

IBM WebSphere Development Studio Client for iSeries

Build a Web user interface with subfiles using iSeries Web Tools

Version 6.0



IBM WebSphere Development Studio Client for iSeries

Build a Web user interface with subfiles using iSeries Web Tools

Version 6.0

Contents

Introduction v
Chapter 1. Module 1. Reviewing the existing RPG e-business application 1 Exercise 1.1: Reviewing the Web customer inquiry application
Chapter 2. Module 2. Creating the
iSeries service program for your Web application
Exercise 2.1: Opening the Remote System Explorer
perspective
Exercise 2.2: Editing an iSeries source member 10
Exercise 2.3: Compiling source changes
Chapter 3. Module 3. Creating the
customer list input page
Exercise 3.1: Starting the product workbench 20
Exercise 3.2: Opening the Web perspective 21
Exercise 3.3: Creating the customer list page 22
Exercise 3.4: Linking to the customer list page 37
Exercise 3.5: Creating another Web interaction 40

Chapter 4. Module 4. Creating the

Exercise 4.1: Generating the output page from the	
Web Interaction wizard	53
Exercise 4.2: Linking to the update record page .	65
Exercise 4.3: Creating the update error page	70

Chapter 5. Module 5. Creating the

update complete output page	75
Exercise 5.1: Generating another output page using	
the Web Interaction wizard	. 75
Exercise 5.2: Adding the flow control error page .	85
Exercise 5.3: Linking the update complete page to	
the inquiry page	. 88
Exercise 5.4: Visualizing the flow structure of your	
Web application	. 90
Chapter 6 Medule 6 Dupping the Web	
Chapter 6. Module 6. Running the Web application	95
application Exercise 6.1: Restarting the server	95 96
application	95 96 01
application	95 96 01 03 104
application	95 96 01 03 104

Introduction

In the tutorial called Build a Web user interface for an RPG application using iSeriesTM Web Tools, you learned how to create an e-business RPG customer inquiry application. When you ran your Web RPG application a 'new' Web customer inquiry page opened in the workbench Web browser. You then entered a customer number and clicked the submit button. The customer details from a DB2/400 database appeared in the workbench Web browser.

This tutorial teaches you how to display a list of customers, display the details for each selected customer and then update selected customer information. You will learn how to use Page Designer to create a new customer list input page from a subfile. You will then learn how to link this new page to the customer inquiry page that you built in the previous tutorial. Then you will learn how to create new output pages (update record and update complete) using the Web Interaction wizard. You will also learn how to link these pages using the Web Interaction wizard. Finally you will learn how to test your new and updated input and output pages in your existing e-business RPG customer inquiry application.

Prerequisites

In order to complete this tutorial end to end, you should already have completed the tutorial called Build a Web user interface for an RPG application using iSeries Web Tools and 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:

- HTML
- 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 **1 hour and 40 minutes**.

Learning objectives

This tutorial is divided into six 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. Reviewing the existing RPG e-business application," on page 1 teaches you about the completed customer inquiry application that you

built in the tutorial called Build a Web user interface for an RPG application using iSeries Web Tools. You will then see how this application changes with the updates you will make working through this tutorial. In this module you will:

- View the completed pages of the customer inquiry Web application
- Look at the enhanced customer inquiry application

Chapter 2, "Module 2. Creating the iSeries service program for your Web application," on page 9 teaches you how to select the RPG service program to handle the Web Interaction shown in Chapter 1, "Module 1. Reviewing the existing RPG e-business application," on page 1. In this module, you will:

- Check the Remote System Explorer perspective is open
- Create a library filter
- Open an ILE RPG source member for edit
- Add code to an ILE RPG source member
- Create an RPG module
- Create a service program

Chapter 3, "Module 3. Creating the customer list input page," on page 19 teaches you how to use Page Designer to create a Web input page of a subfile. This new input page will pass the customer number back to the inquiry page. You will also learn how to modify the inquiry page by adding a link (Find) that will open the customer list page. In this module, you will:

- Start the product
- Set the workspace location
- Check the Web perspective is open
- Create a JSP file
- · Add an iSeries table component
- Select attributes for the table
- Insert a form on the page
- · Add the iSeries button component
- Add iSeries button attributes
- Add a heading to the page
- Link the inquiry page to the customer list page
- · Create another Web interaction to use the new input page and output pages
- Begin the process to add a program to the Web interaction
- Create an iSeries connection
- Select the service program
- Add an output parameter
- Map an output parameter to an output field

