: Opening a source member for edit" on page 15
- "Exercise 2.7: Reviewing Remote System LPEX Editor features" on page 16
- "Exercise 2.8: Editing an RPG source member and creating a *PGM object" on page 19

Time required

This module will take approximately **30 minutes** to complete.

Exercise 2.1: Starting the product

Now go ahead and start the product.

On your desktop taskbar:

	iBM WebSphere Homepage Builder V5.0	•
	💼 IBM WebSphere Studio	🕨 🛅 Development Studio Client Advanced Edition for iSeries 5.1.2 🌗
	🛅 ID Workbench	Development Studio Client Advanced Edition for iSeries 5.1.2
	🛅 Java 2 Runtime Environment	→
	🛅 Lavasoft Ad-aware 6	>
	🛅 Lotus Applications	>
	🛅 Lotus SmartSuite	>
	💼 Microsoft Office Tools	>
	🛅 Netscape Communicator	>
	💼 Norton AntiVirus Corporate Edition	>
All Programs	🛯 🛅 Object REXX for Windows	>
Air Frograms	🖬 🛅 Palm Desktop	→
	🛅 QuickTime	•
A	- 🛅 SoundMAX	•
🦺 Start	🛗 Startup	▶

1. Click Start > All Programs > IBM[®] WebSphere Studio > Development Studio Client Advanced Edition for iSeries 5.1.2

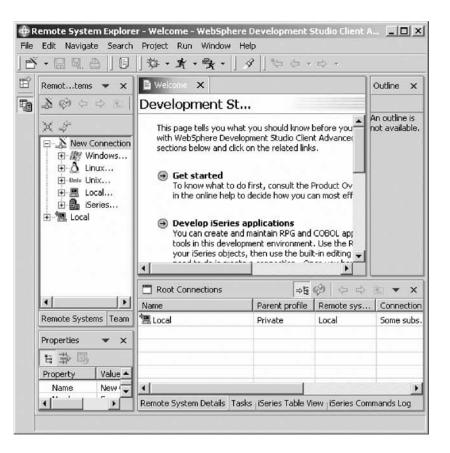
A dialog for workspace selection will appear asking you for the workspace location, unless you used the product before and selected not to show this dialog again.

WebSphere Studio		×
WebSphere Studio stores your work in a directory called a workspace. You can you start WebSphere Studio. Specify the directory to use for this session:	change the workspace	e each time
D:\Documents and Settings\yurich\My Documents\IBM\wdsc\WSSLABxx		Browse
Use this workspace as the default and do not show this dialog box again	OK	Cancel

The workspace contains all the information about your Development Studio projects. You can accept the default or store the work related to this tutorial, in a separate workspace.

- 2. To name the directory of the workspace as shown above, use the directory name WSSLABxx where XX is your unique team number.
- 3. Click OK.

After a few moments of loading, the workbench opens, and the initial window of the product appears on your desktop.



You have just started the product, set the default workspace and now you are ready to begin "Exercise 2.2: Opening the Remote System Explorer perspective."

Exercise 2.2: Opening the Remote System Explorer perspective

Before you begin, you must complete "Exercise 2.1: Starting the product" on page 8.

Check if the window title bar says: Remote System Explorer. If it does you are all set. Skip the next steps and go directly to "Exercise 2.3: Creating a connection to the iSeries server" on page 11.

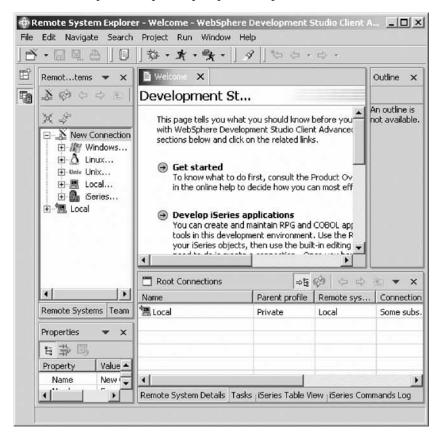
To open the Remote System Explorer perspective:

- 1. In the workbench, check whether you have the Remote System Explorer perspective already open. Look for the icon in the left taskbar of the workbench.
- 2. If you see it click the 🛐 RSE icon.

File				Search	1.			/indo
	• 6	1 및	. 🖱]0	175	· *	* *	* *
Ê	10 F	temot	e Sys	tems			•	×
	1	Ś	4	\$ E	Xd	÷.		
	E	Unix	Unix Loca iSerie	 I				
				las				

3. Otherwise, click **Window** > **Open Perspective** > **Remote System Explorer** on the workbench menu.

The Remote System Explorer perspective opens.



You have checked that the Remote System Explorer perspective is open and now you are ready to begin "Exercise 2.3: Creating a connection to the iSeries server" on page 11.

Exercise 2.3: Creating a connection to the iSeries server

Before you begin, you must complete "Exercise 2.2: Opening the Remote System Explorer perspective" on page 9.

- 1. Click the plus sign + beside **iSeries** under **New Connection** in the Remote Systems view.
- 2. Accept the default profile name.

			2
Name personal profile			
Uniquely name user profile			=
Welcome to Remote Systems, C you. Enter a profile name to unit decide for each new connection profile.	quely identify you fro	om your team mei	mbers. You will
Profile: //URICH			
Pronie: Irokich			

3. Click Next.

⊕ New				x
Remote iSeries Sy Define connection in		tion		
Parent profile:	YURICH			
	\$400a			ling
Host name: Description:	s400a			<u>•</u>
Verify host name				
	< Back	Next >	Finish	Cancel

In this exercise s400a is used as the name for the iSeries server. You will need to use the name or IP address of your particular iSeries server.

- 4. Type a host name, for example, s400a in the Host name field.
- 5. The Connection name field will be filled automatically with the same entry.
- 6. Click Finish.

Your new connection appears in the Remote Systems view and you are ready to work with objects on this iSeries system.

You have set up a new connection to an iSeries server and now you are ready to begin "Exercise 2.4: Creating the IFS Directory for the Web project."

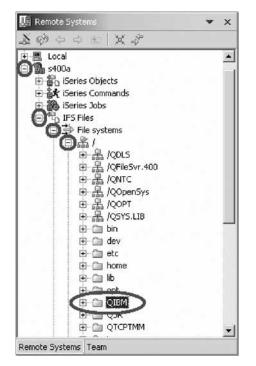
Exercise 2.4: Creating the IFS Directory for the Web project

Before you begin, you must complete "Exercise 2.3: Creating a connection to the iSeries server" on page 11.

IFS Files allow you to explore folders and files in the Integrated File System of the remote iSeries system. You will now create a directory in the IFS that will be used when you compile your iSeries source member.

Locate the newly created iSeries connection.

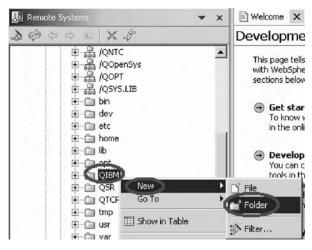
1. Expand your iSeries connections.



- 2. Expand the IFS Files by clicking the plus sign + beside it.
- 3. Expand the File systems.
- 4. If prompted, type your User ID and Password.

5ystem typ	e: iSeries
Host name:	
User ID:	WLABXX
Password:	serense serense
	☑ Save user ID
	Save password

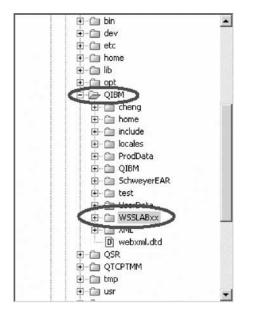
- 5. Select the Save user ID check box.
- 6. Select the Save password check box.
- 7. Expand the / Root node.
- 8. Select **QIBM**. Right-click **QIBM** and click **New** > **Folder** on the pop-up menu.



9. Enter WSSLABxx where XX is your unique team number in the **Folder name** field.

	×
2	7
	_
	2

10. Click Finish.



You should see the new WSSLABxx directory under QIBM.

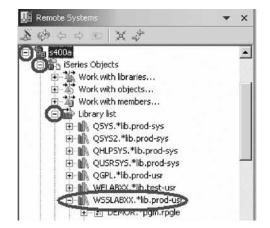
You have set up an IFS folder to contain the compiled iSeries source member and now you are ready to begin "Exercise 2.5: Finding an iSeries source member."

Exercise 2.5: Finding an iSeries source member

Before you begin, you must complete "Exercise 2.4: Creating the IFS Directory for the Web project" on page 12.

You will see a list of libraries under Library list in the Remote Systems view. Now you locate the source you want to edit.

To find an iSeries source member:

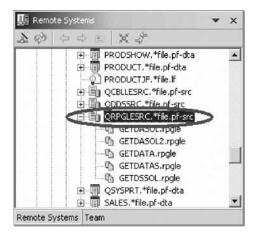


- **Note:** If you don't see WSSLABxx in your library list, then you need to add the library. You can right-click the Library list and click Add Library List Entry from the pop-up menu to add WSSLABxx to your library list.
- 1. Expand your iSeries connections.
- 2. Expand iSeries Objects by clicking the plus sign + beside it.

- 3. Expand Library List.
- 4. Expand the library WSSLABxx.

You will see all objects in this library.

5. Expand the source file QRPGLESRC.



6. Scroll down to the source members in the expanded list.

You have located an RPG source member you want to edit and now you are ready to begin "Exercise 2.6: Opening a source member for edit."

Exercise 2.6: Opening a source member for edit

Before you begin, you must complete "Exercise 2.5: Finding an iSeries source member" on page 14.

A shell source member is provided for you to use.

To edit a source member:

- 1. Right-click member GETDATA.
- 2. Click **Open with > Remote Systems LPEX Editor** on the pop-up menu.



The Editor window opens and you are ready to write your program.

Note: You can also double-click on a member to open the Remote Systems LPEX editor.

You have opened an RPG source member for edit and now you are ready to begin "Exercise 2.7: Reviewing Remote System LPEX Editor features."

Before you work with the RPG program source, if you have not worked with the Remote Systems LPEX editor, go to "Exercise 2.7: Reviewing Remote System LPEX Editor features."

If you already know the LPEX editor, then you are ready to write the RPG program; go to "Exercise 2.8: Editing an RPG source member and creating a *PGM object" on page 19.

Exercise 2.7: Reviewing Remote System LPEX Editor features

Before you begin, you must complete "Exercise 2.6: Opening a source member for edit" on page 15.

Before you start, here are some useful tips on using the LPEX editor. Let's look at some of the features of the LPEX editor, so you can later easily find your way around and use them:

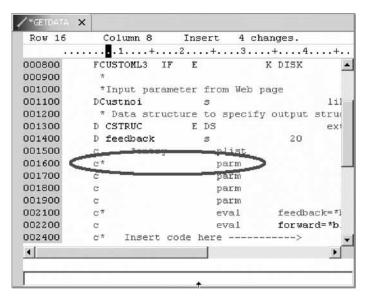
- ALT+L marks a line
- ALT+U unmarks selection
- ALT+D deletes a selected text
- ALT+C copies selected text
- ALT+M moves selected text
- Enter inserts a new line

Highlighting specification fields

The LPEX 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 get updated only after you move your cursor off the line.

To see how token highlighting works:

- 1. Move the cursor to a Calculation statement.
- 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 you typed the asterisk (*) becomes a comment line and its color changes accordingly.



- 5. Move the cursor back to column 7 and remove the asterisk (*).
- 6. Move the cursor off that line of code and the statement is tokenized. The token highlighting changes to reflect that this is a non-commented C specification.

Displaying types of lines

Using the LPEX editor, it is possible to have only particular types of source lines displayed at a time.

To display types of lines:

- 1. Right-click anywhere in the Editor window.
- 2. Click **Filter view** on the pop-up menu.

The menu items list the types of line selections that can be made.

3. Click Comments.

The Editor window now contains only those RPG statements that are comments.

*GETDATA Row 20	10000	Column 25 Insert 5 changes.
KOW 20		
	_	······································
000200		
000300		Sample RPGIV program to access file C
000400	*	Used in WDT hands on LAB
000500	*	Created by Claus Weiss
000600	*	
000700	*	F spec for file CUSTOML3 keyed by custom
000900	*	
001000	*	Input parameter from Web page
001200	*	Data structure to specify output structu
002100	c*	eval feedback=*bla
002400	c*	Insert code here>
003100	C*	EVAL FEEDBACK='CUS
003600	C.*	EVAL FEEDBACK='O'
		Done with inserting code>
004700		
•		
		1

- 4. Right-click anywhere in the Editor window.
- 5. Click Show all on the pop-up menu.

New	•
Prompt	
Syntax Check Line	
Select Format Line	
Cut	
Copy	
Paste	
Select line	
Select character	
Select rectangle	
Selected	
Deselect	
Filter view	•
Show all	
Save	

All statements types are displayed. In addition, the choices in the pop-up menu can be used to include only control specifications, user subroutines, and procedures and others. The Filter Selection option under the Selected option in the Edit menu allows the selection of only those lines containing a particular selected character string.

Checking the syntax of a file

Syntax checking is available for RPG code, and by default will be active. The syntax of 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.

Keeping track of columns in a specification line

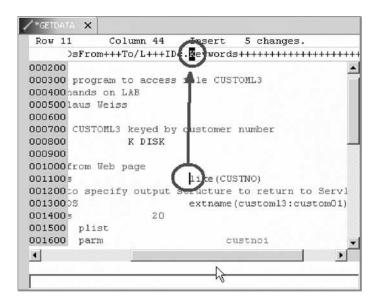
The format line is at the top of the Editor window, just above the first statement. A format line is used to help keep track of the columns in a particular specification line. The content of the format line can vary to reflect the particular type of specification being keyed such as F specs, C specs,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

To display a format line:

1. Click a line or use the arrow keys and click the left mouse button to make a line active.

The format line gets updated as a line gets focus.

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.



You can select a format line for any specification you want.

- 2. Click **Source** from the Editor menu.
- 3. Click Select Format Line on the pop-up menu.

The Format Line Selection window opens.

🕆 Format Line		2
Select format line: H: Control		E
	ОК	Cancel

Now you can start and create the RPG program.

4. Click OK.

You have highlighted tokens of your RPG source, shown only comment lines, shown all lines, displayed a format line and now you are ready to begin "Exercise 2.8: Editing an RPG source member and creating a *PGM object."

Exercise 2.8: Editing an RPG source member and creating a *PGM object

Before you begin, you must complete "Exercise 2.6: Opening a source member for edit" on page 15 and if you were unfamiliar with the editing features of the Remote Systems LPEX Editor then you must complete "Exercise 2.7: Reviewing Remote System LPEX Editor features" on page 16.

To make it easier for you, most of the RPG source is already prepared for you. You have already opened the correct source member GETDATA in the previous exercise using the Remote Systems view.

Read through the source. It contains an F spec to access file CUSTOML3, which contains the customer data keyed by customer number. The D specs define 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 Feedback variable is defined as a 20 length character field.

The Entry parameter list is defined as:

- CUSTNOI field (this is the input parameter).
- CSTRUC structure (this the data structure for the customer output data).
- Feedback 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.

To edit then create a *PGM object with RPG:

1. Add the code to get the customer record and check whether the chain was successful.

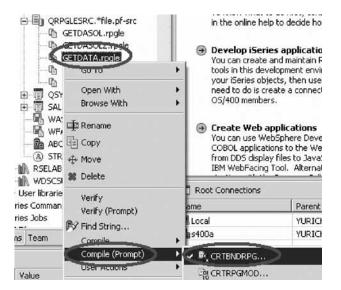
CLON01Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D+HiLoEq								
С	CUSTNOI	CHAIN(e)	CUSTOML3					
С		IF	not %found(CUSTOML3)					
С		EVAL	FEEDBACK='CUS0001 ' + CUSTNOI					
С		ELSE						
С		EVAL	FEEDBACK='0'					
С		ENDIF						

Your coding is complete, so let's create the program.

- 2. Click the Save \blacksquare icon on the workbench toolbar to save the member.
- 3. Click the **X** in the Editor window title bar to close the member.

L] Welcome

Be careful not to click the X on the workbench window title bar.



4. In the Remote Systems view, right-click the GETDATA member in source file QRPGLESRC and click **Compile (Prompt)** > **CRTBNDRPG** on the pop-up menu.

The Create Bound RPG Program (CRTBNDRPG) dialog opens.

Library:	WSSLABXX 💌	Name	1
Source file:	QRPGLESRC	Name, QRPGLESR	ic T
Library:	WSSLABXX 💌	Name	
Source member:	GETDATA	Name	
		<u>_</u>	
Source stream file:			
		<u>×</u>	
Seneration severity level:	10	0-20	
Text 'description':	*SRCMBRTXT	Character value	÷
Default activation group:	*YES 💌		1.
Activation group:	QILE	Name, QILE	
Binding directory:	*NONE	Name	
Library:	*LIBL 💌	Add Na	me
		Remove	
		Move up	
	J. J	Move down	
	Advanced Parameters		
Compiler options:		Add	
	*EVENTF	Remove(B)	
		Mave up	
		Move down	
Debugging views:	*ALL -		
•••		_	1
C Advanced (1)	All Parameters(2)	Keywords(3)	

5. In the **Debugging views** list, select *ALL

You need to specify two more parameters, these parameters are used to tell the compiler to generate the program interface description in a language called Program Call Markup Language (PCML). Later on you can use this description in the Web Interaction wizard.

- 6. Check the All Parameters check box at the bottom of the dialog.
- 7. Scroll down in the dialog until you see the Generate program interface field.
- 8. Select the ***PCML** value from the list.

TBNDRPG)		
		j.
	down(K)	
	(0)	r i
- Hone	Add(L)	
	Remove(Q)	
	Move up(S)	
	Move down(T)	1
*PEP 💌		-
*NOCOL		
	-	
	Character valu	e
-	<u> </u>	
*NONE	Add(X)	
	Remove(Y)	
	Move up(Z)	1
	Move down(0)	i I
*PCML		-
/gibm/wsslabxx/getdata.pcml		
	<u>×</u>	Ē
All Parameters(2)	Keywords(3)	
A) SRCFILE(WSSLABXX/ORPGLESRC) SRCME	R(GETDATA) OPTION(*EVENTE)	5
PGMINFO(*PCML) INFOSTMF('/qibm/wsslab>	x/getdata.pcml")	
		-
0	K Restore defaults C	ancel
	*CURRENT	Mere up(3) Move down(K) *NO *NO *NO *NO *NO *NO *NO *NO *NO *NONE Add(L) Move up(S) Move down(T) *NOCOL *NONE Add(X) PEP *NONE Add(X) Perove(Y) Move up(2) Move down(0) *PCML *PCML *PCML * All Parameters(2) Keywords(3) A) SRCFILE(WSSLABXX/QRPGLESRC) SRCMBR(GETDATA) OPTION(*EVENTF) PGMINFO(*PCML) INFOSTMF('/qbm/wsslabxx/getdata.pcm)

- 9. Enter /qibm/wsslabxx/getdata.pcml for the **Program interface stream file** field.
- 10. Click **OK** to start to compile.