Chapter 4, "Module 4. Creating the update record output page," on page 53 teaches you how to create an output page using the Web Interaction wizard. This is the page that shows the customer information you can update. In this module, you will:

- Create the update record page
- Add a program
- Add parameters
- Design the output page

- Link the customer details page to the update record page
- Create an error page

Chapter 5, "Module 5. Creating the update complete output page," on page 75 teaches you how to create another output page using the Web Interaction wizard. This is the page that shows the customer record was updated on the iSeries. In this module, you will:

- Generate another output page
- Add a program
- Add a parameter
- Design the output page
- Add a flow control error page
- Link the update complete page to the inquiry page
- View the flow structure of your Struts-based Web application

Chapter 6, "Module 6. Running the Web application," on page 95 teaches you how to run the updated Web application in the WebSphere[®] Test Environment. In this module, you will:

- Start the application server
- Test the customer list page
- Test the update record and update complete pages
- View the flow structure of your Struts-based Web application

When you are ready, begin Chapter 1, "Module 1. Reviewing the existing RPG e-business application," on page 1.

Chapter 1. Module 1. Reviewing the existing RPG e-business application

This module teaches you how the existing RPG e-business application that you built in the tutorial called Build a Web user interface for an RPG application using iSeries Web Tools works and looks. You will then see what this application will look like when you have completed this tutorial.

In this module, you will:

- View the completed pages of the customer inquiry Web application
- · Look at the enhanced customer inquiry application

Requirement: Before beginning this module, you should have the prerequisite knowledge outlined in "Introduction" on page v.

Exercises

The exercises in this module must be completed in order.

- "Exercise 1.1: Reviewing the Web customer inquiry application"
- "Exercise 1.2: Introducing the enhanced Web application" on page 3

Time required

This module will take approximately 10 minutes to complete.

Exercise 1.1: Reviewing the Web customer inquiry application

In the tutorial called Build a Web user interface for an RPG application using iSeries Web Tools you 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 invoked 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 added this RPG program or service program to the Web application. Finally you tested the application in the local test environment on your workstation. You also created an informative error page for when a customer entered an incorrect customer number.

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

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

Customer Inquiry	
Enter customer number: 0010100	

When you entered a customer number and clicked the **Submit** push button, the Customer Details output page (result.jsp) appeared.

Cus	tomer Details	*
Customer Number: Customer Name:	0010100 Meridien Electronics Limited	
Representative Numb		
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 clicked the Back button in the browser to return to the Customer Inquiry input page. Here you entered an incorrect customer number and clicked **Submit**.

Customer Inquiry
Enter customer number: 9999
Submit Reset

An error message appeared.

Customer Inquiry			
er 9999 not valid			

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

Customer Inquiry	*
Enter customer number: 9999	
Submit Reset	
	*

an error page opened instead of a message.

Sorry, there is no record for that customer number.	<u></u>
Please check the number and <u>try again</u> .	
	T

You have viewed the completed pages of the customer inquiry Web application and now you are ready to begin "Exercise 1.2: Introducing the enhanced Web application."

Exercise 1.2: Introducing the enhanced Web application

Before you begin, you must complete "Exercise 1.1: Reviewing the Web customer inquiry application" on page 1.

By the end of this tutorial you will have enhanced the Web customer inquiry application. You will invoke the Web Interaction wizard to first create a new input page to view a list of customers (custList.jsp). Next you will add an iSeries RPG service program to handle this Web interaction that you just created. Then you will invoke the Web Interaction wizard to use the existing output page (result.jsp) to create new output pages to update customer records and confirm updates were completed on the iSeries (custDispUpdateResults.jsp and custDoUpdateResults.jsp). You will then run the application in the local test environment on your workstation. You will also create an informative error page if there is an error during the update of a customer record.

Here is the completed e-business customer inquiry application.

This is the Customer Inquiry input page.

Customer Inquiry
EEM) WESTABUT
IBM Laboratory
Please enter customer number and press Submit button to
Enter customer number
Submit Reset

You click the **Find** link. The Customer List page opens:

+++	1218 9	Select Action 🗾 🖸 Go	
Select ^	Customer Number A	Customer Name 🔿	Cust
۲	0010100	Meridien Electronics Limited	206-
0	0010200	Royal Hardware Supplies	905-
0	0010300	Webster Appliances	619-
0	0010400	ProLine Building Supplies	905-
0	0010700	Universal Communications Ltd.	415-
0	0010800	Baker Electronics	818-
0	0010900	Village Telephone	707-
0	0011100	BelAir Communications incorp.	914-
0	0011300	Burnham Trading Inc	613-
0	0011400	Calderone Imports	407-
Page	1 of 8 D 1	Go Total: 71 Filtered: 71 Display	ed: 10 S

You select the first entry on the list and then click the **Get Customer Number** button. You return to the Customer Inquiry page with the customer number entered into the customer number field.