After compile, the iSeries Error List view is shown, listing all compile errors. You may see only information and warning messages which means your compile was completed and the program object was created.

💐 iSeries Commands Log	<i>]</i> . ▼ ×
s400a	
CRTBNDRPG PGM(WSSLABXX/GETDATA) SRCFILE(WSSLABXX/QRP OPTION(*EVENTF) DBGVIEW(*ALL) REPLACE(*VES) Program GETDATA plastd intibrary WSSLABXX. OU highest severity Cause Program GETDATA was successfully created in libr message severity that resulted was 00. The program creation data 16:01:07.	ary WSSLABXX. The highest
Command Normal	Prompt Run
Remote System Details Tasks iSeries Table View iSeries Commands	s Log iSeries Error List

11. Click the **iSeries Commands Log** tab to check that the compile was successful. This log shows the Remote System Explorer job messages.

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

Fixing errors

In the error list view, check the errors:

1. Double-click the error.

This positions the cursor in the Editor window on the statement that is wrong.

- 2. Fix all errors.
- 3. Save the source member.
- 4. Go back to the Remote Systems view.
- 5. Right-click the member to run the CRTBNDRPG command.
- 6. Continue this cycle until you get a clean compile.

You have edited an RPG source member and created a program object.

Module recap

You have completed Chapter 2, "Module 2. Creating an RPG program," on page 7. You have learned how to:

- · Start the product
- Set the default workspace
- · Check the Remote System Explorer perspective is open
- · Set up a new connection to an iSeries server
- Set up an IFS folder to contain the compiled iSeries source member
- · Locate an RPG source file you want to edit
- · Open an RPG source member with the Remote Systems LPEX Editor
- Highlight the tokens of your RPG source
- Show only comment lines in the RPG source
- Show all lines in the RPG source
- Display a format line
- · Edit an RPG source member with the Remote Systems LPEX Editor
- · Compile a program object
- Create a Program Call Markup Language (PCML) program interface description

Now that you have created the RPG program, you can continue to Chapter 4, "Module 4. Creating a Web project," on page 45.

Skip Chapter 3, "Module 3. Creating the RPG service program for your Web application (optional)," on page 25 as you don't need to do this set of exercises.

Chapter 3. Module 3. Creating the RPG service program for your Web application (optional)

You do not need to do this module if you have done Chapter 2, "Module 2. Creating an RPG program," on page 7.

This module teaches you how to add the iSeries service program to your Web application using the Remote System Explorer perspective. Remote System Explorer provides a PDM like environment but with a GUI interface, and full integration with all iSeries programmer tools in the product workbench.

In this module, you will:

- · Start the product
- Set the default workspace
- Check the Remote System Explorer perspective is open
- · Set up a new connection to an iSeries server
- Set up an IFS folder to contain the compiled iSeries source member
- · Locate an RPG source file you want to edit
- · Open an RPG source member with the Remote Systems LPEX Editor
- Highlight the tokens of your RPG source
- Show only comment lines in the RPG source
- Show all lines in the RPG source
- Display a format line
- · Edit an RPG source member with the Remote Systems LPEX Editor
- · Compile a service program object
- Create a Program Call Markup Language (PCML) program interface description

If you haven't used the LPEX editor before, you may want to spend some time trying some LPEX editor features.

Exercises

The exercises in this module must be completed in order:

- "Exercise 3.1: Starting the product" on page 26
- "Exercise 3.2: Opening the Remote System Explorer perspective" on page 27
- "Exercise 3.3: Creating a connection to the iSeries server" on page 29
- "Exercise 3.4: Creating the IFS Directory for the Web project" on page 30
- "Exercise 3.5: Finding an iSeries source member" on page 32
- "Exercise 3.6: Open a source member for edit" on page 33
- "Exercise 3.8: Editing a member and creating a service program module" on page 37

Time required

This module will take approximately 30 minutes to complete.

Exercise 3.1: Starting the product

Now go ahead and start the product.

On your desktop taskbar:

	_					
	6	IBM WebSphere Homepage Builder V5.0	٠			
	m	IBM WebSphere Studio	÷	è	Development Studio Client Advanced Edition for iSeries 5.1.2	•
	è	ID Workbench	►	0	Development Studio Client Advanced Edition for iSeries 5.1.2	
		Java 2 Runtime Environment	×			_
		Lavasoft Ad-aware 6	⊁			
	6	Lotus Applications	►			
		Lotus SmartSuite	⊁			
	6	Microsoft Office Tools	►			
	6	Netscape Communicator	⊁			
	6	Norton AntiVirus Corporate Edition	►			
All Programs	i 🖻	Object REXX for Windows	⊁			
Air Frograms	6	Palm Desktop	⊁			
	6	QuickTime	₽			
(h)	6	SoundMAX	۲			
🛃 Start	6	Startup	₽			

1. Click Start > All Programs > IBM WebSphere Studio > Development Studio Client Advanced Edition for iSeries 5.1.2

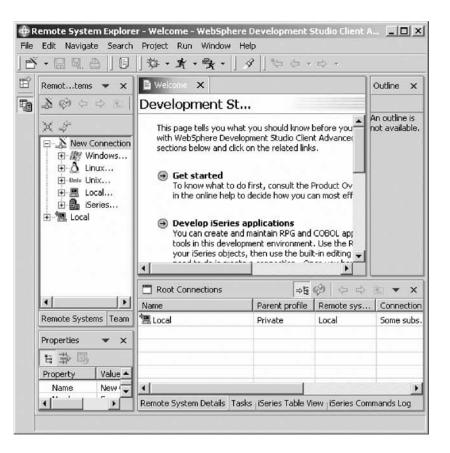
A dialog for workspace selection will appear asking you for the workspace location, unless you used the product before and selected not to show this dialog again.

WebSphere Studio		×
WebSphere Studio stores your work in a directory called a workspace. You can you start WebSphere Studio. Specify the directory to use for this session:	change the workspace	each time
D:\Documents and Settings\yurich\My Documents\IBM\wdsc\W/SSLABxx		Browse
\square Use this workspace as the default and do not show this dialog box again	ОК	Cancel

The workspace contains all the information about your Development Studio projects. You can accept the default or store the work related to this tutorial, in a separate workspace.

- 2. To name the directory of the workspace as shown above, use the directory name WSSLABxx.
- 3. Click OK.

After a few moments of loading, the workbench opens, and the initial window of the product appears on your desktop.



You have just started the product, set the default workspace and now you are ready to begin "Exercise 3.2: Opening the Remote System Explorer perspective."

Exercise 3.2: Opening the Remote System Explorer perspective

Before you begin, you must complete "Exercise 3.1: Starting the product" on page 26.

Check if the window title bar says: Remote Systems Explorer. If it does you are all set. Skip the next steps and go directly to "Exercise 3.3: Creating a connection to the iSeries server" on page 29.

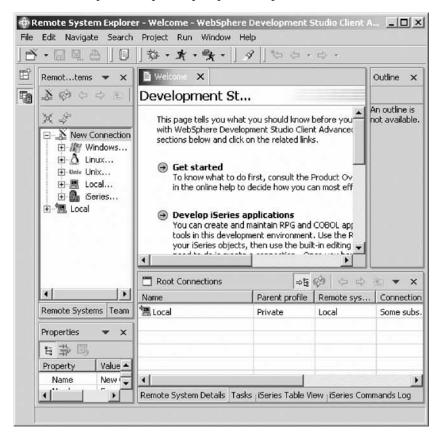
To open the Remote System Explorer perspective:

- 1. In the workbench, check whether you have the Remote System Explorer perspective already open. Look for the icon in the left taskbar of the workbench.
- 2. If you see it, click the 🜆 RSE icon.

File				Search	1.			/indo
	• 6	1 및	. 🖱]0	175	· *	* *	* *
Ê	10 F	temot	e Sys	tems			•	×
	1	Ś	4	\$ E	Xd	÷.		
	E	Unix	Unix Loca iSerie	 I				
				las				

3. Otherwise, click **Window** > **Open Perspective** > **Remote System Explorer** on the workbench menu.

The Remote System Explorer perspective opens.



You have checked that the Remote System Explorer perspective is open and now you are ready to begin "Exercise 3.3: Creating a connection to the iSeries server" on page 29.

Exercise 3.3: Creating a connection to the iSeries server

Before you begin, you must complete "Exercise 3.2: Opening the Remote System Explorer perspective" on page 27.

To create a connection:

- 1. Click the plus sign + beside **iSeries** under **New Connection** in the Remote Systems view.
- 2. Accept the default profile name.

₽ New			2
Name personal profile			
Uniquely name user profile			
Welcome to Remote Systems. Co you. Enter a profile name to unic decide for each new connection profile.	uely identify you	from your team i	members. You will
Profile: IrURICH			
		5	1
< Back	Next >	Finish	Cancel

3. Click Next.

🕀 New				×
Remote iSeries S Define connection i		ion		_#
Parent profile:	YURICH			•
Connection name	\$400a			
Host name:	\$400a			•
Description:				
✓ Verify host nam	e			
				Cancel
	< Back	Next >	Finish	

In this exercise s400a is used as the name for the iSeries server, you will need to use the name or IP address of your particular iSeries server.

4. Type a host name, for example, s400a in the **Host name** field.

- 5. The Connection name field will be filled automatically with the same entry.
- 6. Click **Finish**.

Your new connection appears in the Remote Systems view and you are ready to work with objects on this iSeries system.

You have set up a new connection to an iSeries server and now you are ready to begin "Exercise 3.4: Creating the IFS Directory for the Web project."

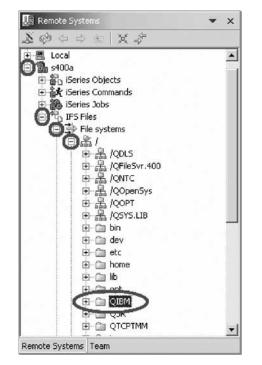
Exercise 3.4: Creating the IFS Directory for the Web project

Before you begin, you must complete "Exercise 3.3: Creating a connection to the iSeries server" on page 29.

IFS Files allow you to explore folders and files in the Integrated File System of the remote iSeries system. You will now create a directory in the IFS that will be used when you compile your iSeries source member.

Locate the newly created iSeries connection.

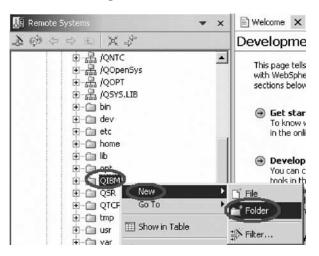
1. Expand your iSeries connections.



- 2. Expand the IFS Files by clicking the plus sign + beside it.
- 3. Expand the **File systems**.
- 4. If prompted, type your User ID and Password.

System type	: Iberies	
Host name:		
User ID:	WLABxx	
Password:	actolosolos	
	🔽 Save user ID	
	Save password	

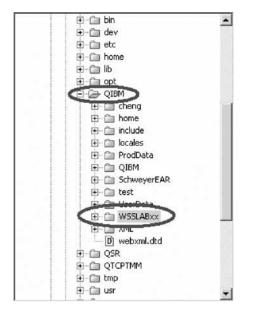
- 5. Select the Save user ID check box.
- 6. Select the Save password check box.
- 7. Expand the / Root node.
- 8. Select QIBM. Right-click QIBM and click New > Folder.



9. Enter WSSLABxx in the Folder name field.

💠 New Folder		×
Remote Folder		1.225
Create a New Folder		
Connection name: s400a]
New Folder in: /QIBM		
Folder name: WSSLABxx		
	Finish	Cancel

10. Click Finish.



You should see the new WSSLABxx directory under QIBM.

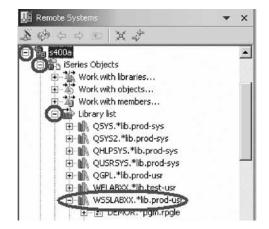
You have set up an IFS folder to contain the compiled iSeries source member and now you are ready to begin "Exercise 3.5: Finding an iSeries source member."

Exercise 3.5: Finding an iSeries source member

Before you begin, you must complete "Exercise 3.4: Creating the IFS Directory for the Web project" on page 30.

You will see a list of libraries under Library list in the Remote Systems view. Now you locate the source you want to edit.

To find an iSeries source member:



- **Note:** If you don't see WSSLABxx in your library list, then you need to add the library. You can right-click the Library list and click Add Library List Entry from the pop-up menu to add WSSLABxx to your library list.
- 1. Expand your iSeries server. Use the same server that you have used to specify the reference fields in the data structure.

- 2. Expand iSeries Objects by clicking the plus sign + beside it.
- 3. Expand Library List.
- 4. Expand the library WSSLABxx.
 - You will see all objects in this library.
- 5. Expand the source file QRPGLESRC.
- 6. Scroll down to the source members in the expanded list.

You have located an RPG source member you want to edit and now you are ready to begin "Exercise 3.6: Open a source member for edit."

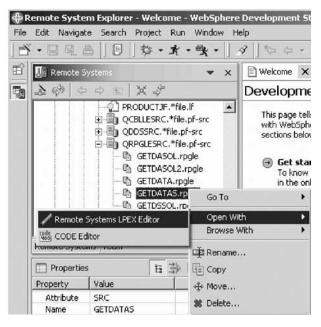
Exercise 3.6: Open a source member for edit

Before you begin, you must complete "Exercise 3.5: Finding an iSeries source member" on page 32.

One shell source member is provided for you to use.

To open a source member for edit:

- 1. Right-click member GETDATAS.
- 2. Click **Open with > Remote Systems LPEX Editor** on the pop-up menu.



The Editor window opens and you are ready to write your program.

Tip: You can also double-click on a member to open the Remote Systems LPEX editor.

You have opened an RPG source member for edit and now you are ready to begin "Exercise 3.7: Reviewing Remote System LPEX Editor features" on page 34.

Now before you edit the member, if you have not worked with the Remote Systems LPEX editor, go to "Exercise 3.7: Reviewing Remote System LPEX Editor features" on page 34.

If you already know the LPEX editor, you are ready to write your service program, go "Exercise 3.8: Editing a member and creating a service program module" on page 37.

Exercise 3.7: Reviewing Remote System LPEX Editor features

Before you begin, you must complete "Exercise 3.6: Open a source member for edit" on page 33.

Before you start, here are some useful tips on using the LPEX editor. Let's look at some of the features of the LPEX editor, so you can later easily find your way around and use them:

- ALT+L marks a line
- ALT+U unmarks selection
- ALT+D deletes a selected text
- ALT+C copies selected text
- ALT+M moves selected text
- Enter inserts a new line

Highlighting specification fields

The LPEX 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 get updated only after you move your cursor off the line.

To see how token highlighting works:

- 1. Move the cursor to a Calculation statement.
- 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 you typed the asterisk (*) becomes a comment line and its color changes accordingly.

Row 43	Co	olumn 5 Re	eplace 1 ch	ange.
		.1+2.	+3	+ 4 + 5 +
005034	De	ust1		likeds(cst)
005035	D f	eedback		20
005036	D *	end of proce	dure interfa	ace
005037	C*			
005038	С		eval	feedback=*blank
005041	С			
005042	C*	> add :	your code a:	fter this statement <
005043	C *			
005055	С	CUSTNOI	CHAIN	CUSTOML3
005056	С			
005057	С		If	not *IN50
005060	С		F17 b1.	cust1=cstruc
005061 🧲	C*		EVAL	feedback='0'
005062	С		LUNE	
005063	С		EVAL	feedback='CUS0001 '
005064	С		ENDIF	
•			1	

- 5. Move the cursor back to column 7 and remove the asterisk (*).
- 6. Move the cursor off that line of code and the statement is tokenized. The token highlighting changes to reflect that this is a non-commented C specification.

Displaying types of lines

Using the LPEX editor, it is possible to have only particular types of source lines displayed at a time.

To display types of lines:

- 1. Right-click anywhere in the Editor window.
- 2. Click **Filter view** on the pop-up menu.

The menu items list the types of line selections that can be made.

3. Click Comments.

The Editor window now contains only those RPG statements that are comments.

Row 26	Column 1 Replace	
	1+2+	.3+4+5
005011	*	
005012	*	
+ 005013 + 005016	****************	* * * * * * * * * * * * * * * * * * * *
005016	D* simple structure use	d for output parameters
005019	D *	
005020	D* Prototype for proced	ure getrecord
005021	D* if you don't specify	extproc with lowercase :
005022	D* GETRECORD in upperca	se in InterAction Wizard
005023	D*	
005025	D custnoi	7
005026	D myds	likeds (c
005027	D feedback	20
+ 005027 + 005029 + 005031	D* Start of procedure	
+ 005031	P* Procedure interface	
005033	D custnoi	7
005034	D cust1	likeds (c
•	1	

- 4. Right-click anywhere in the editor window.
- 5. Click **Show all** on the pop-up menu.

New	•
Prompt	
Syntax Check Line	
Select Format Line	
Gut	
Copy	
Paste	
Select line	
Select character	
Select rectangle	
Selected	•
Deselect	
Filter view	•
Show all	
Save	

All statements types are displayed. In addition, the choices in the pop-up menu can be used to include only control specifications, user subroutines, procedures and others. The Filter Selection option under the Selected option in the Edit menu allows the selection of only those lines containing a particular selected character string.

Checking the syntax of a file

Syntax checking is available for RPG code, and by default will be active. The syntax of 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.

Keeping track of columns in a specification line

The format line is at the top of the 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:

1. Click a line or use the arrow keys and click the left mouse button to make a line active.

The format line gets updated as a line gets focus.

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.

Row 45		Column 36	Replace		
	.CLO	N01Factor1++	+++++Opcode	(E)+Extended-factor2-	+++++++++
005038	С		eval	feedback=*blank	-
005041	С			T	
005042	C *	> add	l your code	after this statement	<
005043	C*				
005055	С	CUSTNOI	CHAIN	CUSTOML3	
005056	С				
005057	С		IÍ	not *IN50	
005060	С		EVAL	cust1=cstruc	
005061	С		EVAL	feedback='0'	
005062	С		ELSE	-	
005063	C		EVAL	feedback='CUS000:	1 '+ CU
005064	с		ENDIF	9	
005065	C*				
005066	C*	> VC	ur code end	ls here <	
005067	С		return		
005068	P		e		-
•					× Ē
	_				_

You can select a format line for any specification you want.