Cu	stomer Inquiry
EBM) WSSIABOOR	
IBM Labor	ratory
	Please enter customer number and pres
Enter customer number	0010100
Submit Reset	
166161662	T

Next you click the **Submit** button. The Customer details page opens showing the details of the selected customer.

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:	USA	
Zip:	98007	
Update Customer Back to Customer Ingu		

When the information is displayed, you click the **Update Customer Record** button.

Update	Customer Record
Customer Number:	0010100
Customer Name:	Meridien Electronics Limited
Representative Number:	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:	98006
Submit Reset	

You change some of the fields and click the **Submit** button. You see the confirmation page with the new values.

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>U.S.A</u> .
Zip:	98006
<u>Back to Customer Inq</u> ı	uiry

You then click the **Back to Customer Inquiry** link and you return to the Customer Inquiry page.

You have looked at the enhanced customer inquiry application that you will build in this tutorial.

Module recap

You have completed Chapter 1, "Module 1. Reviewing the existing RPG e-business application," on page 1. You have learned how to:

- View the completed pages of the customer inquiry Web application
- Look at the enhanced customer inquiry application

You now know what the customer inquiry application will look like when you complete this tutorial and you are ready to begin Chapter 2, "Module 2. Creating the iSeries service program for your Web application," on page 9.

Chapter 2. Module 2. Creating the iSeries service program for your Web application

This module teaches you how to use the Remote System Explorer perspective to edit and compile an iSeries service program.

Note: To access iSeries data and resources you will need to have a job description setup with WSSLABxx library in the list of libraries and associated with your user profile on your iSeries system.

In this module, you will:

- · Check the Remote System Explorer perspective is open
- Create a library filter
- Open an ILE RPG source member for edit
- · Add code to an ILE RPG source member
- Create an RPG module
- · Create a service program

Exercises

The exercises in this module must be completed in order.

- "Exercise 2.1: Opening the Remote System Explorer perspective"
- "Exercise 2.2: Editing an iSeries source member" on page 10
- "Exercise 2.3: Compiling source changes" on page 12

Time required

This module will take approximately 10 minutes to complete.

Exercise 2.1: Opening the Remote System Explorer perspective

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 Remote System Explorer icon.
- 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 2.2: Editing an iSeries source member."

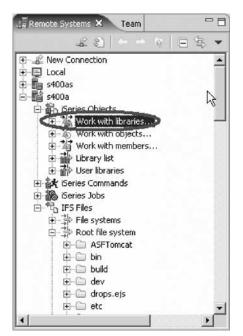
Exercise 2.2: Editing an iSeries source member

Before you begin, you must complete "Exercise 2.1: Opening the Remote System Explorer perspective" on page 9.

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

To edit a member:

- 1. Expand Library list. If you do not see WSSLABxx then complete these steps:
 - a. Double-click Work with libraries.



The Library Filter page opens.

@ New	43		×
Library Filter Create a new iSeries library filter			
Library: wsslabxx		<u> </u>	Browse
< Back	Next >	Finish	Cancel

b. In the **Library** field, type WSSLABxx.

c. Click Next.

@ New				×
Library Filter				
Name the new filter				
Filters are saved for appear in the Remot				lter. This name will
Filter name:	labxx			
Only create filter	in this connec	tion		
Select a profile to ov sharable by the tear				
Owner profile:	eckh7tp			•
	< Back	Next >	Finish	Cancel

- d. In the **Filter name** field, type WSSLABxx.
- e. Click **Finish** to create the library filter WSSLABxx. Next you edit the source file.
- 2. Expand WSSLABxx.*lib.prod.



- 3. Expand QRPGLESRC.*file.pf-src.
- 4. Select WDSCSRV.rpgle.
- 5. Right-click WDSCSRV.rpgle.
- 6. Click **Open with > Remote Systems LPEX Editor** on the pop-up menu.

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

The Editor window opens. You are ready to write your program.

The logic you add will do the following. It will retrieve a changed record from a table component. The changed record is the selected record in the table. It will pass the customer number from this selected table record back to the caller.