- 2. Click Source from the Editor menu.
- 3. Click **Select Format Line** on the pop-up menu. The Format Line Selection window opens.

💠 Format Line		×
Select format line: H: Control		Ľ
	ОК	Cancel
		9

Now you can start and create the service program.

4. Click OK.

You have highlighted tokens of your RPG source, shown only comment lines, shown all lines, displayed a format line and now you are ready to begin "Exercise 3.8: Editing a member and creating a service program module."

Exercise 3.8: Editing a member and creating a service program module

Before you begin, you must complete "Exercise 3.6: Open a source member for edit" on page 33 and if you were unfamiliar with the editing features of the Remote Systems LPEX Editor then you must complete "Exercise 3.7: Reviewing Remote System LPEX Editor features" on page 34.

To make it easier for you most of the RPG source is already prepared for you. You have already opened the correct source member GETDATAS in the previous exercise using the Remote Systems view. Read through the source. It contains an F spec to access file CUSTOML3 which contains the customer data keyed by customer number. The D specs 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 FEEDBACK is defined as 20 characters, to return message id and customer number in case the customer record was not found.

The prototype for procedure getrecord with the parameters is defined as the:

- CUSTNOI field (this is the input parameter)
- CSTRUC structure (this is the data structure for customer data)
- FEEDBACK field (returns error or success)
- **Note:** The external name is case sensitive. It has to match exactly what you specified in the Web Interaction wizard.

The procedure interface defines the parameters in the same way as the prototype. Inside the procedure after the procedure interface you need to add the RPG code.

Adding RPG code

To add the RPG code:

1.	CLONO	1Factor1+++++	+Opcode(E)	+Factor2++++++Result++++++Len++D+HiLoEq
	С	CUSTNOI	CHAIN(E)	CUSTOML3
	С		IF	%found(CUSTOML3)
	С		EVAL	FEEDBACK='0'
	С		EVAL	cust1=cstruc
	С		ELSE	
	С		EVAL	FEEDBACK='CUS0001 ' + CUSTNOI
	С		ENDIF	

2. You do a CHAIN to the database and move the database fields to the local procedure structure. If the database access fails, move the MSGID and Customer number to the FEEDBACK variable.

Your coding is complete, so let's create the service program module.

Creating the service program module

To create the service program module:

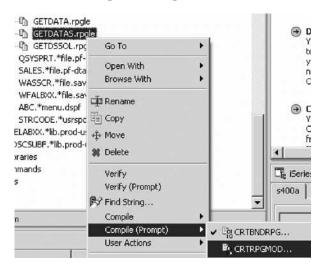
1. Click the Save 📃 icon on the workbench toolbar to save the member.

2. Click **X** in the Editor window title bar to close the member.



Be careful, don't click the X on the workbench window title bar.

3. In the Remote Systems view, right-click GETDATAS member in QRPGLESRC and select **Compile (Prompt)** > **CRTRPGMOD** on the pop-up menu.



The Create RPG Module (CRTRPGMOD) dialog opens.

GETDATAS	Name -
	Name -
	Name, QRPGLESRC
	Name
	Name
and the second se	<u></u>
	-
10	0-20
	 Character value
J SKCHOKTAT	Character value
Advanced Parameters	1
	d
*EVENTF Remo	oye
Move	2.00
1000	- dep
Move	down
(*ALL)	
*PRINT	
*NONE	
*NONE	Character value
*NONE Ad	id l
Remov	ve(B)
and the second	
1 Marca	i mili
Parameters(2)	Keywords(3)
	*EVENTF Rem Move Move Move Move *ALL *PRINT *NONE *NONE *NONE *NONE Ad

4. In the **Debugging views** list, select *ALL

You need to specify two more parameters, these parameters are used to tell the compiler to generate the program interface description in a language called Program Call Markup Language (PCML). Later on you can use this description in the Web Interaction wizard.

- 5. Select the All Parameters check box at the bottom of the dialog.
- 6. Scroll down in the dialog until you see the Generate program interface field.
- 7. Select the ***PCML** value from the list.

Create RPG Module (CRTRPGMO		Remove(I)
		Move up(J)
		Move down(K)
inable performance collection:	*PEP 💌	
Profiling data:	*NOCOL	
Sinding directory:	*NONE	Name
Library:	*LIBL 💌	Add(L) Nan
		Remove(Q)
		Move up(5)
		Move down(T)
icensed Internal Code options:		Character value
include directory:	*NONE	Add(x)
		Remove(V)
		Move up(Z)
Generate program interface:	PCML -	Move down(0)
Program interface stream file:	/gibm/wsslabxx/getdatas.pcml	
•		• •
Advanced(1)	All Parameters(2)	☐ Keywords(3)
CRTRPGMOD MODULE(WSSLABXX/GET DBGVIEW(*SOURCE) REPLACE(*YES) F	DATAS) SRCFILE(WSSLABXX/QRPGLESRC) SR 'SGMINFO(*PCML) INFOSTMF('/qibm/wsslabxx/	CMBR(GETDATAS) OPTION(*EVENTF)
	ОК	Restore defaults Cancel

- 8. Enter/qibm/wsslabxx/getdatas.pcml for the **Program interface stream file** field.
- 9. Click **OK** to submit the command.

After the compile the iSeries Error List view is shown, listing all compile errors. You may only see information and warning messages that means your compile was completed and the program module was created.

P. iSeries Commands Log 5400a	<i>∃</i> . ▼ ×
:400a	
CRTRPGMOD MODULE(WSSLABXX/GETDATAS) SRCFILE(WSS OPTION(*EVENTF) DBGVIEW(*ALL) REPLACE(*VES) Module GETDATAS placed in library wSSLABAA. To trigness se Cause	vency: Croated on 04/07/04 at 16:04:43. in library WSSLABXX, To highest
J Command Normal	Prompt
emote System Details Tasks iSeries Table View iSeries Comm	nands Log iSeries Error List

10. Click the **iSeries Commands Log** tab to check if the compile was successful. This log shows the Remote System Explorer job messages. If you get a message that there have been errors found in your program, go through the edit compile fix cycle.

Fixing errors

In the iSeries Error List view, check the errors:

1. Double-click on the error.

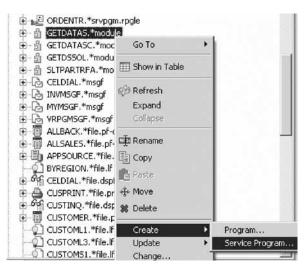
This positions the cursor in the Editor window on the statement that is wrong.

- 2. Fix all errors.
- 3. Save the source member.
- 4. Go back to the Remote Systems view.
- 5. Right-click the member to run the CRTRPGMOD command.
- 6. Continue this cycle until you get a clean compile.

Creating a service program

To create a service program:

1. In the Remote Systems view find the GETDATAS module in library WSSLABxx.



- **Note:** If you don't see the module, right-click library WSSLABxx and click **Refresh** on the pop-up menu.
- 2. Right-click GETDATAS and click **Create** > **Service Program** on the pop-up menu.

The Create Service Program (CRTSRVPGM) dialog opens.

Service program:	GETDATAS	Name
Library:	WSSLABXX 💌	Name
1odule:		Name, generic*
Library:		Add Name
	WSSLABXX/GETDATAS	Remove
		Move up
Export:		Move down
Export source file:	QSRVSRC V	Name, QSRVSRC
Library:	*LIBL V	Name
Export source member:	*SRVPGM	Name
ext 'description':	*BLANK	Character value
	All Parameters	Keywords

- 3. In the Export list, select *ALL.
- 4. Click OK.

Check the command log to see that the Service program was created.

5. Click the **iSeries Commands Log** tab at the bottom of the workbench. You should see a message that the Service program was created.

🔓 iSeries Commands Log	Э.	▼ ×
s400a		
CRTSRVPGM SRVPGM(WSSLABXX/GETDATAS) MODULE(WSSLABXX/GETDATAS) EXPO Service program GETDATAS construction by WSSLABXX. Cause		<u>ه</u> ۲.
		Ŧ
Command Normal	Prompt	Run
Remote System Details Tasks iSeries Table View iSeries Commands Log iSeries Error	List	

You have edited an RPG source member and created a service program object.

Module recap

You have completed Chapter 3, "Module 3. Creating the RPG service program for your Web application (optional)," on page 25. You have learned how to:

- Start the product
- Set the default workspace
- Check the Remote System Explorer perspective is open
- · Set up a new connection to an iSeries server
- · Set up an IFS folder to contain the compiled iSeries source member
- · Locate an RPG source file you want to edit
- Open an RPG source member with the Remote Systems LPEX Editor
- Highlight the tokens of your RPG source
- Show only comment lines in the RPG source
- Show all lines in the RPG source
- Display a format line
- Edit an RPG source member with the Remote Systems LPEX Editor
- Compile a service program object
- Create a Program Call Markup Language (PCML) program interface description

Now that you have created a RPG service program, you can continue to Chapter 4, "Module 4. Creating a Web project," on page 45.

Chapter 4. Module 4. Creating a Web project

This module teaches you how to create a Web project and then supply the information which the iSeries server will use for serving the business information for this Web application. You will also learn about perspectives, J2EE settings, Web projects, and the iSeries Web Tools Run-Time Configuration wizard.

In this module, you will:

- · Access tools and views for iSeries Web application development
- Know the difference between a static Web project and a dynamic Web project
- · Set up a dynamic Web project
- Define where the RPG program or service program resides on an iSeries server

Exercises

The exercises in this module must be completed in order.

- "Exercise 4.1: Opening the Web perspective"
- "Exercise 4.2: Creating a dynamic Web project" on page 47
- "Exercise 4.3: Setting up the iSeries server information" on page 51

Time required

This module will take approximately 15 minutes to complete.

Exercise 4.1: Opening the Web perspective

In this exercise you will use the Web perspective. This allows you to access tools and views specifically related to Web application development. However, you are still in the Remote System Explorer (RSE) perspective which gives you the tools to do your iSeries programming tasks as you have discovered in the previous module.

Perspectives

Before we go further, let's review perspectives. Perspectives are a collection of views and tools assembled to help different kinds of users to do their job when they use the workbench. These workbench users will have different jobs (roles), for example, one user needs to work with iSeries objects related to RPG/COBOL, another user is dealing with Java programs, and another user with Web page development. Each of these different user types will need different views of the files/objects they are working with 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.
- Remote System Explorer perspective for iSeries programmers.

You can also create your own perspective by using the **Save Perspective as** option under the **Window** menu in the workbench, after you have modified a perspective. Now that you know what a perspective is we can go ahead with our exercise. You are a Web developer in this tutorial 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. You create a Web project in the Web perspective.

To open the Web perspective:

🕀 Remote System Explorer - Welcom	e - WebSohere Development Stud	io Client Advanced Edition for iSeries
File Edit Navigate Search Project R	Window Help	
Š·BR&]0 ☆·J	- New Window	
🛱 🕼 Remote Systems	Open Perspective	ISeries Projects
5 0 0 0 0 X 0 1 X	Show view Hide Editors	Remote System Explorer
New Connection	Lock the Toolbars	Other

- 1. Click **Window** > **Open Perspective** on the workbench menu.
- 2. Then click **Other** on the pop-up menu.

The Select Perspective dialog opens.

🖏 Java	*
Java Browsing	
Java Type Hierarchy	
Plug-in Development	
(Profiling and Logging	
Remote System Explorer (defai	ult) i
Resource	
Leonver	
Web	
Line webFacing	
X XML	
IN OUR	
The VSL Debug	
扬 XSL Debug	1000
3 K XSL Debug	-
	ancel

- 3. Select **Web** from the list.
- 4. Click OK.

The workbench will now show the Web perspective with the Project Navigator view open on the left hand side.

š • 🛛 🔍 🖨	· # [@ @ [] · \$] @ [# &]	0 4
5 4 * 4 *		
Projegator 💌 🗙	🗎 Welcome 🗙	Palet ×
合业团发给	Development Stud	
 Project Struts E Project Struts E Page Data × No data components found. Page D Client I () 		A palette is not available for this editor
Attributes × No attributes are availa	your iSeries objects, then use the built-in editing and need to do is create a connection. Once you have a OS/400 members. Create Web applications You can use WebSphere Development Studio Client f	
	🗟 Quick Edit 🗸 🗸 🗸	
	No script to edit	
Attributes Styles	Quick Edit Tasks Colors Servers	Pale 1

You have accessed tools and views for iSeries Web application development and now you are ready to begin "Exercise 4.2: Creating a dynamic Web project."

Exercise 4.2: Creating a dynamic Web project

Before you begin, you must complete "Exercise 4.1: Opening the Web perspective" on page 45.

Web projects hold all of the Web resources that are created and used when developing your Web application. The first step to creating or importing a Web application is to create either a static or a dynamic Web project. Static Web projects are meant to contain only simple Web site resources, such as HTML files. Dynamic Web projects are used to structure Web applications that will use more complicated, dynamic Web technologies, such as JavaServer Page files, and possibly data access resources.

To create a dynamic Web project:

- 1. Click **File** > **New** on the workbench menu.
- 2. Then click Dynamic Web Project on the pop-up menu.

Edit Navi New		oject Run Window Help
Close Close All	Ctrl+F4 Ctrl+Shift+F4	Dynamic Web Project
Save	Ctrl+5	 Static Web Project Folder
Save As,		TTML/XHTML File
Save All Revert	Ctrl+Shift+S	3SP File
Move		- Faces JSP File

This starts the Dynamic Web Project wizard. The initial page of the wizard opens.

🕀 New Dynamic Web Project		×
Dynamic Web Project Create a Web project for dynamic content (HTML files, JSPs, and Servlets)).	
Project name: WSSLABxx		
Project location: D:\Documents and Settings\yurich\My Documents\IBM\v	wdsc\WSS	Browse
Configure advanced options		
<back next=""> Fin</back>	ish	Cancel

3. Accept the default value in the **Project location** field.

This is where your project is stored in your file system.

- 4. In the **Project name** field, type WSSLABxx.
- 5. Select the **Configure advanced options** check box to specify or customize the project options, such as adding Struts support to your Web project.
- 6. Click Next.

The J2EE Settings page opens.

This page allows you to specify the J2EE level for the Web project. If you plan to use the WebSphere Application Server Version 5 only, then you 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 use J2EE level 1.2 since these earlier versions of WebSphere Application Server don't support the J2EE level 1.3. In this exercise you use the Web application in the WebSphere Test Environment, which is Version 5, so you use J2EE level 1.3 which is the default.

🕀 New Dyna	mic Web Project	ų.
	s Page orise Application project settings, context root, J2EE level and target ble only if enabled in the J2EE preference page).	
EAR project	WSSLABXXEAR	New
Context root:	WSSLABxx	
J2EE level:	1.3	
	32EE Level 1.3 includes a Servlet Specification level of 2.3 and a 35P Specification level of 1.2. Applications developed for this 32EE level typically target a WAS version 5.0 server.]
	<u>×</u>	J
	< Back Next > Finish	Cancel
		Cancer

7. In the EAR project field, type WSSLABXXEAR.

A new or existing Enterprise Application project must be associated with your new Web project for purposes of deployment.

The Enterprise Application project is new, so you typed the name of the new project. When your Web project is created at the end of the wizard, the new Enterprise Application project (EAR file) is also created. The default is an Enterprise Application project named DefaultEAR located in the same directory as your new Web project.

8. Leave the default value for the Context root field.

The default value for the context root is the name you provided for your Web project. The context root is the Web application root, the top-level directory of your application when it is deployed to a Web server. The context root can also be used by the links builder to ensure that your links remain ready to publish as you move and rename files inside your project.

9. Leave the default value in the J2EE level list.

By default, the Web project's J2EE level is set to the Workbench's J2EE level.

Any new servlets and JSP files that you create should adhere to the latest specification level available; previous specification levels are offered to accommodate any legacy dynamic elements that you expect to import into the project.

10. Click Next.

The Features Page opens.

New Dynamic Web Project Features Page Select a Web Project feature. A feature can prov Web Project.	vide additional functionality for the
Web Project features:	Description: Select this feature to have support for Struts added to your project
< Back	Next > Finish Cancel

- 11. Leave the **Default style sheet (CSS)** check box selected to create a default CSS file (called Master.css) for any HTML and JSP files included in the project.
- 12. Select the **Add Struts support** check box as you want to create a project that uses Struts technology.
- 13. Select the **iSeries Web Components Tag Library** check box as you want to add iSeries fields and controls to the Web pages of your Web application.
- 14. Click Next to go to the Struts Settings page and ensure that Create a Resource Bundle for the Struts Project is selected. If not, select the Override default settings check box and then select Create a Resource Bundle for the Struts Project.

💠 New Dynamic Web	Project X
Struts Settings	
Select the initial settings	tor Struts
🔽 Cr <u>eate a Web diag</u> ra	am for the project
Override default set	
- Settings	-
Struts version: 1.1	
Default Java package	prefix; com.ibm.wsslabxx
Create a Resource	Bundle
Resource bundle	
Java package:	com.ibm.wsslabxx.resources Browse
Resource bundle nar	ne; ApplicationResources
	< Back Next > Finish Cancel

15. Click Finish to create the Web project.

Now you are back in the workbench and your new project and project files appear in the Project Navigator.



You know the difference between a static Web project and a dynamic Web project and have set up a dynamic Web project and now you are ready to begin "Exercise 4.3: Setting up the iSeries server information."

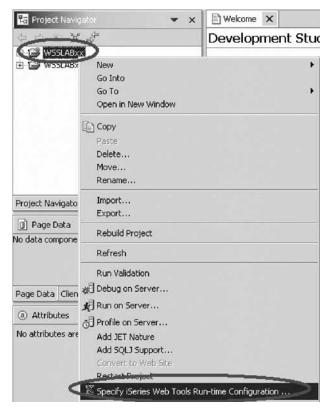
Exercise 4.3: Setting up the iSeries server information

Before you begin, you must complete "Exercise 4.2: Creating a dynamic Web project" on page 47.

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

To set up the iSeries server:

1. In the Web perspective, right-click the WSSLABxx project in the Project Navigator.



2. Click **Specify iSeries Web Tools Run-Time Configuration** on the pop-up menu.

Tip: You could also use the button from the workbench toolbar to open the Run-Time Configuration wizard.

The iSeries Web Tools Run-Time Configuration wizard opens.

💠 iSeries Web Tools Run-time Configuration	×
Configure Authentication	
Specify how program call Web interactions should connect and authenticate	
C Prompt for user ID and password	
Host name: \$400a	
User ID: WLABxx	
Password: ******	
Enable password encoding	
C Use JCA connector and JAAS credentials	
Specify the JNDI name:	
C Single signon	
Host name:	
JNDI resource reference: eis/IdentityToken_MyApplication_Reference	
EIM application identifier: MyApplication_EID	
< Back Next > Finis	h Cancel
< Back Next > Finis	

You use this wizard to specify how Web interactions that perform program calls to an iSeries host should be connected and authenticated.

- 3. In the **Host name** field, type the name of the iSeries host where your program is located, for example s400a.
- 4. Enter your user ID and password for the iSeries host in the appropriate fields. You need to ensure that your User ID has been set up in a way that it contains the correct library list. If not, you will have to use the table on the next page to add a library to the library list when the job gets started to run your program.
- 5. **Enable password-encoding** is selected by default. This option ensures that the password appears in encoded format in the web.xml file. If you clear the check box for this option, the password appears as plain text in the web.xml file.
- 6. Click Next.
- 7. (Optional) If you need to add WSSLABxx to your library list, then type WSSLABxx in the **Library** field and click **Add**.
- 8. Select the **Display detailed run-time errors** check box to see the details for any errors that occur during the runtime of your Web application. This is useful for tracing and debugging development errors that may occur.

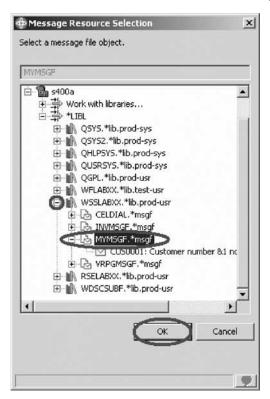
	ols Run-time Configurat	ion		2
nfigure Run Tim pecify the run-time	ne e configuration for program	call of i5eries Web deve	lopment tools	_
Run-time library lis	st			
Library:			_	Add
Library	Library Position			Remove
				Move Up
				Move Down
Session timeout pr Library:	ocedure	Program object:	ſ	
			-	
Entry point:		Program type:	*SRVPGM	-
Entry point:		Program type:	*SRVPGM	•
		Program type:	I. 75.00.77.0	erties Browse
Message handling	; com\ibm\wsslab		I. 75.00.77.0	erties Browse
Message handling Message resource	; com\ibm\wsslab		I. 75.00.77.0	erties Browse
Message handling Message resource	; com\ibm\wsslab		I. 75.00.77.0	erties Browse

- 9. Select the **Use iSeries message file** check box to retrieve error messages from the iSeries host.
- 10. Click **Browse** to specify the iSeries message file.
- 11. Click the plug sign + beside s400a.

12. Click the plus sign + beside *LIBL.

Message Resou	ce Selec	tion		>
elect a message file	object.			
🗆 💁 New Connec	tion			
COLUMN AND ADDRESS OF	on create	d successf	ully, Exp	and this aga
	h libraries			
Ū				
•				Þ
		OK	1	Cancel
		1999		

13. Drill down to WSSLABXX > MYMSGF.*msgf.



- 14. Click OK.
- 15. Click **Finish** to define authentication and run-time values for your host program or procedure call to be made by your Web interaction in the Web project.

You have defined where the RPG program or service program resides on the iSeries server.

Module recap

You have completed Chapter 4, "Module 4. Creating a Web project," on page 45. You have learned how to:

- · Access tools and views for iSeries Web application development
- Know the difference between a static Web project and a dynamic Web project
- Set up a dynamic Web project
- Define where the RPG program or service program resides on an iSeries server

Now that you have created a Web project, you can continue to Chapter 5, "Module 5. Creating a Web application," on page 57.

Chapter 5. Module 5. Creating a Web application

This module teaches you how to create the interaction to use an input and output page, and to create the servlet to invoke an RPG program to get data from the iSeries database. You will also learn about interactions, Web input and output pages and Program Call Markup Language (PCML). Finally you will learn how to see the flow structure of your Web application.

In this module, you will:

- Start the Web Interaction wizard
- Create an input page that invokes the interaction
- Create an output page that displays the results of the interaction
- Get the program interface definitions
- Specify the input and output parameters
- Specify the input parameters to show on the input page
- Specify what data to show on the output page
- · Specify what will happen when an incorrect customer number is entered
- Change the background color of the output page
- Make sure the customer number field is read only
- Know what Struts is all about
- View the flow structure of a Struts-based Web application

Exercises

The exercises in this module must be completed in order.

- "Exercise 5.1: Invoking the Web Interaction wizard" on page 58
- "Exercise 5.2: Specifying the input and output page" on page 59
- "Exercise 5.3: Defining the iSeries GETDATA program invocation and parameters" on page 61
- "Exercise 5.4: Defining the iSeries GETDATAS service program invocation and parameters" on page 66
- "Exercise 5.5: Defining the input page content" on page 72
- "Exercise 5.6: Defining the output page content" on page 74
- "Exercise 5.7: Specifying error handling" on page 76
- "Exercise 5.8: Enhancing the output page" on page 78
- "Exercise 5.9: Ensuring the customer number field cannot be modified" on page 82
- "Exercise 5.10: Visualizing the flow structure of your Web application" on page 84

Time required

This module will take approximately 30 minutes to complete.

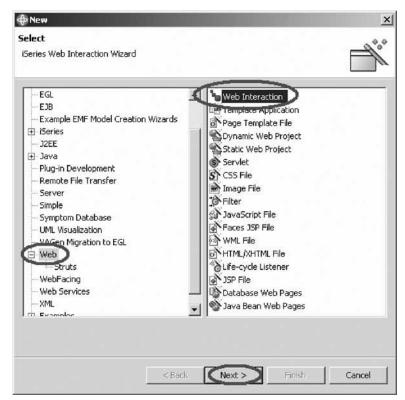
Exercise 5.1: Invoking the Web Interaction wizard

During this exercise you will use the Web 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.

To invoke the Web Interaction wizard:

😼 Project Nav	igator 💌 🗙	Welcome X
(c)		Development Studio Client Advanc
E-1 WSSLAP E-1 WSSLAP	New Go Into Go To Open in New Window	M HTML/XHTML File
	Copy Paste Delete Move Rename	 M Page Template File M JavaScript File M CSS File M Image File
Project Naviga	Import Export	Servlet
🚺 Page Data	Rebuild Project	Project
No data compor	Refresh	Other

- 1. Right-click WSSLABxx in the Project Navigator.
- 2. Click **New** > **Other** on the pop-up menu.
- 3. Select **Web** from the left pane.



4. Select Web Interaction from the right pane.

5. Click Next.

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. The Specify a Name and Location for your Web Interaction page opens.

🕀 Web Interaction					×
	Location for your Web Interaction location for the resources created for this W	/eb Interaction.			°°
Destination folder:	/WSSLABxx/WebContent				Browse
Web Interaction name	custInquiry				
Java package prefix:	com.ibm.wsslabxx				
Vse error page	/error.jsp				Browse
Invalidate session	after the interaction occurs				
			< Back	ext > Firish	Cancel

- 6. Type custInquiry in the Web Interaction name field.
- 7. Click Next in the Web Interaction wizard.

The Specify the Input and Output Pages for your Web Interaction page opens.

You have started the Web Interaction wizard and now you are ready to begin "Exercise 5.2: Specifying the input and output page."

Exercise 5.2: Specifying the input and output page

Before you begin, you must complete "Exercise 5.1: Invoking the Web Interaction wizard" on page 58.

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 displays the results of interaction. The output page shows in the browser at the end of the interaction. The wizard generates both Web pages for you.

Web Interaction		
cify the Input and Output Pages for your u can use previously designed input and output	r Web Interaction pages for this Web interaction or you can have the wizard generate them.	°0(
Generate input page Use input page(s):	Page preview:	
custinguiryInput.jsp	Add Remove Preview	
1.0	Rename	
Generate output page Use output page(s):		
custInquiryResults.jsp		
	Remove	
	Preview	
	Preview Rename(C)	
Generate input/output page(s) using the page	Rename(C)	
Generate input/output page(s) using the page	Rename(C)	

- 1. Click Generate input page radio button.
- 2. Click Generate output page radio button.
- 3. Select custInquiryInput.jsp and click Rename.

💠 Input JSP File		×
Specify a name for the input JSP fil	e	
inquiry.jsp		
	ОК С	ancel

- 4. Enter inquiry.jsp as the new input JSP file name.
- 5. Click OK.
- 6. Select custInquiryResults.jsp and click Rename.
- 7. Enter result.jsp as the new output JSP file name.
- 8. Click OK.
- 9. Click Next.

The Specify the Input and Output Parameters for your iSeries Host Program page opens.

You have created an input page that invokes the interaction and an output page that displays the results of the interaction and now you are ready to begin "Exercise 5.3: Defining the iSeries GETDATA program invocation and parameters" on page 61.

To define the iSeries GETDATA program invocation and parameters go to "Exercise 5.3: Defining the iSeries GETDATA program invocation and parameters" on page 61.

To define the iSeries GETDATAS service program invocation and parameters go to "Exercise 5.4: Defining the iSeries GETDATAS service program invocation and parameters" on page 66.

Exercise 5.3: Defining the iSeries GETDATA program invocation and parameters

Before you begin, you must complete "Exercise 5.2: Specifying the input and output page" on page 59.

The next page asks you to define the parameters you want to pass between the Web pages and the iSeries program that processes the requests.

Here you will re-use the PCML program interface definition that was created when you compiled your source member.

Use an iSeries program or procedure	C Use a Java bean	C No p	rogram call
rogram call definitions	Add Program Add <u>Pars</u> Add Program Program alias: Program object: Library: Program type: Entry point: CCSID of entry point: Return type: Thread safe: Source location: I Associate this program	*PGM void false	Browse
Import PCML Import Source			Cancel

Getting the program interface definitions

To get the program interface definitions that you want to call on the iSeries:

1. Click Import **PCML**

The Import PCML dialog opens.

CML file: Remote	tte File System	2	
elect programs or structures to import		of PCML file	N
Select All Deselect All			

- 2. Click the arrow on the Select PCML file from field
- 3. Select **Remote File System** from the list. The **Browse For File** dialog opens.

New Connection		
⊕- ¹ Local		-
🗊 💁 s400a		
TE-品 /QDLS		
由品 /QFileSvr.400 由品 /QNTC		
田 品 /QOpenSys		
画品/QOPT		
Ⅲ—品 /QSYS.LIB		
😟 - 🧰 bin		
⊞- 🛅 dev		
⊞- ∰ etc ⊞- ∭ home		
te-tin lib		
E opt		
CIBM		
🖭 🗇 🗇 QSR		
і ф 👄 отсятим		-
	OK	Cancel

4. Expand the tree view of the IFS drive to the directory where you stored the PCML file generated by the RPG compiler:

Since you followed the instructions and used a program call, it should be in ${\tt QIBM/WSSLABxx/getdata.pcml}$

/QIBM/WSSLABxx/getdata.pcml	
i etc e- î home	
⊕- 🚰 home	
Ē- Ē līb	
😐 🛅 opt	
🕀 🗁 cheng	1
🗈 - 🛅 home	
i⊞ include	
dia locales The ProdData	
ProdData QIBM	
E ChweyerEAR	
E ci test	
E Gar UserData	
WSSLABXX	
getdata.pcml>	
+- C XML	
webxml.dtd	-
	·
	<u>C</u> ancel
<u>_</u>	Cancer
	•

- 5. Select the correct file.
- 6. Click OK.
- 7. The Import PCML dialog appears with a list of the program interface constructs contained in the PCML file.

Remote Fil	e System
weissT40.to	pras48f:/QIBM/wsslabxx/getdata.pcml
s to import	Tree view of PCML file
inputOutput	CSTRUC (GETDATA_CSTRUC_Inpu CUSTNO CUSTNA REPNO CONTAC CONTAC CPHONE CFAX CADDR CCITY CCOUNT CZIP CZIP CZIP
Deselect All	
	weissT40.to

• A data structure

- A call interface
- 8. Click Select All.
- 9. Click OK.

Specifying the input and output parameters

The Specify the Input and Output Parameters for your iSeries Host Program page opens again, now containing the program description you just imported.

Jse an ISeries program or procedure	C Use a Java bean	C No program call	
ogram call definitions	Add Program	d Structure	
	Edit Parameter Parameter name: CUSTNOI Data type: character Structure name: Length: 7	• •	
	Precision: Count: Usage: Initial value: Advanced		
· ·	Specify database reference field Show database field definition	Specify Synchroni	ze
Import PCML Synchronize All			Cancel

- Expand GETDATA in the left pane under **Program call definitions**. The information from the PCML file gets included in the right pane and the parameters defined for this program appear.
- 2. Select the first parameter CUSTNOI.
- 3. In the right pane, change the **Usage** field from Input/Output to **Input** to indicate that this parameter is used to pass a value from the Web page to the program.
- 4. Click OK

e this page to define the input and output pa	rameters for your iSeries host program.		0
CT IT CETDATA	C Use a Java bean Add Program Add Parameter Add Edit Parameter Parameter name: CSTRUC Data type: structure Structure name: GETDATA_CSTRUC Length: Precision: Count: Usage: output Initial value: Advanced	C No program call	
Import PCML Synchronize All	Specify database reference field Show database field definition	Specify	Synchronize

- 5. In the left pane select the CSTRUC parameter.
- 6. In the right pane, change the **Usage** field from Input/Output to **Output**, indicating that this parameter is used to pass a value from the program to the result Web page.
- 7. Click OK.
- 8. Do the same for the **FEEDBACK** parameter.

ise an Geries program or procedure ogram call definitions	C Use a Java bean	○ No program call ameter Add Structure	
CETIDATA_CSTRUC_InputOutput	Edit Program Program alias: Program object: Likeary: Program type: Entry point: CCSID of entry point. Reburn type: Thread safe: Source location: This sociate this program	GETDATA GETDATA GETDATA WSSLABOX *PGM *PGM folse	Brogse
Import PCM Synchronize Al	-		(2) <u>200</u>

- 9. Now select GETDATA in the left pane under **Program call definitions**. This will fill the right pane with the program information.
- 10. In right pane, select the check box **Associate this program with the interaction**.
- 11. Click **OK**.
- 12. Click Next.

You have defined the program interface definitions and now you are ready to begin "Exercise 5.5: Defining the input page content" on page 72.

Skip the next exercise and continue with "Exercise 5.5: Defining the input page content" on page 72.

Exercise 5.4: Defining the iSeries GETDATAS service program invocation and parameters

Before you begin, you must complete "Exercise 5.2: Specifying the input and output page" on page 59.

The next page asks you to define the parameters you want to pass between the Web pages and the iSeries service program that processes the requests.

Here you will re-use the PCML program interface definition that was created when you compiled your source member.

Use an iSeries program or procedure	🔿 Use a Java bean	C. No progr	am call
rogram call definitions	Add Program Add Eard	Minister	
	Program alias: Program object: Library: Program type: Entry point: CCSID of entry point: Return type: Thread safe: Source location:	PGM PGM Fold Fold Folde Folde Folde Fold Fold Fold Fold Fold Fold Fold Fold	Browse
Import PCML Import Source Synchronize All			Cancel

Getting the program interface definitions

To get the program interface definitions that you want to call on the iSeries:

1. Click Import **PCML**

The Import PCML dialog opens.

ielect PCML file from:				(
CML file:	iepate iote File Sys	stem			
elect programs or structures to import		Tree view of PO	IML file	N	
Select All Desele	et all				
Desete	JC 740				

- 2. Click the arrow on the Select PCML file from field
- 3. Select **Remote File System** from the list. The **Browse For File** dialog opens.

elect a file		
New Connection		
🕂 🖳 Local Till s400a		
回島 /		
□ ඛ / 田 晶 /QDLS		
田晶 /QFileSvr.400		
田晶 /QNTC		
田 品 /QOpenSys		
回品/QOPT		
画品 /QSYS.LIB		
⊞-		
🗄 🛅 dev		
i ⊡- 🛅 home		
tin in the second seco		
🚊 🗇 opt		
E CI QSR		
п са отсятим		
	OK Ca	ancel

4. Expand the tree view of the IFS drive to the directory where you stored the PCML file generated by the RPG compiler:

Since you used a service program, it should be in QIBM/WSSLABxx/getdatas.pcml.

opt	
CD QIBM	
🛨 🗄 🗁 cheng	
🕀 🗁 home	
😟 🛅 include	
🕀 🧰 locales	
🕀 🗁 opt	
😟 🛅 ProdData	
⊞ 🛅 QIBM	
🕀 🧰 SchweyerEAR	
⊞- C test	
⊕ ☐ UserData	
E WSSLAB02	
getdatas.pcml	
H-CA XML	
D webxml.dtd	

- 5. Select the correct file.
- 6. Click OK.
- 7. The Import PCML dialog appears with a list of the program interface constructs contained in the PCML file.

ielect PCML file from:	Remote File	System 💆
CML file:	YURICH.s40	Da:/QIBM/WSSLABxx/getdatas.pcml
elect programs or structure	s to import	Tree view of PCML file
☑ ⊶ GETRECORD_CSTRL	IC_InputOutput	CUSTI (GETRECORD_CSTRUC_Inp CUSTNO CUSTNA CUSTNA CUSTNA CONTAC CONTAC CPHONE CADDR CADDR CADDR CCITY CCOUNT CCOUNT CZIP CCIPLO
Select All	Deselect All	
Select All	Deselect All	

- A data structure
- A call interface
- 8. Click Select All.
- 9. Click OK.

Specifying the input and output parameters

The Specify the Input and Output Parameters for your iSeries Host Program page opens again, now containing the program description you just imported.

Use an iSeries program or procedure	C Use a Java bean	C No program call
ogram call definitions 판 국, GETRECORD_CSTRUC_InputOut	Add Program Add Parameter Add	Structure
GETRECORD CUSTNOT CUSTIT (GETRECORD_CSTRU	Edit Parameter Parameter name: CUSTNOI Data type: character Structure name:	
	Advanced Specify database reference field	Specify Synchronize
	Show database field definition	Show
Import PCML Import Source Synchronize All		

- 1. Expand GETRECORD in the left pane under **Program call definitions**. The information from the PCML file gets included in the right pane and the parameters defined for this program appear.
- 2. Select the first parameter CUSTNOI.
- 3. In the right pane, change the **Usage** field from Input/Output to **Input** to indicate that this parameter is used to pass a value from the Web page to the program.
- 4. Click OK

Web Interaction cify the Input and Output Parameters	for your iSeries Host Program		
e this page to define the input and output pa			0
Use an iSeries program or procedure	C Use a Java bean	C No program call	
rogram call definitions	Add Program Add Parameter Add 1	Structure	
CUSTNOT CUSTN		• • •	
12	Specify database reference field	Specify,	Synchronize
	Show database field definition	Show,	
Import PCML Synchronize All			Concel
		Back Next >	Finish Cancel

- 5. In the left pane select the CUST1 parameter.
- 6. In the right pane, change the **Usage** field from Input/Output to **Output**, indicating that this parameter is used to pass a value from the program to the result Web page.
- 7. Click OK.
- 8. Do the same for the **FEEDBACK** parameter.