7. Now, add these source changes at the bottom of the source file:

Line 217	Column 33	Insert		
	PName+++++++	++++B	Кеуъ	vords++++++++++++++++++++++++++++++++++++
000295	*			
000296	* readc			
000297	*			
000298	* Return cu	stomer number		
000299	*			
000300	*			
000301	PdoUpdate	E		
000302	*			
000303	preadc	В	expo	ort
000304	d readc	pi		
000305	d info		like	(CUSTNO)
000306	*			
000307	C	Eval	rc=readcSF(C	CustList:
000308	C			%Addr (CustRec) :
000309	С			%Size(CustRec):
000310	С			1)
000311	C	IÍ	rc > 0	
000312	С	Eval	info = SflID)
000313	C	Endif		
000314	C			
000315	*			
000316	Preadc	E		
000320	**** End of	Source		· · · · · · · · · · · · · · · · · · ·
•				•

8. Save the file (Ctrl + S).

You have created a library filter, opened an ILE RPG source member for edit, and added code to this member and now you are ready to begin "Exercise 2.3: Compiling source changes."

Exercise 2.3: Compiling source changes

Before you begin, you must complete "Exercise 2.2: Editing an iSeries source member" on page 10.

You need to compile your source file in order for the changes to take effect.

There are two steps to compile: create a module and then create the service program.

Note: The library WSSLABxx must be in your library list to compile your source changes.

Creating the RPG module

To create the RPG module:

1. In the Remote Systems view, select WDSCSRV.rpgle.

.:≣Remote Systems ≻	Team	- 0	inquiry.jsp	WDSCS	RV.R 23 🗖	02 -
- <u>-</u> 8		ⓐ 🖂 🔄 👻	Line 21	B Col	umn 25 In	🗢 🗸
	PRODDTL. *fill PRODINV.*fill PRODUST.*fil PRODUCTJF.* QCBLLESRC.* QCBLESRC.* QCBCLESRC.* QCPGLESRC.* GETDASO GETDASO GETDASO GETDASO GETDASC G	e.pf-dta e.pf-dta le.pf-dta le.pf-dta le.pf-dta le.pf-dta trile.pf-src le.pf-src trile.pf-src trile.pf-src trile.pf-src t.rpgle S.rpgle L.rpgle Go To Open With Browse With	Ph 020105 * 020106 pr 020107 d 020108 d 020109 * 020110 C 020111 C 020112 C 020112 C 020113 C 020114 C 020115 C 020117 C 020117 C 020117 C 020117 C	ame++++++ readc readc info readc cessful.	E V All chang	Press Refr
Properties 🕄		Copy ++ Move X Delete	;400;			<i>L</i> •
Property	Value					
Attribute Name Number of children Source	SRC WDSCSRV 0 WSSLABXX/Q	 Find String Verify Verify (Prompt). 		meter).	AR parameter), Fi	eld values 🔺
Status	ОК	Compile Compile (Prompt User Actions)	CRTBNDRPG	45.00	
		USCI MUUUIS	~ a	CRTRPGMOD	ha N	

2. Right-click and click **Compile (Prompt)** > **CRTRPGMOD** in the pop-up menu. The Create RPG Module (CRTRPGMOD) dialog opens.

Create RPG Module (CRTRPGMOD)		N	x
Module:	> WDSCSRV	4	Name
Library:	> WSSLABXX]	Name
Source file:	> QRPGLESRC 💌]	Name, QRPGLESRC
Library:	> WSSLABXX		Name
Source member:	> WDSCSRV _]	Name
Source stream file:		-	
		<u></u>	1
Generation severity level:	10		0-20
Text 'description':	*SRCMBRTXT	2	Character value
	Advanced Paramete	rs	
Compiler options:	>	1	
	*EVENTF	Remove	
		Move up	
		Move down	
Debugging views:	> *SOURCE V	There as in the	
Replace module:	> *YES -		
Replace mousie:			
Advanced(1)	All Parameters(2)	🗖 Keywords(3)
CRTRPGMOD MODULE(WSSLABXX/WDSCSR	W) SRCFILE(WSSLABXX/QRPGLE	SRC) SRCMBR(WDSCSRV) OP	TION(*EVENTF)
DBGVIEW(*SOURCE) REPLACE(*YES)			
			<u>~</u>
		OK Restore de	faults Cancel

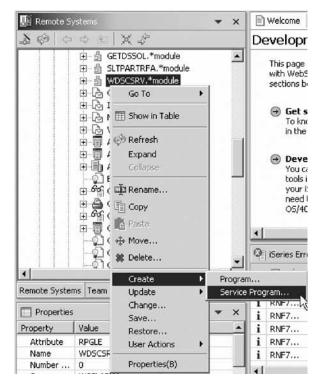
3. Click OK.