Use an iSeries program or procedure ogram call definitions	C Use a Java bean	C No program call	
GETRECORD CSTRUC_Inputout	Edit Program Program alias: Program object: Library: Program type: Entry point: CCSID of entry point: Return type:	GETDATAS GETDATAS WSSLAB::: *SRVPGM getrecord void false	Browse
Import PCML Import Source			

- 9. Now select GETRECORD in the left pane under **Program call definitions**. This will fill the right pane with the program information.
- 10. Change the name of the service program to GETDATAS.
- 11. Select the Program type *SRVPGM.
- 12. In right pane, select the check box **Associate this program with the interaction**.
- 13. Click OK.
- 14. Click Next.

You have defined the program interface definitions and now you are ready to begin "Exercise 5.5: Defining the input page content."

Exercise 5.5: Defining the input page content

Before you begin, you must complete "Exercise 5.3: Defining the iSeries GETDATA program invocation and parameters" on page 61or "Exercise 5.4: Defining the iSeries GETDATAS service program invocation and parameters" on page 66.

Now you need to specify which of the input parameters you want to see on the input page for this interaction that gets generated by the wizard. This is easy in this application since you only have the customer number input field to handle. You also have the capability to enhance the page layout by changing colors of the page background and changing the heading. The Design the Input Form page contains a properties table that shows all input capable parameters, and a sample of the Web page that will be created.

esign the Input Fo Specify the page prop		it parameters that the	e generated Web pa	ge will expose to the user	for input.	°°G
Input parameters:				Customa	r Inquiry	*
		Move up Move down Select All Deselect All	CUSTNOL		rinquiry	
Property Page Title Background Color Title Color Form Bean Scope	Value Eustomer Inquiry RGD (200, 200, 2					
Page Fields						*

Changing the heading

To change the heading:

- 1. Click the field beside the **Page Title** property in the Property table and type Customer Inquiry.
- Press Enter to apply this change to the sample page. Now you want to change the customer number field properties.
- 3. Click the **Fields** tab on the Properties table.

This opens the field properties for the selected field in the list.

Changing a field label

To change a field label:

Input parameters:		
₽ j [‡] custnoi	Move up Move down Select All Deselect All	Customer Inquiry Enter customer number
Property	Value	
Component Name:	CUSTNOL	1
	Enter customer number	
	Character(7)	
Data Type:		
Data Type: Input Style:	Text Entry	
Data Type: Input Style: Input Choices:		
Data Type: Input Style: Input Choices: Size:	7	
Data Type: Input Style: Input Choices: Size: Name in Session:	7 CUSTNOI	
Data Type: Input Style: Input Choices: Size: Name in Session: Restore from Session:	7 CUSTNOI false	
Data Type: Input Style: Input Choices: Size: Name in Session: Restore from Session: Save to Session:	7 CUSTNOI false false	
Data Type: Input Style: Input Choices: Size: Name in Session: Restore from Session:	7 CUSTNOI false	

- 1. In the left pane of the Design the Input Form page, select CUSTNOI.
- 2. Click the field beside the Label property, and type Enter customer number.
- 3. Press Enter.
- 4. Click Next to advance to the next wizard page.

You have specified the input and output parameters and the input parameters to show on the input page and now you are ready to begin "Exercise 5.6: Defining the output page content."

Exercise 5.6: Defining the output page content

Before you begin, you must complete "Exercise 5.5: Defining the input page content" on page 72.

Now you can specify what data should be displayed on the output Web page generated for you. In the right pane of the wizard you see a sample of the page that will be generated. First, change the Input style of the output parameter CSTRUC.CUSTNO. This will enable you to use customer number as an input parameter to future features of your Web project.

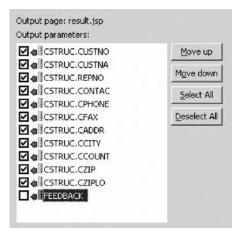
To define the output page:

utput page: result.jsp utput parameters:		Move up Move down Select All	CUSTNO: CUSTNA:	2
	Value	Deselect All	REPNO: CONTAC: CPHONE: CFAX: CADDR: CCITY: CCOUNT:	
roperty Component Name: Label: Data Type: Input Style: Input Choices:	CUSTNO CUSTNO cherenter(Z) Text Entry		CZIP: CZIPLO: FEEDBACK:	
Size: Name in Session: Save to Session: Message Controller:	7 CUSTNO false false			

- 1. Select CSTRUC.CUSTNO.
- 2. Click the Fields tab.
- 3. From the Input Style list, select Text Entry.

By default all output parameters will be added to the Web page, but you don't want the data in the FEEDBACK parameter to be displayed.

Let's go through the steps to erase the FEEDBACK parameter from the output page.



4. In the left pane of the Design the Result Form, clear the **FEEDBACK** check box. This will remove it from the Web page.

You will stay on this page to specify the error handling.

You have specified what data to show on the output page and now you are ready to begin "Exercise 5.7: Specifying error handling."

Exercise 5.7: Specifying error handling

Before you begin, you must complete "Exercise 5.6: Defining the output page content" on page 74.

Now you can specify what should happen in case a customer number didn't match the customer numbers in your database.

1. In the left pane of the Design the Result Form, select the **FEEDBACK** parameter.

You will see the properties for this variable in the Properties table.

Property	Value	1
Component Name:	FEEDBACK	
Label:	FEEDBACK	
Data Type:	character(20)	
Input Style:	Label	
Input Choices:		
Size:	20	
Name in Session:	FEEDBACK	
Save to Session:	false 🖉	-
Message Controller:		I)

2. Locate the Message Controller heading in this Properties table.

Message controller	false True		
Parameter Value	Message Id	Web Component	New
			Delete
			Move up
			Move dow
			1

- 3. Click the field beside the Message Controller and set it to true.
- 4. Click New.
- 5. Set the **Parameter Value** to *OTHER from the list.

Parameter Value	Message Id	Web Component	New
THER	*SELECT	inquiry.jsp-label	Delete
	*SELECT		Move up
			Move dow

- 6. Set the **Message ID** to ***SELECT** from the list. This opens a Message Selection Window.
- 7. Drill-down to s400a > WSSLABXX/MYMSGF > MYMSGF > CUS0001

IS0001: Customer number &1 not valid	Property	Value
Rew Connection	ID ID	CUS0001
s400a	Message help	
	Severity	0
E - + *LIBL	Text	Customer number &1 not valid
SSLABXX/MYMSGF OBJTYPE(*MSGF:*)		
MYMSGE,*msaf		
CUS0001: Customer number &1 not		
Costoner number & not	dilu	
17		
		DK << Details Cano

- 8. Click OK.
- 9. Leave the default value of the Web Component.
- 10. Click New.
- 11. Enter 0 as the **Parameter Value**.

Parameter Value	Message Id	Web Component	New
*OTHER	CU50001	inquiry.jsp-*DEFAULT	
0	*NONE	result.jsp-*DEFAULT	Delete
			Move up
			Move down

- 12. Press Enter.
- 13. Set **Message ID** to *NONE using the list.
- 14. Set Web Component to result.jsp-DEFAULT using the list.
- 15. Click OK.

Here is what you just did:

- If the return value of the FEEDBACK parameter is **0**, display the customer details in the **result.jsp** Web page.
- Any other return value in the FEEDBACK parameter, use the message ID CUS0001 in the specified message file, place the Customer number input in the message substitution variable, and display the Web page **inquiry.jsp** with the error message CUS0001.

The message handling is defined.

You have specified what will happen when an incorrect customer number is entered in the customer number field and now you are ready to "Exercise 5.8: Enhancing the output page."

Exercise 5.8: Enhancing the output page

Before you begin, you must complete "Exercise 5.7: Specifying error handling" on page 76.

Next you enhance the look of the Web page.

To enhance the output page:

1. Click the **Page** tab in the Properties table.

🖶 Web Interaction		×
Design the Result Form Specify the page properties and the output pa	ameters that the generated Web page will display to the user as output.	°°
Image: Struc.CUSTNA Mgs Image: Struc.REPNO Mgs Image: Struc.CONTAC See Image: Struc.CPHONE See	Ave up e down ket Al eket Al cUSTNO: CUSTNA: REPNO: CONTAC: CPHONE: CFAX: CADDR: CCTTY: CCOUNT: CZIP: CZIPLO:	I
Page Fields	< Back Next > Einish	Cancel

- 2. Click the field beside the Page title and type Customer Details.
- **3**. Click the field beside the **Background Color** and click the selection button on the right side of the field.

The Color palette opens.

4. Select a **light yellow** from the Color palette.



- 5. Click **OK** on the Color palette.
- 6. Click the field beside the **Title Color** and click the selection button on the right side of the field.
- 7. Select **Blue** from the Color palette.
- 8. Click **OK** on the Color palette.

You notice the sample page gets updated as you choose different properties. If you don't like the suggested color choices, go ahead and create your own design.

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

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

Output page: result.jsp Output parameters: P = Stuc.CUSTNO P = Stuc.CUSTNO P = Stuc.CONTAC P = Stuc.CONT	Mgve Sele	et All Custom Represe Contact	er Number er Name entative Number Number mber:	mer Details	×
Property Component Name: Label: Data Type: Input Style: Input Choices: Size:	Value CLISTAIO Customer Number characcer(7) Text Enbry 7	Zip:	r.		
Name in Session: Save to Session: Message Controller:	CUSTNO false false				×

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

11. In the left pane of the Design the Result Form page, clear the **struc.CZIPLO** check box. You don't need to display this variable.

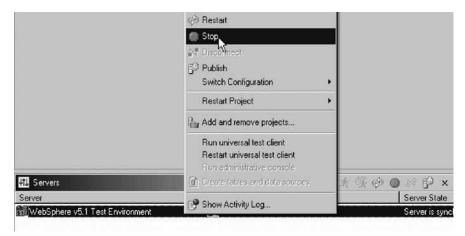
Click **Finish** on the Design the Result Form page.
 The Interaction wizard creates the necessary files for your application:

- The file **custInquiry.wit** is created in the project. This file reflects the data you entered on the pages of the Web Interaction wizard for the interaction you identified as cust. To modify these settings, double-click on the file to open the Web Interaction wizard.
- The default package under the Java Source folder contains the .java files generated by the wizard.
- The classes folder under Web Content/WEB-INF contains the .class files generated from the .java files that were created by the wizard.

Note: If you encounter the following error message in your Tasks list:

27	Ta	asks	(1 item)	* 4 8 8 *	×
	1	1	Description	Resource	In
0			The project was not built due to "Problems encountered while deleting resources.",	WSSLABxx	
-					
					+
-					-
C					

- 1. Click the **Server** tab to access the Server view.
- 2. Right-click the test server and select Stop on the pop-up menu.



- 3. Switch to your Project Navigator and right-click your project folder.
- 4. Click **Refresh** on the pop-up menu.

😼 Project Naviga	an (中中 € ¥ 👉 ▼ 🗙
E 😂 WSSLAB	
Web	New
- 🐨 Web	Go Into
🕀 🔂 Java	GoTo
⊡ 🥔 Web(⊡ 🕞 M	Open in New Window
± @ ₩	Сору
1 🕀 🗁 W	Paste
	Delete
Project Navigator	Move
Page Data	Rename
No data compone	Import
	Export
	Rebuild Project
	🕼 Refresh

5. Right-click again your project folder and click Rebuild Project.

🖬 Project Navi	galor 🗇 🗢 🗧 🗶 🧬 💌 🗙
E 😂 WSSLAB	
We We	New
🐨 We	Go Into
😟 🔂 Jav	Go To
⊡ 🖉 We	Open in New Window
± @= [🖹 Сору
• 🗁	Paste
	Delete
Project Navigal	Move
🚺 Page Data	Rename
No data compor	Import
	Export
	Rebuild Project
	2

This should remove the error message in your Task view.

You have changed the background color of the output page and now you are ready to begin "Exercise 5.9: Ensuring the customer number field cannot be modified."

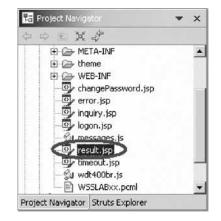
Exercise 5.9: Ensuring the customer number field cannot be modified

Before you begin, you must complete "Exercise 5.8: Enhancing the output page" on page 78.

Let's make sure that the Customer Number field on your results.jsp page cannot be modified.

To ensure the customer number field cannot be modified:

1. In the Project Navigator, expand WebContent and locate result.jsp.



2. Right-click result.jsp.

00 E X 2	
$\gamma = \gamma + \gamma$	
META-INF Heme WEB-INF WEB-INF ChangePassword.jsp Gy error.jsp	
- Dy inquiry.js New - Dy logon.jsp Go To - Su message Open	
Tresult.is: Open With	
VSSLAB: Project Navigator Strut Delete	

3. Click **Open** on the pop-up menu.

This opens the result.jsp page on the Page Designer.

👫 Project Navigator 🔅 🖘 🙀	* • ×	U) Welcome	*result.jsp	×		
Ger META-INF Ger theme	<u>*</u>	result.jsp - Untit	led *		• [4	IWCL:WTEXTE
George WEB-INF George Password.jsp George error.jsp		Session se	2			
inquiry.jsp	-		(Custor	mer Det	ails
Project Navigator Struts Explorer						
👔 Page Data 📼 📼	% • ×	%result labe	l Customer N	lumber_	June 1	
	-	5	L_Customer_N		CUSINA	
Page Data Client Data Gallery			l_Representati			
O Attributes IWCL:WTEXT	ENTRY • ×	%result lab		··· · ······	CONTAC	
General Data Events Styles		%resul lab	l_Phone_Num	lber	CPHONE	
Name: CUSTNO	-1	%result lab	l_Fax_Numbe	r	CFAX	
Input type: Text		%result.lab	l_Address		CADDR	
Size: 7		%result.lab	l_City		CCITY	
Initial state: Dicabled		%result labe	l_Country		CCOUNT	
Read-only		%result.labe	l_Zip	·····	CZIP	
Orientation: (Default)	-	Nucht				
Initial value:			tton.submit	%result.butt	on.reset	
Tool tip:		Design Source	Preview			

- 4. Select the **Text Entry** field of Customer Number in Page Designer. Notice the **Attributes** view of that Text field appears.
- 5. Select the **Read-only** check box.

This prevents any modifications to the Customer Number.

6. Save your changes and close result.jsp.

You have made sure that the customer number field is read only and now you are ready to begin "Exercise 5.10: Visualizing the flow structure of your Web application."

Exercise 5.10: Visualizing the flow structure of your Web application

Before you begin, you must complete "Exercise 5.9: Ensuring the customer number field cannot be modified" on page 82.

A Web diagram is a view that helps you visualize the flow structure of a Struts-Based Web Application. Because of the indirectness involved with a Struts application, being able to visually see the application's flow can help you to better understand the application.

Struts

Before you create a Web diagram, let's first look at what Struts is all about. First, Struts is a set of Java classes and JSP tag libraries that provide a conceptual framework for developing Web applications. The Struts technology is open source and was developed as part of the Apache Software Foundation's Jakarta project.

Second, Struts provides numerous, custom JSP tags that are simple to use but are powerful in the sense that they hide information. The Page Designer does not need to know much about form beans, for example, beyond the bean names and the names of each field in a given bean.

Third, you can use the diagram editor to show all or part of a Struts application. For example, suppose you have a three-part Struts application. One part handles the login process, one part handles product inquiries, and a third part handles product updates. In this case you could draw three diagrams to represent this system, or you could draw the entire system in a single diagram. Because one diagram can be included inside another, it would probably make more sense to represent this Struts application using a set of three diagrams.

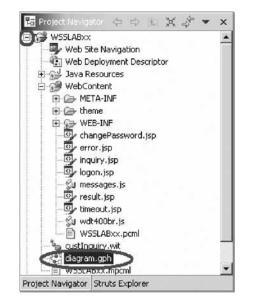
Now, that you know what Struts is, you can go ahead with this exercise.

You will now create a Web diagram that will show you the visual components of the Web application you just created.

An empty Web diagram (diagram.gph) should have been generated for you during project creation.

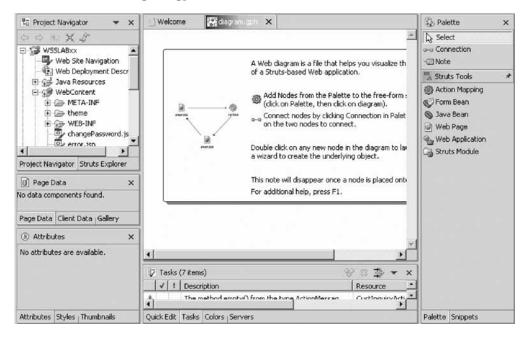
To create a Web diagram:

1. If you don't see it (diagram.gph) in the editor, in the Project Navigator view, expand WSSLABxx.



2. Locate and double-click diagram.gph.

You will now see diagram.gph in the editor.

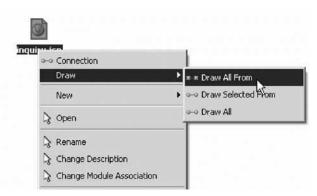


Now, let's populate the Web diagram:

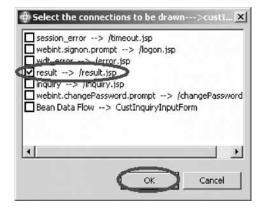
- 1. Locate inquiry.jsp under WSSLABxx and then WebContent in the Project Navigator.
- 2. Click inquiry.jsp and drag it to diagram.gph. You will notice the cursor change

from 🛇 to 🗟

3. In the Web diagram, right click inquiry.jsp.

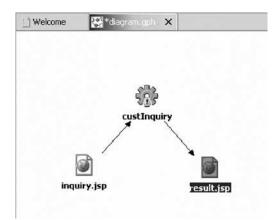


- Click Draw > Draw All From on the pop-up menu. You will now see the custInquiry interaction.
- 5. Right-click custInquiry and click Draw > Draw Selected From.
- 6. Select the result.jsp check box



7. Click OK.

You should now see the graphical representation of the Web interaction that you've created in a previous exercise.



8. Save and close diagram.gph.

You have seen the flow structure of a Struts-based Web application.

Module recap

You have completed Chapter 5, "Module 5. Creating a Web application," on page 57. You have learned how to:

- Start the Web Interaction wizard
- Create an input page that invokes the interaction
- Create an output page that displays the results of the interaction
- Get the program interface definitions
- Specify the input and output parameters
- Specify the input parameters to show on the input page
- Specify what data to show on the output page
- · Specify what will happen when an incorrect customer number is entered
- Change the background color of the output page
- Make sure the customer number field is read only
- Know what Struts is all about
- View the flow structure of a Struts-based Web application

You have created your Web application. Continue to Chapter 6, "Module 6. Running the Web application," on page 89.

Chapter 6. Module 6. Running the Web application

This module teaches you how to test the Web application locally without deploying the files in the Web project to the host server. Testing your Web application locally includes using the iSeries Web Tools Run-time Configuration wizard to specify the authentication and run-time values for every host program or procedure call made by any Web interaction in the Web project. This is something you have already done in Chapter 2, "Module 2. Creating an RPG program," on page 7 or Chapter 3, "Module 3. Creating the RPG service program for your Web application (optional)," on page 25.

In this module, you will:

- · Check the Web perspective is open
- Locate input page of the Web application
- Invoke the Web application to run in the WebSphere Test Environment
- Test the Web application

Exercises

The exercises in this module must be completed in order:

- "Exercise 6.1: Opening the Web perspective"
- "Exercise 6.2: Finding the Web application and running it" on page 90

Time required

This module will take approximately **10 minutes** to complete.

Exercise 6.1: Opening the Web perspective

To open the Web perspective:

1. Click the Web perspective 🕥 icon, on the left taskbar of the workbench.



2. If there is no Web perspective icon, select **Window** > **Open Perspective** > **Web** from the workbench menu.

You have checked that the Web perspective is open and now you are ready to begin "Exercise 6.2: Finding the Web application and running it."

Exercise 6.2: Finding the Web application and running it

Before you begin, you must complete "Exercise 6.1: Opening the Web perspective" on page 89.

To test your application locally using the WebSphere test environment:

- 1. In the Project Navigator, expand your project WSSLABxx.
- 2. Expand the WebContent folder.
- 3. Right-click inquiry.jsp and click Run on Server on the pop-up menu.

🖻 🥔 WebContent		
😟 🗁 META-INF		
😟 🦢 theme		
😟 🗁 WEB-INF	ALCONOMIA -	
changePassv	word.jsp	
error.jsp		
- 9 Inquiry isc - 9 logon isp	New	
Su messages.	Go To	•
result.jsp	Open	
timeout.jsj	Open With	•
🖗 wdt400br.	Copy	
WOOLADX>	Paste	
- 💊 custInquiry.wi	Delete	
WSSLABxx.mp	Move	
Project Navigator Struts	Rename	
Page Data	Kenamen.	
No data componente foune	Import	
Page Data Client Data 🤅	Export	
④ Attributes	Add Bookmark	
No attributes are available	Rebuild Project	
	Refresh	
	Run Validation	
8	🗐 Debug on Server	
2	🗸 Run on Server	
0	Profile on Server	
	Convert to XHTML	
	Validate HTML Syntax	

You will see a message to select the Server to run in the Test Environment.

Server Selection	A DOOR DOOL ST	2
Server selection Select which server to launch.		
C Use an existing server		
Server	Status	
J Status:		
Create a new server		
Folder: Servers		•
Server type: Server type: Server type: Server type: Server Server type: Server Server Server Server Server Server Server type: Server S	Server	*
Description: Runs J2EE projects o	ut of the workspace on the loca	al test environment.
Det server as project default (do no	ot prompt)	
< Back	Next > Finish	Cancel

- 4. Make sure Test Environment is selected in the Server type list.
- 5. Select the Set server as project default (do not prompt) check box.
- 6. Click Finish to configure and launch the server.

After the WebSphere server has started, your Web application input page will display in the workbench browser.

🕒 Welcome 🔮 Web Browser 🗙	
http://localhost:9080/WSSLA8xx/inquiry.jsp	3 🔳
Customer Inquiry	<u></u>
Enter customer number: 0010100	
Submit Reset	
Done	*

- 7. Type 0010100 as the **Customer Number**.
- 8. Click Submit.

This launches the generated output or result page.

a. If you see a Page not found error, go to the Servers view.



b. Restart the server instance. Then go back and try the **Run on Server** option again; the Server sometimes needs to be restarted to pick up the changes.

You will see the Result page with customer data appearing in the workbench browser.

Cus	tomer Details	
Customer Number:	0010100	
Customer Name:	Meridien Electronics Limited	
Representative Numb	er: 43443	
Contact:	Alfredo Bayonne	
Phone Number:	206-865-4027	
Fax Number:	206-865-4037	
Address:	10423 S.E. 30th Place	
City:	Bellevue, WA	
Country:	U.S.A.	
Zip:	98007	

Next, test the message handling.

- 9. Click the 🔄 Back button in the browser to get back to the first page.
- 10. Enter a wrong customer number, for example, 9999.

	🕙 Web Browser 🛛 🗙		
http://localhost	:9080/WSSLABxx/inquiry.jsp	. 0 ⇔ ⇒ ● ∲ ≙	
	0	- for an effect	~
	Custome	er Inquiry	
Enter cus	tomer number: 9999		
Cubmit	Reset		
Submit	Resel		
			*

11. Click Submit.

An error message appears.

	🔮 Web Browser 🛛 🗙			
http://localhost	:9080/WSSLABxx/custIn	quiry,do	• • •	0 🗇 🗛 🛛 🔳
	Custo	mer In	quiry	Ă
Enter cus	tomer number:			
Submit	Reset			
Customer	number 9999 not	valid		
Done				*

This capability is generated by the Web 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.

You have found and run the Web application.

Module recap

You have completed Chapter 6, "Module 6. Running the Web application," on page 89. You have learned how to:

- Check the Web perspective is open
- Locate input page of the Web application
- Invoke the Web application to run in the WebSphere Test Environment
- Test the Web application

Now that you have run your Web application you learn how to debug your Web application. Continue to Chapter 7, "Module 7. Debugging a Web application," on page 95.

Chapter 7. Module 7. Debugging a Web application

This module teaches you how to debug a Web application. You will learn how to set up the debug environment, use a service entry point, add and remove breakpoints, display a variable, and view the call stack.

You will debug a Web application that invokes the iSeries GETDATA program.

In this module, you will:

- · Recognize the features of the iSeries Integrated Debugger
- · Start a debug session with service entry points
- Set breakpoints
- Monitor an expression
- Expand a thread to show every program, module, procedure and method on the call stack at the current execution point
- Run the program to termination

Exercises

The exercises in this module must be completed in order.

- "Exercise 7.1: Introducing the iSeries Integrated Debugger"
- "Exercise 7.2: Starting a Debug session using service entry points" on page 96
- "Exercise 7.3: Adding and deleting breakpoints" on page 97
- "Exercise 7.4: Displaying a variable" on page 98
- "Exercise 7.5: Viewing the call stack" on page 99
- "Exercise 7.6: Closing the debug session" on page 99

Time required

This module will take approximately 15 minutes to complete.

Exercise 7.1: Introducing the iSeries Integrated Debugger

The Integrated Debugger is a source-level debugger that enables you to debug and test an application that is running on an iSeries system. It provides a functionally rich interactive graphical interface that allows you to:

- View source code or compiler listings, while the program is running on the iSeries system.
- Set, change, delete, enable and disable line breakpoints in the application program. You can easily manage all your breakpoints using the Breakpoints view.
- Set watch breakpoints to make the program stop whenever a specified variable changes.
- View the call stack of your program in the Debug view. As you debug, the call stack gets updated dynamically. You can view the source of any debug program by clicking on its call stack entry.
- Step through your code one line at a time.
- Step into or step over program calls and ILE procedure calls.

- Display a variable and its value in the Monitors view. The value can easily be changed to see the effect on the program's execution.
- Locate procedure calls in a large program quickly and easily using the Modules/Programs view.
- Debug multithreaded applications, maintaining separate stacks for each thread with the ability to enable and disable any individual thread.
- Load source from the workstation instead of the iSeries useful if you don't want the source code on a production machine.
- Debug client/server and distributed applications.

The Debugger supports RPG/400 $^{\rm \circledast}$ and ILE RPG, COBOL and ILE COBOL, C, C++ and CL.

In the following exercises you will be given the opportunity to learn about some of the basic features of the Debugger.

You have been introduced to the debugger and now you are ready to begin "Exercise 7.2: Starting a Debug session using service entry points."

Exercise 7.2: Starting a Debug session using service entry points

Before you begin, you must complete "Exercise 7.1: Introducing the iSeries Integrated Debugger" on page 95.

In this exercise you will use service entry points to debug a Web application. The service entry point feature is designed to allow easy debugging of Web applications that invoke business logic written in RPG, COBOL, CL, or even C or C++. The service entry point is a special kind of entry breakpoint that can be set directly from the Remote System Explorer. It is triggered when the first line of a specified procedure is executed in a job that is not under debug. Service entry points allow you to gain control of your job at that point. A new debug session gets started and execution is stopped at that location.

To start a debug session:

- 1. In the Remote System Explorer perspective, expand the iSeries server icon (s400a)
- 2. Expand iSeries Objects
- 3. Expand the **Library List**.

If prompted, type your User-ID and password.

4. Expand the library WSSLABxx.

You will see all objects in this library.

Now you will start the debug session and set a service entry point.

- 5. Right-click GETDATA.*pgm.rpgle
- 6. Click **Service Entry Point** > **Set** on the pop-up menu.
- 7. Click the Web Browser tab to switch back to the Web browser.
- 8. Type 0010100 in the Enter customer number field.
- 9. Click the **Submit** button, which will invoke GETDATA*pgm.rpgle on the iSeries host.

At this point, the Service Entry point is hit and the debug session is started.

	P Variables %LOCALVARS = FOR RPG, COBOL OR CL USE TH %LOCALVARS = FOR RPG, COBOL OR CL USE TH	E "MONITORS" V
WebSphere v5.1 Test Environment [Server] WebSphere v5.1 Test Environment (WebSphere v5.1) Typ [iSeries: Debug Job]		
	-	1
E main : QZRCSRVS : QZRCSRVS	ariables Breakp Expres Registers Storage St	orag (Moni i
Welcome Web Browser		E Outline
Row 1 Column 1 Replace	Browse	
. +1+2+3+4+	5+6+7+8	An outline is not
• 1 *		available.
2 *		
3 * Sample RPGIV program to acc	cess file CUSTOML3	
4 * Used in WDT hands on LAB		
5 * Created by Claus Weiss		
6 *		
7 * F spec for file CUSTOML3 keyed	d by customer number	
i oper for the contents were		
*[l)Č	

Notice that it also opens an instance of the GETDATA source code in the debug editor. You now have the access to debug this source.

Let's look at some more debug features.

You have started a debug session with service entry points and now you are ready to begin "Exercise 7.3: Adding and deleting breakpoints."

Exercise 7.3: Adding and deleting breakpoints

Before you begin, you must complete "Exercise 7.2: Starting a Debug session using service entry points" on page 96.

Breakpoints are markers you place in your program to tell the debugger to stop your program whenever execution reaches that point. For example, if you suspect that a particular statement in your program is causing problems, you could set a line breakpoint on the line containing the statement, then run your program. Execution stops at the breakpoint before the statement is executed. You can then check the contents of variables and view the call stack, and execute the statement to see how the problem arises.

You can only set breakpoints at executable lines. All executable lines are displayed in blue. The easiest way to set a breakpoint is to right-click on the line in the Source view.

To set a breakpoint:

- 1. Click anywhere on line 8.
- 2. Right-click and click Add breakpoint on the popup menu.

A dot with a check mark in the prefix area indicates that a breakpoint has been set for that line. The prefix area is the small grey margin to the left of the source lines.

U Welcome		🎯 Web Browser	👁 GETDATA.RPGLE 🗙				
Ro	w 14	Column 1	Replace		1	Browse	
	.+	1+2	.+3	+4	+5	+6	+
+	1	*					
	2	*					
	з	* Sau	mple RPGIV	program	to acces:	s file CUS'	TOML3
	4	* Use	ed in WDT h	ands on	LAB		
	5		eated by Cl	aus Veis	3		
	6	*					
	7	* F spec	for file	CUSTOML3	keyed by	y customer	number
3	8	FCUSTOML	B IF E		K DISK		
	9	F *					
	10	D*Input	parameter f	rom Web	page		
	11	DCustnoi	2	13		like(CU	STNO)
	12	D* Data :	structure t	o specif	y output		
	13	D CSTRUC	ΕĽ	1993 i se n a manunu sa m			(custom13
	14	D feedba	sk s		20		

- 3. Click on line 31.
- 4. Right-click and click Add breakpoint on the pop-up menu.
- 5. Click the Breakpoints tab in the upper right pane of the workbench.
- 6. In the Breakpoints view right-click the breakpoint for line 8.

Here you can manage your breakpoints. You can disable, or remove breakpoints accordingly.

Line [GETDATA.RP		
	📲 Go to File 📐	
	Add Breakpoint Edit Breakpoint	
	Select All	
	Enable	
	o Disable	
	= Remove	
	🖷 Remove All	

You have set breakpoints and now you are ready to begin "Exercise 7.4: Displaying a variable."

Exercise 7.4: Displaying a variable

Before you begin, you must complete "Exercise 7.3: Adding and deleting breakpoints" on page 97.

You can monitor variables in the Monitors view. Now you will monitor the variable CUSTNOI.

To monitor a variable:

- 1. In the Source view double-click the variable CUSTNOI on line 22.
- 2. Right-click CUSTNOI.
- 3. Click Monitor Expression on the pop-up menu.

The variable appears in the Monitors view.

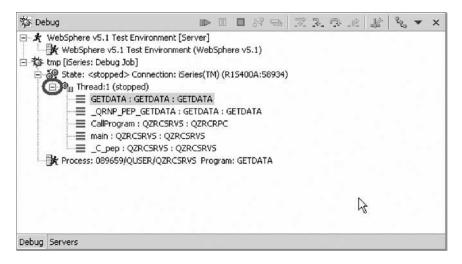
If you quickly want to see the value of a variable without adding it to the Monitor, you can place the cursor on a variable to see its value in a pop-up window. This works well for scalars, but not for arrays or data structures.

You have monitored an expression and now you are ready to begin "Exercise 7.5: Viewing the call stack."

Exercise 7.5: Viewing the call stack

Before you begin, you must complete "Exercise 7.4: Displaying a variable" on page 98.

The Debug view in the upper left pane, lists all call stack entries. It contains a tree view for each thread. The thread can be expanded to show every program, module, procedure and method that is on the stack at the current execution point. If you double-click a stack entry that has debug data, the corresponding source will show up. Otherwise the message No Debug data available appears in the debug editor. In the Debug view, expand the stack entry of **Thread:1** if it is not expanded already, by clicking the plus sign + in front of it.



You have expanded a thread to show every program, module, procedure and method on the call stack at the current execution point and now you are ready to begin "Exercise 7.6: Closing the debug session."

Exercise 7.6: Closing the debug session

Before you begin, you must complete "Exercise 7.5: Viewing the call stack."

To close a debug session:

1. Click **Resume** on the Debug toolbar.

The session will stop at the first breakpoint on line 8.

2. Click **Resume** again.

The session will stop at the breakpoint on line 31.

- 3. Click Resume again.
- 4. Click **OK** on the Program Terminated window. Next, close the Debug perspective.

- 5. Right-click the Debug icon on the left taskbar of the workbench.
- 6. Click **Close** on the pop-up menu.



- 7. Switch to the Remote System Explorer perspective to remove the Service Entry Point on GETDATA.*pgm.rpg.
- 8. Drill down to s400a > iSeries Objects > Library List > WSSLABXX > GETDATA.*pgm.rpg
- 9. Right-click on GETDATA.*pgm.rpg
- 10. Click **Service Entry Point** > **Remove** on the pop-up menu.

You have run the program to termination.

Module recap

You have completed Chapter 7, "Module 7. Debugging a Web application," on page 95. You have learned how to:

- Recognize the features of the iSeries Integrated Debugger
- · Start a debug session with service entry points
- · Set breakpoints
- Monitor an expression
- Expand a thread to show every program, module, procedure and method on the call stack at the current execution point
- Run the program to termination

Now that you have debugged your Web application, you learn how to create an informative error page using Page Designer. Continue to Chapter 8, "Module 8. Adding an error page," on page 101.

Chapter 8. Module 8. Adding an error page

This module teaches you how to create an informative error page using Page Designer for when a customer enters an incorrect customer number. This way, the customer won't just be given an error message, but will have a more informative and helpful response from the Web Application, which is easy to customize. You will also learn how to add the error page to your existing Web application using the Web Interaction wizard and to use the RSE perspective to modify the program or service program to call the error page. In this module, you will:

- Create an error page using Page Designer
- Change the Web interaction to use the new error page
- View the result of adding flow control to the Web application

Exercises

The exercises in this module must be completed in order.

- "Exercise 8.1: Creating the flow control page"
- "Exercise 8.2: Modifying your Web interaction" on page 103
- "Exercise 8.3: Testing the new error page" on page 110

Time required

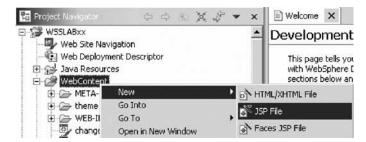
This module will take approximately **20 minutes** to complete.

Exercise 8.1: Creating the flow control page

First we need to create the error page using Page Designer.

To create the error page:

1. In the Web Perspective, expand your project and drill down to the **Web Content** folder.

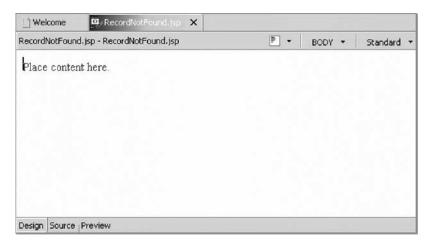


2. Right-click the **Web Content** folder and click **New** > **JSP** on the pop-up menu. The New JSP File page opens.

ISP File				
	location for the new JSP file.			A
Folder:	/WSSLABxx/WebContent			Browse
=ile Name: 🛛 🤇	RecordNotFound			
Aarkup Language:	HTML		-	
Options:	Create from page template			
	Create as JSP Fragment			
	🔽 Use XML Style Syntax			
1odel:	Struts JSP		•	
	Create a new Struts JSP.		<u>^</u>	
			*	
Configure advan	ced options			
		1000	-	s v
	< Back	Next > K Fini	ish 🕽	Cancel

- 3. In the File Name field, type RecordNotFound.
- 4. Click Finish.