ries Tab	le View	📲 iSeries Commands Log 🗙	Tasks	iSeries Error List	Remote System Details	I •	
00as	s400a						
OPTIOP Compila Cause the GEP program Compila Cause specifie	N(*EVEN ation stop NLVL opti n. Compi ation faile d for the	DULE(WSSLABXX/WDSCSRV (F) DBGVIEW(*SOURCE) REP ped. Severity 40 errors foun The RPG compiler found at It on on the CRTRPGMOD or CF e again. d. Module WDSCSRV not cre. Compilation failure normally GENLVL parameter on the CI very: Correct the error	LACE(*YES d in progra east one er TBNDRPG ated in libra occurs whe RTRPGMOD) m. ror in the program command. Recove ary WSSLABXX. n the severity of is 0 command. See th	of severity greater than the ry: Correct the errors ssued messages exceeds th e compiler listing or the job l	at specified in s in the e value log for error	
5.0.x')		'SSLABXX/QRPGLESRC) MBR(EPR2 added to file QRPGLESF			E) TEXT('Prototypes for Sub	file componeni	t
Jurrent	library o	RLIB(WSSLABXX) hanged to WSSLABXX. The current library in the libr	ary list wa	s changed to WSSI	.ABXX.		
OPTIO	V(*EVEN	RTRPGMOD MODULE(WSSLAM (F) DBGVIEW(*SOURCE) REP (KHAM/MELANIEJOB submit)	LACE(*YES)) JOBD(*USRPRF)	BR(WDSCSRV)	
OPTION Module Cause	V(*EVEN WDSCSF	DDULE(WSSLABXX/WDSCSRV (F) DBGVIEW(*SOURCE) REP (V placed in library WSSLABX) Module WDSCSRV was succe . The module creation date ai	LACE(*YES K. 00 highe essfully cre) st severity. Create ated in library WSS	ed on 05/01/25 at 16:35:20. SLABXX. The highest messag		ľ
omman	d Norm	el 🗾			•	Prompt	Run

4. Check the iSeries Command Log and see if the module was created. You have now completed the first step of the compilation process. Now you complete the second step in compiling the source file.

Creating the service program

To complete the second step:



- 1. Right-click WDSCSRV.module.
- 2. Click **Create** > **Service Program** in the pop-up menu. The Create Service Program (CRTSRVPGM) dialog opens.

Create Service Program (CRT	SR¥PGM)						×
Service program:	>	WDSCSRV				Name	
Library:	>	WSSLABXX	•			Name	
Module:	>		•			Name, generic*	
Library:			•	Add	Name		
		WSSLABXX/WDS	CSRV	Remove			
				Move up	1		
				Move down			
Export:		*ALL	•		-		
Export source file:		QSRVSRC	•			Name, QSRVSRC	
Library:		*LIBL	*			Name	
Export source member:		*SRVPGM	•			Native	
Text 'description':		*BLANK				Character value	
		Advanced Pa	rameters				
Bind service program:	- 1	qdtssfl	•			Name, generic*	
Library:	1	qsys	•	Add	Name		
Binding directory:		*NONE	-		-	Name	
Library:		*LIBL	v	Add(F)	Name		
Activation group:		*CALLER	-		-	Name	-
Creation options:		ĺ	•	Add(H)	1		
Listing detail:		*NONE	•				
Allow update:		*YES					
Allow *SRVPGM library update:		*NO					
User profile:		*USER					
osor promot							-
Ndvanced(L)	E 4	I Parameters		ſ	Keywords		
CRTSRVPGM SRVPGM(WSSLABXX/WD	SCSRV) MC	DULE(WSSLABXX	/WDSCSR\) EXPORT(*	ALL) BNDSRVI	PGM(QSYS/QDTSSFL)	
							<u>.</u>
				-			
				OK	Restore de	faults Cancel	

- 3. In the Export list, select *ALL.
- 4. If you cannot see the Bind service program, select the Advanced check box.
- 5. In the **Bind service program** field, type qdtssfl.
- 6. In the Library field, type qsys.
- 7. Click Add.
- 8. Click OK.

This completes the compilation step. The service program is created.

You have created an RPG module and created a service program using that module.