The Page Designer automatically opens. Your window will look like this.



- 5. Click on the words **Place content here.** and change it to something like Sorry, there is no record for that customer number.
- 6. Press Enter.
- 7. Type Please check the number and try again.
- 8. Highlight the words try again and right-click.
- 9. Select **Insert Link** on the pop-up menu. The Insert Link window opens.

- 10. Click Browse next to the URL field.
- 11. Click **File** on the pop-up menu.

The File Selection window opens.

E- 🚰 WSSLABxx	
🖻 🧽 WebContent	
E META-INF	
1 - Coorte theme	
E 🗁 WEB-INF	
WSSLABxxDiagram.gph	
ແລະ ແລະ sages.js	
Su wdt400br.js	
- changePassword.jsp	
O error isp	
Op logon.jsp	
RecordNotFound.jsp	
	-
- 💁 timeout.jsp	-
International Allocation and Allocat	

- 12. Select inquiry.jsp.
- 13. Click OK.
- 14. Click **OK** in the Insert Link dialog.

You have completed creating the flow control page.

15. Save and close the file.

You could spend more time customizing this page, but for the purpose of this exercise you will leave the page as is.

You have created an error page using Page Designer and now you are ready to begin "Exercise 8.2: Modifying your Web interaction."

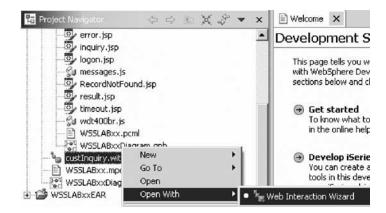
Exercise 8.2: Modifying your Web interaction

Before you begin, you must complete "Exercise 8.1: Creating the flow control page" on page 101.

Now you need to change the Web interaction to use the new error page.

To modify the Web interaction:

- 1. Expand your project WSSLABxx if not already expanded.
- 2. Right-click **custInquiry.wit** and click **Open With** > **Web Interaction Wizard** on the pop-up menu.



The Web Interaction wizard opens.

The name and location of your Web interaction are already specified.

3. Click Next to add a new output page.

		Page previ	ew:	
Generate mout JSP		I.		
Use input pages:				
înquiry.jsp	-	Add		
Ggnerate output JSP				
Use output pages:		-		
result.jsp	-	Add Remove		

4. Next to the **result.jsp** section, click **Add**. The Output JSP window opens.

-01		iound.jsp word.jsp	e.	
	nquiry.jsp ogon.jsp result.jsp timeout.jsp			

- 5. Expand Web Content.
- 6. Select RecordNotFound.jsp.
- 7. Click **OK** to add the new output page.

Web Interaction				2
pecify the Input and Output Pages for your W	eb Interaction			
ou can use previously designed input and output pag	ges for this Web interaction or you car	have the wizard gener	ate them.	°°
		Page preview:		
		[
C Generate input JSP				
Use input pages:				
inquiry, (sp	Add			
	Eenove			
	Breview			
C Ggnerate output, JSP				
Uge output pages:				
result.isp /RecordNotFound.jsp	Add			
/RecordNotFound.jsp	Remove	_		
	Prayetw			
			\sim	
			< Back (Next >)	Einish Cancel

8. Click Next in the Web Interaction wizard.

You will re-use the Feedback parameter to control which page the Web Interaction will send next after it performs its action; the result.jsp or the RecordNotFound.jsp. So you don't have to add another parameter or change the parameters, you just change the values the feedback parameter returns depending on whether a customer record was found or not found.

9. Click **Next** since you don't need to define the program interface. This opens the Input Parameters page. Click Next again since you don't need to change any input parameters. This opens the Output Parameters page.

		Link	Unink	1	
Output parameters:		Parameters	Fields	Output page fields: Utput page fields:	
မှုန်း နော် struc.CUSTNA မှုန်း နှစ် struc.CPNONE မှုန်း နှစ် struc.CPAX မှုန်း struc.CCDV နှစ် struc.CCUV မှုန်း struc.CCUV မှုန်း struc.CZIP ေ struc.CZIP ေ struc.CZIP		struc.CUSTNO struc.CONTAC struc.CONTAC struc.CFHONE struc.CFAX struc.CFAX struc.CCADR struc.CCDP	result.jsp-CUSTNO result.jsp-CUSTNA result.jsp-CPINA result.jsp-CPINA result.jsp-CPINA result.jsp-CPINA result.jsp-CENY result.jsp-CCUY result.jsp-CCUVT result.jsp-CZIP	(RecordNot	ound.jsp
Property	Value	1			
Data Type: Name in Session: Save to Session: Message Controller:	character(20) feedback			Property	Value

- 11. Select feedback under the **Output Parameters** list.
- 12. Change the **Message Controller property** to **false**.
- Select feedback under the Output Parameters list again.
 Now a Flow controller property is added to the Properties table.

ink the parameters in your orresponding output fields	program to the fields or on your Web page.	n your Web page. The	value of the output parametr	er in your program is passed to the	°°
Output parameters:		Link.	Unlink	Output page fields:	
estruc.CUSTNO estruc.CUSTNA estruc.CONTAC estruc.CONTAC estruc.CONTAC estruc.CONTAC estruc.CONTAC estruc.CONTAC estruc.CONTAC estruc.CONTAC estruc.CONTAC estruc.CZIPLO estruc.CZIPLO estruc.CZIPLO		Parameters struc.CUSTNA struc.CUSTNA struc.COSTNA struc.CONTAC struc.CFHONE struc.CADDR struc.CADDR struc.CADDR struc.CCIIY struc.CCIIY	Fields result, isp-CUSTNO result, isp-CUSTNA result, isp-REPNO result, isp-CONTAC result, isp-CATONE result, isp-CFAX result, isp-CATON result, isp-CCUTY result, isp-CCUTY result, isp-CCUTY	Peuk.jsp //RecordNotFound.jsp	
Property	Value			1	0.1
	character(20)	-		Property	Value
Data Type: Name in Session:	feedback	- 8			
Data Type:	60100				

14. Change the **Flow Controller property** to **true**. The Flow Control Specification dialog opens.

Value	Output JSP	New
0	result.jsp	- BARREN
*OTHER	/RecordNotFound.jsp	Delete
		Move up
		Move dowr

- 15. The value *OTHER is already in the **Value** field for the /error.jsp.
- 16. Click the /error.jsp field to change the Output JSP page.
- 17. Select RecordNotFound.jsp from the list.
- 18. Click New.

This creates a new flow condition. The default page is result.jsp.

- 19. The value 0 is already in the Value field with result.jsp in the Output JSP field.
- 20. Click OK.

This will set the flow control such that if the value of Feedback is 0, then the program will output the customer information to the result.jsp page and display it.

However, if the customer number doesn't exist in the database, the Feedback value will be something else other than 0, and instead of displaying a message near the input field, it will show the user the RecordNotFound.jsp page for a more helpful response.

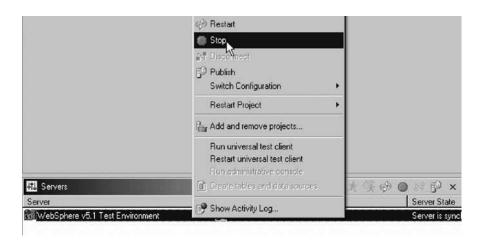
21. Click **Finish** on the Output Parameters page to apply your changes to the Web Interaction.

Heb Interaction					×
Map and Link the Outpu	t Parameters to the I	Output Fields			
Link the parameters in you corresponding output fields	r program to the fields or s on your Web page.	n your Web page. The	value of the output paramet	er in your program is passed to the	0°°
Output parameters:		Link	Unlink	Output page fields:	
Struc.CUSTNO Struc.CUSTNA Struc.REPNO Struc.CPHONE Struc.CADR Struc.CADR Struc.CADDR Struc.COUNT Struc.COUNT Struc.CIP Struc.CIP Struc.CIP Struc.CIP		Parameters struc.CUSTNO struc.CUSTNA struc.CONTAC struc.CFHONE struc.CFHONE struc.CADPR struc.CADPR struc.CCITY struc.CCITY struc.CCITY	Fields result.jsp-CUSTNO result.jsp-CUSTNA result.jsp-CPHONE result.jsp-CPHONE result.jsp-CPAX result.jsp-CADR result.jsp-CCITY result.jsp-CCITY result.jsp-CZIP	Pesult.jsp //RecordNotFound.js	p
Property	Value			Property	Value
Data Type: Name in Session:	character(20) feedback			11000102	
Save to Session	Soke				-
Flow Controller:	true				
			_ <	Back Next > Finit	sh Cancel

Note: If you encounter the following error message in your Tasks list:

	1	1	Description	Resource	In
0			The project was not built due to "Problems encountered while deleting resources."	WSSLABxx	
+	_				_
+	_				_
t					
a i			1 · · · · · ·		lix

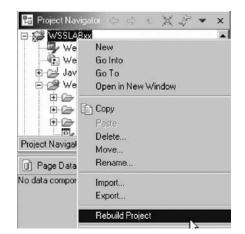
- 1. Click the Server tab to access the Server view.
- 2. Right-click the test server and select **Stop** on the pop-up menu.



- 3. Switch to your Project Navigator and right-click your project folder.
- 4. Click **Refresh** on the pop-up menu.



5. Right-click again your project folder and click Rebuild Project.



This should remove the error message in your Task view.

You have changed the Web interaction to use the new error page and now you are ready to begin "Exercise 8.3: Testing the new error page" on page 110.

Exercise 8.3: Testing the new error page

Before you begin, you must complete "Exercise 8.2: Modifying your Web interaction" on page 103.

	🐑 Web Browser	×			
http://localhost	:9080/WSSLABxx/inqu	iry, jsp	- 0	⇔ ⇔ () 🗇 🗛 🔳
	Custo	omer	Inqu	iry	×
Enter cus	tomer number: [9999			
Submit	Reset				
				r	*

Now that the flow control is in place, if you enter a wrong customer number,

you will get the error page instead of a message.



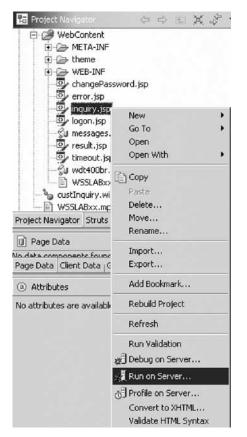
Now follow the instructions below to run the modified project on the WebSphere Test Environment to see the result of adding flow control to your Web application.

The server should still be running from the last time you ran the Web application. Every time you make a change to the Web application, you need to restart the server to pick up the changes.

To restart the server:

9	Get started To know what to do first, consul	t the Product O	6 5 3	Restart
	in the online help to decide how		۲	Stop
۲	tools in this development enviror your iSeries objects, then use th need to do is create a connection	elop iSeries applications an create and maintain RPG and COBOL ap in this development environment. Use the I iSeries objects, then use the built-in editing to do is create a connection. Once you ha 00 members.		Disconnect Publish Switch Configuration Restart Project
۲	Create Web applications You can use WebSphere Develop COBOL applications to the Web.		h	Add EGL Debugger Jar Files Remove EGL Debugger Jar Files Add and remove projects
4	from DDS display files to JavaSer IBM WebFacing Tool. Alternative starting with the Degraph Call wi	ely, you can cre		Run universal test client Restart universal test client Run administrative console
鼎 Ser	vers		Ø	Create tables and data sources
Server		Status	.0	Show Activity Log
Web	Sphere v5.1 Test Environment	🖁 Started	9	Show Rearry Login

- 1. Click the Servers tab to open the Servers view.
- 2. Select the server instance. Right-click and click **Restart** on the pop-up menu. The server instance should restart.
- 3. In the Web perspective Project Navigator locate the **Web Content** folder, and right-click **inquiry.jsp**.



4. Click **Run on Server** on the pop-up menu.

You have viewed the results of adding flow control to the Web application.

Module recap

You have completed Chapter 8, "Module 8. Adding an error page," on page 101. You have learned how to:

- Create an error page using Page Designer
- Change the Web interaction to use the new error page
- View the result of adding flow control to the Web application

Now that you have added an error page to your Web application to handle incorrect customer numbers, you can learn how to customize your input page. Continue to Chapter 9, "Module 9. Enhancing the input page using Web tools," on page 113.

Chapter 9. Module 9. Enhancing the input page using Web tools

this module teaches you how to use the Web tools in the workbench to update the input page of your customer inquiry Web application. The input page named inquiry.jsp is one of the files created by the Interaction wizard. This page is very plain. You will learn how to use Page Designer and some related Web tools in the workbench to add some color to the input page, as well as add some pictures to make it a more interesting Web page.

In this module, you will:

- · Locate the Web application input page and start Page Designer
- View the title of the input page
- Add a style to the input page
- Preview the new style for the input page
- Start WebArt Designer
- · Create a logo
- · Resize the logo
- Save the object as a WebArt object
- Save the object for a Web page
- Place the object on the Design page
- Add a company name to the input page
- Add an image to the input page
- Add a marquee to the input page
- Apply color to certain areas of text

Exercises

The exercises in this module must be completed in order

- "Exercise 9.1: Opening Page Designer" on page 114
- "Exercise 9.2: Working with page properties" on page 115
- "Exercise 9.3: Linking a cascading style sheet to the Web page" on page 116
- "Exercise 9.4: Designing and adding a logo" on page 118
- "Exercise 9.5: Adding a heading 1 tag to the page" on page 127
- "Exercise 9.6: Adding a picture to the page" on page 128
- "Exercise 9.7: Adding moving text to the page" on page 130
- "Exercise 9.8: Changing the text color" on page 134

Time required

This module will take approximately 30 minutes to complete.

Exercise 9.1: Opening Page Designer

The first step is to locate the inquiry.jsp file and to start Page Designer. In Page Designer you will link the page to a cascading style sheet that comes as a sample with the workbench.

To open the Web perspective:

1. In the workbench, open the Web perspective:

You should now have the Project Navigator in your workbench.



- 2. Expand the WSSLABxx web project.
- 3. Expand the Web Content folder.
- 4. Right-click inquiry.jsp and click **Open** on the pop-up menu.

session.se	ustom				
C	actom				
				6	
	ustome	er me	4un y		
			<u></u>		
%inquiry.label_Enter_c	ustomer num	her	5		1
	· · · · · · · · · · · · · · · · · · ·				
%inquiry.button.submit		.button.rese	nið t		
%inquiry.button.submit					

The Page Designer appears in the upper right pane of the workbench and shows the inquiry.jsp page as the Web Interaction wizard created it.

Make sure that you are on the Design page in Page Designer.

5. Click the **Design** tab.

You have located the Web application input page and started Page Designer and now you are ready to begin "Exercise 9.2: Working with page properties" on page 115.

Exercise 9.2: Working with page properties

Before you begin, you must complete "Exercise 9.1: Opening Page Designer" on page 114.

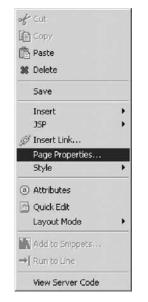
Next you move the form down.

To work with page properties:

1. Click underneath the heading Customer Inquiry.

UWelcome			
inquiry.jsp - Untitled *	₽	Р →	Standard •
Customer	Inquiry		-
BR ₄ BR ₄ BR ₄ BR ₄ BR ₄			
8R. 8R.			
%inquiry.label_Enter_customer_number			

- 2. Press Enter number of times as shown above.
- 3. To change the page properties, right-click the background of the inquiry.jsp page in Page Designer.



4. Click **Page Properties** on the pop-up menu. The Page Properties window opens.

Page title: 🕻	Customer Inquiry	\triangleright	
Base URL:			
Default target:			-
Encoding: 🔇	ISO Latin-1		

This window allows you to change some of the page properties. For example you could change the title. When this page is shown in a browser, the Window title bar of the browser will display Customer Inquiry.

You have viewed the title of the input page and now you are ready to begin "Exercise 9.3: Linking a cascading style sheet to the Web page."

Exercise 9.3: Linking a cascading style sheet to the Web page

Before you begin, you must complete "Exercise 9.2: Working with page properties" on page 115.

Now you will add a style to the input page. You can use a style that is used in your company or you can use one of the sample style sheets that are provided in the product.

To link a style sheet to an input page:

🕫 Project Navigator 🗇 🔿 🖭 💥 🗳 💌	×	Welcome	
WSSLABxx Web Site Navigation Web Deployment Descriptor Web Deployment Descriptor	<u>+</u>	inquiry.jsp - Cus	tomer Inquiry *
Project Navigator Struts Explorer			Custo
Gallery Control Con	×	BR ₄ BR ₄ BR ₄ BR ₄ BR ₄	
Page Data Client Data Gallery			
Thumbnails	×	[%mqury.la]	bel_Enter_customer_n
Style Sheet Files (*.css)	-	Design Source	Preview
Image: second		🖳 Console [W	ebSphere v5.1 Test Environ

- 1. Click the **Gallery** tab, if the Gallery view isn't already displayed.
- 2. Click the **Style Sheet** icon in the Gallery list.
- 3. Click the **Thumbnail** tab in the left bottom stacked view in the workbench. You should see thumbnail icons of all the styles available as shown above.
- 4. In the thumbnail view, scroll down to the bottom, until you see style sheet **denim001.css** in the list, or select a style sheet that you like best.
- 5. Click the thumbnail picture of denim001.css.
- 6. Hold the left mouse button down and drag the mouse cursor to the Page Designer window.

The cursor will change from this shape \bigcirc , to this shape \boxdot . When the latter cursor shape appears in the Page Designer window then, let go of the mouse left button.

After a short while the Style sheet properties will be applied.

7. Click the **Preview** tab.

You will see the colors in the page have changed to the style sheet definitions.

iiry.jsp - Custon	mer Inquiry *		<p th="" €<="" ⇔=""></p>
	Custo	omer Inquiry	
0/1i			
	bel_Enter_custom	ner_number	

8. Click the **Design** tab to get back to the Design page.

You have added a style to the input page and previewed the new style for the input page and now you are ready to begin "Exercise 9.4: Designing and adding a logo."

Exercise 9.4: Designing and adding a logo

Before you begin, you must complete "Exercise 9.3: Linking a cascading style sheet to the Web page" on page 116.

Now that you have the overall Web page look specified you will use the WebArt Designer to create a Logo that you then will add to this page.

Starting the WebArt Designer

To start the WebArt Designer:

	Sphere Development Studio Clie	
File Edit Toolbar Page	Tools Navigate Search Project	Run Window Help
Š • 🛛 🔍 🗛 1	Launch External Web Browser 🕨	
1 8 8 • B m	🔁 WebArt Designer	
🛱 🖥 Project Navigator	AnimatedGit Designer	Welcome 🔍 *inquiry isp

- 1. Make sure the Page Designer is in focus.
- 2. Click **Tools** > **WebArt Designer** in the workbench menu. The WebArt designer opens.

🗞 Untitled - WebSphere Studio WebArt Desig	gner	_ 🗆 🗙
File Edit View Object Tools Help		
℃₩日間×車 ♡		
‱ ≱/ ⊾ Щ 🔆 Ф		
WebArt Gallery Button Button Rollover Shapes Text:	Image: Second state ✓	
For Help, press the [F1] key.		

The WebArt Designer shows the Template Gallery on the left, where you find samples of logos, buttons, rollovers, images and more. The large white area in middle of the dialog is the canvas that is used to work with objects that you want to create or change. Now you will create a logo from scratch.

You could also select one from the template gallery as the base for your own logo.

Creating the logo

To create the logo:

 Click the Create Logo button above the canvas or click Object > Create Logo. The Logo Wizard opens.

Text		
		1 0000000000000000000000000000000000000
—		IELA 1 WESLADIST
Font		1
Font name:	Sans MS	
Font size: 46-	D	abo
Line 0-	% Character 0	
spacing.		
Style	Alignment	
Ttalic	Center	
Strikethrough	C Right	
Underline	Direction	
🔽 Antialias	Vertical	

- 2. In the **Text** field, type your company name and in the next line in the **Text** field, type WSSLABxx.
- 3. In the Font name list, select Comic Sans MS.
- 4. In the Font size list, select 46.
- 5. Under Alignment, select the Center radio button.

Notice in the upper right corner of the dialog, a sample of the logo as specified is displayed.

 Click Next to go to the next page of the wizard. The Select Color page opens.

Logo Wizard (Select Color - 2/4)	×
Type:	
List	MH States
autumn beer	blue01
beer	Others
< Back	Next > Cancel Help

7. Select the **gradation type** button; the middle one of the three type push buttons.

Tip: The other buttons select color types: solid and textured.

8. Select beer from the colors available, or any other color you like best, just scroll through the list to find a gradation you like.

Tip: You can change the colors by clicking the Others button on this dialog and create you own gradation.

 Click Next to go to the next page of the wizard. The Select Outline page opens.

	a	IEM
Normal	Seal Hollow	wflabeor
Options		
Edge width: Edge color:		
Transparency:	J 2	
Bulge for edge:	A A	

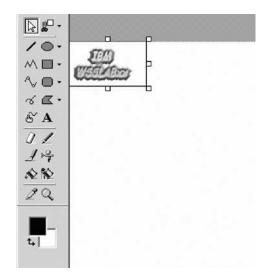
- 10. Select the **Seal outline** from the list, or any outline you like best.
- 11. Click Next.

The Select Text Effect page opens.

a	a) &	IEM wildber
] Dptions Embossment		
strength: Embossment direction:		
	к † л	

- 12. Select the **Emboss text** effect, or one that you like best.
- 13. Click Finish.

You return to the WebArt designer window.



Resizing the logo

To resize the logo object on the canvas:

- 1. Click the logo object to select it.
- 2. Move the cursor to the rectangle at the right bottom corner of the object; watch the cursor changing shape.
- 3. Drag the rectangle up and to the left so the object becomes smaller.

Saving the object as a WebArt object

Now you need to save this object. First save it as a WebArt object. This allows you later on to work with the object again in WebArt Designer, but you can't use that format for your Web page.

To save this object:

1. Click File > Save Canvas As on the WebArt Designer menu.

The Save Canvas As window opens. Make sure you see the WebContent folder in the Save Canvas As window.

Save Canvas	As		? ×
Save in: 🔀	WebContent	E e	-
META-INF theme WEB-INF			
File name:	mylogo		Save
Save as type:	WebArt files (*.mif)	<u> </u>	Cancel

- 2. In the **File name** field, type mylogo.
- 3. Click Save.

Now you need to save the object in a form that can be displayed on a Web page.

Saving the object for a Web page

To save the logo for a Web page:

- 1. Click the logo object on the canvas.
- Click File > Save wizard for Web from the WebArt Designer menu. The Save Wizard opens.



- 3. Click the Save the selected object radio button.
- 4. Click Next.

The Select File Format page opens.

This format is in general use on the Internet because of its small file ize and inconspicuous image degradation. This format is suitable or full-color images, such as photos taken with a digital camera. You cannot make a part of a JPEG image transparent.
This format maintains the quality of an image, but the file size becomes larger than that of a JPEG. This format is suitable for ull-color images whose quality is important, such as computer raphics. You can make a part of a PNG image transparent; however, only a ew browsers can handle it correctly.

- 5. Click the **GIF** radio button.
- 6. Click Next.

The GIF Format page opens.

Save Wizard (GIF Format) - 3/3	x
[Original image]	[Image to be saved]
	ECC) Wilders
File size before: File size after: Palette Use the palette for the current file Use the optimized palette Colors: 256 colors Dither Yes	19380 bytes 3836 bytes Properties Transparent Interlace Background color Color of the Canvas
< Back Finish	Cancel Help

7. Click Finish.

The Save As window opens.

				?)
WebContent	•	\$ E	₫	
ou.aif				
mylogd			Save	9

Make sure the directory is pointing to the Web Content directory in the workspace where your Web project is located.

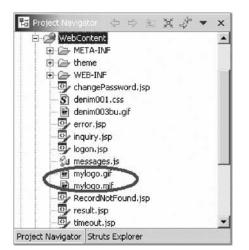
- 8. In the **File name** field, type mylogo.
- 9. Click Save.
 - Close the WebArt designer.
- 10. Select **File** > **Exit** from the menu.

You return to the workbench in the Web perspective and see the Project Navigator.

Placing the logo on the Design page

To place the logo on the Design page:

- 1. Make sure you switch from the Gallery view to the Project Navigator.
- 2. Expand the WSSLABxx Web project.
- 3. Expand the Web Content folder.



The mylogo.gif file should appear in the list as shown below. If it doesn't appear, the list might need to be refreshed.

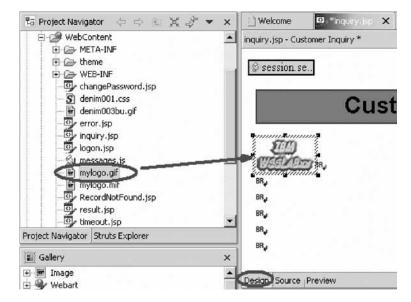
- 4. Right-click the Web Project WSSLABxx folder icon.
- 5. Select Refresh.

Hopefully you can see the file now in the Web Content folder, if not, go to Windows Explorer and search for the mylogo.gif on your hard drive. Move it to the Web Content folder in the product workspace. If you didn't use the default location as shown above, do a search for the workspace and the WSSLABxx directory in it. Move the mylogo.gif file into the Web Content sub directory under the WSSLABxx directory. Now you can take the logo and put it on your Web page that is still open in Page Designer. If Page Designer has been closed just open the inquiry.jsp file.

Note: Make sure your are on the Design page, not the Preview page.

- 6. In the Project Navigator view select the mylogo.gif file.
- 7. Hold the left mouse button down.
- 8. Drag the file to the upper left side in the Page Designer window.
- 9. Release the mouse button.

The logo is placed on the Design page.



You have started WebArt Designer, created a logo, resized a logo, saved the object as a WebArt object, saved the logo for a Web page, placed the object on the Design page and now you are ready to begin "Exercise 9.5: Adding a heading 1 tag to the page."

Exercise 9.5: Adding a heading 1 tag to the page

Before you begin, you must complete "Exercise 9.4: Designing and adding a logo" on page 118.

Now you want to insert a heading below the logo.

To add a heading 1 tag:

1. Position the cursor just below the logo at the first BR tag.

Edit Toolba	Insert JSP Format Table Fram	ne Page	Tools Navigate :	Search F
š • 🔛 🖳 (🗏 🖉 Link			4 <u>1</u>
×₽ ● E	🖥 🕖 Anchor			
Ta Project N	Image File		tome 💿	inquiry. (s)
⊡-@ 9 ₩ ⊡-@	Ducinground masic		Isp - Customer Ir	nquiry *
E G				
€-@	Text Cell		5510n.se	
- <u>S</u>	Paragraph		Normal	Ctrl+0
			Heading 1	Ctrl+1
	Table	Ctrl+T	Heading 2	Ctrl+2
	- Horizontal Rule		Heading 3	Ctrl+3
-9	J Form and Jonut Fields		Heading 4	Ctrl+4
	Form and input Fields		Heading 5	Ctrl+5
	Script		Heading 6	Ctrl+6

- 2. Click **Insert** from the workbench menu.
- Click Paragraph > Heading 1 from the pop-up menu. A frame appears that allows you to enter text.
- 4. Enter your company name.

	🔍 *inquiry.jsp 🗙		
inquiry.jsp - Customer Inquiry *		🖭 т 🛛 н1 т	Standard 🔻
S session.s	e		<u>^</u>
	Custome	r Inquiry	
E			
WSSLAB	∰ BR _↓		
2	l Laboratory	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
	Laboratory		
BR _e			-
Design Source	Preview		

Now you want to use one of the sample pictures that comes with the product and place this picture on the page.

You have added a company name to the input page and now you are ready to begin "Exercise 9.6: Adding a picture to the page."

Exercise 9.6: Adding a picture to the page

Before you begin, you must complete "Exercise 9.5: Adding a heading 1 tag to the page" on page 127.

To add a picture to the page:

1. In the Project Navigator, click the Gallery tab.

🖪 Project Navigator 👙 😄 📹 💥 👙 👻 😒	K I Welcome . Inquiry type X
	inquiry.jsp - Customer Inquiry * session.se
Project Navigator Struts Explorer	Custo
I Gallery	
Image Banner Bullet Button Inustration Line Photo	
Page Data Client Data Gallery	IBM Laborat
🗟 Thumbnails 💦 刘	<
Image Files (*.gif,*.jpg,*.png,*.bmb,*.wmf,*.emf,*. Uds2cut.gff u014cut.gif u015cut.gif	Design Source Preview Console [WebSphere v5.1 Test Enviro

- 2. Expand the Image folder.
- 3. Select the **Illustration** folder.
- 4. On the Design page beside the Gallery view select one of the sample illustrations, for example, use file u013cut.gif.
- 5. Drag the picture onto the Design page beside the logo.

The Design page now contains a picture.

uiry.jsp - Cus	tomer Inquiry *		\$ \$ \$
	Custo	morloquiny	
	Cusic	omer Inquiry	
	Phile .		
(III) Maaya	1 Sen		
IBN	l Laborat	ory	
		/	
%inquiry.	label_Enter_custom	er_number	
%inc	quiry.button.submit	%inquiry.button.reset	
201110	fully.button.submit	78inguny.button.reset	

You are almost done. Next you add moving text to the page.

You have added an image to the input page and now you are ready to begin "Exercise 9.7: Adding moving text to the page."

Exercise 9.7: Adding moving text to the page

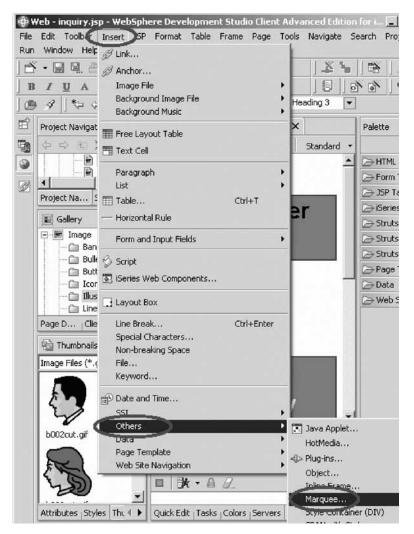
Before you begin, you must complete "Exercise 9.6: Adding a picture to the page" on page 128.

To add moving text:

- 1. Back to the Design page, position the cursor underneath your company name.
- 2. Click **Insert** from the workbench menu.
- 3. Click **Paragraph** > **Heading 3** on the pop-up menu.

iquiry.jsp - Customer Inquiry *	F	• НЗ •	Standard
🖇 session.se			
Custom	ner Inqui	ry	
500			
1646			
CUEST BIT			
IBM Laboratory	ý		
IBM Laboratory	y		
IBM Laboratory			
IBM Laboratory			

- 4. Leave the cursor positioned inside the heading 3 frame.
- 5. Click **Insert** from the workbench menu.



6. Click **Others** > **Marquee** on the pop-up menu The Insert Marquee window opens.

Movemen Slide	Repetition Finite	
Direction: (Auto) Display speed Delay time: Scroll amount:	Count: 1	- -
Use true speed		

- 7. Enter into the **Text** field Please enter customer number and press Submit button to start application.
- 8. Select **Slide** from the **Movement** list.
- 9. Select **Finite** from the **Repetition** list.
- 10. Select **1** from the **Count** list.

The two last selections just avoid the text sliding in forever and not standing still. If you want more movement on the page, you can change these settings.

- 11. Select Lime in the Background color list.
- 12. Click OK.

The Design page should look like:

To save some space, you can remove some of the line break tags.

13. Position the cursor on the **BR** tag.

U) Welcome	
inquiry.jsp - Customer Inquiry *	🖹 👻 BODY 👻 Standard 🕶
S session se	
Custome	r Inquiry
EBIA) WASILIBIT	
IBM Laboratory	
Please enter customer num button to start application	ber and press Submit
O %inquiry.label_Enter_customer_numb	er 🗖

14. Press the **Delete** key until the entry field and push buttons appear on your page:

Next you view the page as it would appear in a browser.

15. Click the **Preview** tab at the bottom of the Design page.You will notice that your heading 3 text is sliding in.

Welcome *inquiry sp ×	යට ම
idairy.jsp - Cascomer Indairy	1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
Custo	omer Inquiry
USEL Bro	
IBM Laborat	ory
Please en	ter customer number and press Sub
%inquiry label_Enter_custon	
%inquiry.button.submit	%inquiry.button.reset
First message Second message Third message	
Design Source Preview	

You have added a marquee to the input page and now you are ready to begin "Exercise 9.8: Changing the text color."

Exercise 9.8: Changing the text color

Before you begin, you must complete "Exercise 9.7: Adding moving text to the page" on page 130.

Sometimes you want to change the text color. There is an easy way to apply another color to certain areas of text. To do that you return to the Design page.

To change the text color:

1. Click the **Design** tab.

	oratory
Please enter o	customer number and p
Colors	
#66ff00 - //	Text Color Set as Table Background Color Set as Background of Layout Box Set as Text Color Set as Background Color
Quick Edit Tasks Colors Ser	rvers Console

2. Select the company text (IBM) you want to change color on, by swiping the text area with the mouse cursor.

Below the Page Designer window there are several tabs.

- 3. Click the **Colors** tab.
- 4. Select a color from the Color palette.
- 5. Click **Text color** from the Color menu.
- Click Set as Text Color on the pop-up menu. You are done; now the selected text will be displayed with the color you selected for it.
- 7. Click the Preview tab to view your completed page.
- 8. Save the file inquiry.jsp.

	■ *inquiry.jsp ×	
inquiry.jsp - Cus	tomer Inquiry *	(+ c) (*)
-		×
	Customer In	quiry
EBA) WISSIAD		
IB	Laboratory	
-		Please enter customer
%inquiry	label_Enter_customer_number	⊒ .
Design Source	Preview	

- 9. Exit Page Designer.
- 10. In the Project Navigator expand the WSSLABxx folder.
- 11. Expand the Web content folder.
- 12. Right-click inquiry.jsp.
- 13. Click **Run on server** on the pop-up menu.

Your new designed input page opens and you can run your Web application.

http://localhost:9080/WSSLABxx/inquiry.jsp	🖸 🖉 🗇 🗇 🖗 🖨 📘 🔳
Custome	er Inquiry
USE Com	
IBM Laboratory	/
Pleas	e enter customer number and
Enter customer number:	
Done	

You have applied color to certain areas of text on the input page.

Module recap

You have completed Chapter 9, "Module 9. Enhancing the input page using Web tools," on page 113. You have learned how to:

- Locate the Web application input page and start Page Designer
- View the title of the input page

- Add a style to the input page
- Preview the new style for the input page
- Start WebArt Designer
- Create a logo
- Resize the logo
- Save the object as a WebArt object
- Save the object for a Web page
- Place the object on the Design page
- Add a company name to the input page
- Add an image to the input page
- Add a marquee to the input page
- Apply color to certain areas of text

Finish your tutorial by reviewing the materials in Chapter 10, "Summary," on page 137.

Chapter 10. Summary

In this tutorial you learned how to create a simple e-business customer inquiry application that used a Web-based front end to communicate with the business logic written in ILE RPG residing on an iSeries server. While creating a browser user interface, you used the iSeries Web Interaction wizard to generate input and output JSP files as well as Page Designer to enhance the input and output Web pages. You also added iSeries Web components to your pages, for example, Web equivalents of iSeries command keys, input fields that accept only particular types of data, or output fields such as subfile names. You then ran the application in the WebSphere Test Environment that is part of the product.

Completed learning objectives

If you have completed all of the modules, you should now be able to:

- Use the Remote System Explorer perspective to edit and compile an iSeries program or service program
- Use the Web perspective tools and views to create a Web project for your Web application
- Create a Web interaction to use an input page and an output page and to create a servlet to invoke an RPG program or service program to get data from the iSeries
- Run the Web application in the WebSphere Test Environment
- Use service entry breakpoints to debug your Web application
- Create an informative error page for customers when an incorrect customer number is entered
- Add color, pictures, change text to make the input page more attractive

More information

For more information on the product and the iSeries Web Tools, see http://ibm.com/software/adwtools/iseries.

Appendix. Notices

Note to U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing 2-31 Roppongi 3-chome, Minato-ku Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Lab Director IBM Canada Ltd. Laboratory 8200 Warden Avenue Markham, Ontario Canada L6G 1C7

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 1992, 2004. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM iSeries OS/400 RPG/400 VisualAge WebSphere

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

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT[®], Win32, Win32s and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX[®] is a registered trademark of The Open Group.

Other company, product, and service names may be trademarks or service marks of others.

IBW ®

Printed in USA