Module recap

You have completed Chapter 2, "Module 2. Creating the iSeries service program for your Web application," on page 9. You have learned how to:

- Check the Remote System Explorer perspective is open
- Create a library filter
- Open an ILE RPG source member for edit
- Add code to an ILE RPG source member
- Create an RPG module

• Create a service program

Now that you have used the Remote System Explorer perspective to edit and compile an iSeries service program, you are ready to begin Chapter 3, "Module 3. Creating the customer list input page," on page 19.

Chapter 3. Module 3. Creating the customer list input page

This module teaches you how to create a new input page using a table iSeries Web component that works similar to a subfile and link that new input page to the customer inquiry input page. You will also learn how to add the iSeries business logic (service program) and additional parameter to the existing Web application in order to process the request to get a customer number.

In this module, you will:

- · Start the product
- Set the workspace location
- · Check the Web perspective is open
- Create a JSP file
- Add an iSeries table component
- Select properties for the table
- Insert a form on the page
- · Add the iSeries button component
- Add iSeries button properties
- Add a heading to the page
- Link the inquiry page to the customer list page
- Create another Web interaction to use the new input page and output pages
- · Begin the process to add a program to the Web interaction
- Create an iSeries connection
- Select the service program
- Add an output parameter
- Map an output parameter to an output field

Exercises

The exercises in this module must be completed in order.

- "Exercise 3.1: Starting the product workbench" on page 20
- "Exercise 3.2: Opening the Web perspective" on page 21
- "Exercise 3.3: Creating the customer list page" on page 22
- "Exercise 3.4: Linking to the customer list page" on page 37
- "Exercise 3.5: Creating another Web interaction" on page 40

Time required

This module will take approximately 30 minutes to complete.

Exercise 3.1: Starting the product workbench

Skip this exercise if you imported WSSLABxx.zip or if you have already started the product workbench.

Now go ahead and start the product.

To start the product:

 On the desktop task bar, click: Start > Programs > IBM[®] Rational[®] > IBM WebSphere Development Studio Client for iSeries V6.0 > WebSphere Development Studio Client for iSeries.

🔚 IBM WebSphere Development Studio Client for iSeries V6.0 🔸 🚳 WebSphere Development Studio Client for iSeries 🛛

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

2. Name the directory of the workspace as shown. Use the directory name WSSLABxx. This is the same directory that you used for the tutorial called Build a Web user interface for an RPG application using iSeries Web Tools.

Workspace L	auncher							×
Select a worl	kspace							
IBM Rational Select the wo					ojects in a dir	ectory called	a workspa	ace.
Workspace:	C:\Document	s and Sett	tings\Admin	nistrator\IBM	rationalsdp6.	0\wsslabxx	<u>•</u>	Browse
🗖 Use this a	s the default	and do nol	t ask again					
							ov	Correct [
							ОК	Cancel

- **Note:** You must use the same workspace used for the previous tutorial as you need to access an existing Web project and Web application that you created in the previous tutorial called Build a Web user interface for an RPG application using iSeries Web Tools.
- 3. Click OK.

After a few moments of loading, the workbench opens, and the initial product window.



You have started the product and set the workspace location and now you are ready to begin "Exercise 3.2: Opening the Web perspective."

Exercise 3.2: Opening the Web perspective

Before you begin, you must complete "Exercise 3.1: Starting the product workbench" on page 20.

Skip this exercise if you imported WSSLABxx.zip.

In this exercise you will use the Web perspective since it gives you access to unique views and tools targeted towards Web tasks.

You first need to open the Web perspective to create a new customer list page. The perspective should already be open, but if not, check the workbench taskbar on the left for the Web perspective icon, otherwise follow the steps below.

To open the Web perspective:

1. Click **Window** > **Open Perspective** > **Other** from the workbench menu.

© Remote System Explorer - IBM Rational File Edit Navigate Search Project Run	Software Development P Window Help	latform
1 1 · I · I · V · Q · Q · I	New Window	Remote Syste
Remote Systems 🛛 Team	Open Perspective Show View	ISeries Projects Image: Remote System Explorer
GRDENTR.*srvpgm.rp(GrDATAS.*module GrDATAS.*module GrDATAS.*module GrDATASC.*module GrDATASC.*module GrDATASC.*module	Customize Perspective Save Perspective As Reset Perspective Close Perspective Close All Perspectives	Cother
🔁 🔂 CELDIAL.*msgf	Navigation	
⊕ - ৣি INVMSGF.*msgf ⊕ - ৣি MYMSGF.*msgf	Preferences	

The Select Perspective dialogs opens.

∜>Debug	
🗄 iSeries Projects	
🐻 J2EE	
aj Java	
🕵 Java Browsing	
🖁 Java Type Hierarch	ny
📳 Remote System Ex	plorer (default)
Resource	
Web)	
webFacing	
	~
Show all	
I DITOTY OF	

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

The workbench shows the Web perspective with the Web Projects open in the Project Explorer.

You have checked that the Web perspective is open and now you are ready to begin "Exercise 3.3: Creating the customer list page."

Exercise 3.3: Creating the customer list page

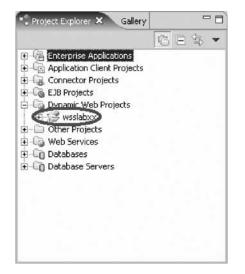
Before you begin, you must complete "Exercise 3.2: Opening the Web perspective" on page 21.

You will create the customer list page using several Page Designer iSeries Web components which will pass the customer number to the inquiry page (inquiry.jsp).

Creating a new customer list file

To create a new customer list file:

1. Locate the project WSSLABxx that you created in the previous tutorial.



- 2. Expand the WSSLABxx folder, if not already expanded.
- 3. Navigate to the WebContent folder in your project.
- 4. Right-click the WebContent folder and click **New** > **JSP File** on the pop-up menu.

Project Explorer 🗙 Gallery 🗖 🗖 🕞 inquiry.jsj	p WDSCSRV.R 🛛 🗖 3
The State Line	
E C WebCon	
E C MET	Project
⊕-	과 Java Project 毌 Package
	Class
	Gr Interface GS Source Folder
🕅 mylo 🔗 Refresh	💣 JSP File
- resul Run Validation - Timec Run - E u015 Debug	Faces JSP File

The New JSP File page opens.

New JSP File			×
ISP File Specify a name and	d location for the new JSP file.		4
Folder:	/wsslabxx/WebContent		Browse
File Name:	custList.jsp		
Markup Language:	HTML	-	
Options:	Create from page template		
	Create as JSP Fragment		
	🔲 Use XML Style Syntax		
Model:	None		
	Generate a new blank JSP page.	<u>ح</u>	
			Ę
Configure advan	iced options		

- 5. In the **File Name** field, type custList.jsp.
- 6. Click **Finish** to create a new JSP file.

The Page Designer opens in the workbench:

Web - custList.jsp - IBM Rational Softw	are Development Platform	_ 🗆 ×
File Edit Toolbar Insert JSP Format Ta	able Frame Page Tools Navigate Search Project	Run Window
Help		
📷 • 🗒 🛎] t] a 🚳 / 🖽 😡	Web
BIUATATA		ote Syste
8819.• 888 19.• 8		
None		
Project Explorer 🕅 Gallery 🗖 🗖	🛛 custList.jsp 🗙	
		H + BHTMLT
E WebContent	custList.jsp 📃 🕶 BODY 🕶 Standard	Form Tags
E-C= META-INF	Place content here.	JSP Tags
🕀 🗁 theme	period method for finite methods with States 201	iSeries
🕀 🗁 WEB-INF		SP Struts
- changePassword.jsp custList.jsp		Struts B
error.jsp		Struts L
- inquiry.jsp		Page T
		🖉 Web Sit
- RecordNotFound.jsp		🖪 Data 📌
- o result.jsp	Design Source Preview	Web
timeout isn	Properties 4% Servers 🛛 Console 2	Service
0 Page Data 🛛 Client 🄧 🗖 🗖		ED Session
	Server Host name	bean
	WebSphere Application Server v6.0 localhost	Relatio Record
		Record
		Record
		List
	•	Domino
/// wsslabxx/WebContent/custList.jsp		
Co		

Adding the iSeries table component

To add the iSeries table component:

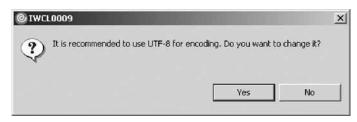
1. Click the **Palette** tab in the top right of the workbench.

🚱 Palatte 🗙 Snippets 🗖 🗖
HTML Tags
🕒 Form Tags
JSP Tags
👄 iSeries Web Components 🔹 🖈
Form
Simple Table
Button
🗹 Check Box
Combo Box
By Hyperlink
💽 Image
E Image Button
T Label
Radio Button Group
E Selection Box
Table
- Text Area
it Text Entry
Struts HTML Tags
🛞 Struts Bean Tags
😚 Struts Logic Tags
🕒 Page Template
🐣 Web Site Navigation
🕒 Data

- 2. Expand iSeries Web Components.
- Select Table and drag it to your JSP under the text "Place content here". Make sure the Design page is selected to place the table component on the new custList.jsp page.

You have selected the iSeries Table Component.

4. If you see this dialog, click Yes.



The Page Designer should look like this where the iSeries Table is on the Designer page.

💿 inquiry.jsp 💿 *custList.jsp 🗙 🥖	WDSCSRV.RPGLE		ć
custList.jsp - custList.jsp *	F	iwd:WTable 🔹	Standard
Place content here.			
4			
Design Source Preview			

Now you select the properties for the table.

Selecting properties for the table

To select properties for the table:

1. The Properties pane in the bottom left of the workbench should open automatically. If not, right-click on your table object and click **Properties** on the pop-up menu.

	sustLi 🗙 🖉 WDSCS		"₁ □	
	iwcl:WTable	• Star	idard .	
Pla	ce content here.			
101				
1000				
1.0	f Cut			
	Сору			
1	Paste			
3	🖇 Delete			
	Save			
De	Insert			+
Ē	JSP			•
- 4	🖉 Insert Link	Ctrl+Q		
	Page Properties			
B(-	Style			*
Ē	Embedded Document			•
£	Switch Active Document	in .		
	Properties	Ctrl+Al	t+Enter	

Next you fill in the general properties for the iSeries Table.

2. In the Properties pane, complete the following:

Properties ×	Quick Edit Servers	Console	Problems		•	- 4
BODY	Name:	STLIST)			
📰 iwd:WTable	Initial state:	Hidden				
Parameters	Orientation: (De	efault)	-			
Features	Summary:	10 49 10 1				
 Columns 						
 Styles 	Selection mode:					
	Formatting:					
	Get locale-sensitive v	alues fron	n: host system	values 👻		
	Decimal symbol:					
				N		
	Currency symbol:		\$	14		
	Thousands separato	r:	<i></i>			
	Date separator:		1			

- 3. In the Name field, type CUSTLIST.
- 4. Select **Single** from the **Selection mode** list.
- 5. Click the **Parameters** tab to set the parameters for the iSeries Table.

BODY	Get data from an iSeries service program	Service pro	gram parameters:	
iwcl:WTable	Library WSSLABxx	Name	Data type	Lengi
· Parameters				
Features	Service program behavior:			
Columns	O Writes a page at a time			
 Styles 	10 - 31 258			
	O Writes a page as requested			
	OWrites entire subfile			

To indicate that you want to provide the data by using a service program on the iSeries:

- 6. Select the **Get data from an iSeries service program** check box to specify the service program information.
- 7. In the **Library** field, type WSSLABxx.
- 8. In the **Object** field, type wdscsrv.
- 9. Click the **Writes entire subfile** radio button to indicate that you want to fill the table with all data available.

Writes a page at a time allows you to programmatically control page down and page up.

Writes a page as requested allows you to fill pages as needed.

If no new data is needed the paging will be handled by the table component. Next you set the features for the iSeries Table.

10. Click the **Features** tab.

Properties ×	Quick Edit	Servers	Console	Problems			•
BODY	Table featu	res:					
iwd:WTable	Show to	olbar					
Parameters	🗹 En	able table	e rows to	be collapsed and ex	xpanded		
Features	🗹 En	able sort	ing				
Columns	🗹 En	able filter	ing				
 Styles 	1	Show	filter row				
	Show h	eader					
	Show fo	oter				52	
	🗹 En	able pagi	ng			R	
	I	nitial pag	е: [
	F	lows per	page (10	0			
	🗹 Show g	ridlines					

This will switch to the Features pane.

- 11. In the **Rows per page** field, type 10.
- 12. Click the **Columns** tab.

This will switch to the Columns pane where you set the columns for the iSeries Table. There you specify the fields to be displayed.

IODY	-	-	1	1	1		
wd:WTable	Features	Туре	Properties	Header Style	Data Style	Add	2
Parameters						Remove	a
Features							
Columns	_					Up	
 Styles 						Davie	
	1					Down	
	- · · ·						

13. Click Add.

14. Fill in the following, and click **Add** after each row.

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

Your completed Properties pane for your table columns should look like this:

	Columns:							
BODY iwcl:WTable	Name	Length	Headi	ng text	Features	Туре	Properties	Header S
Parameters Features	custNumber custName	7 40	Custo Custo	mer N mer N	1 cocaros	Label Label	Troperces	Thedder 5
Columns	custPhone	17	Phone	e Number		Label		
Styles						1		
* ocyles								
							<i>⊾</i>	
							R	

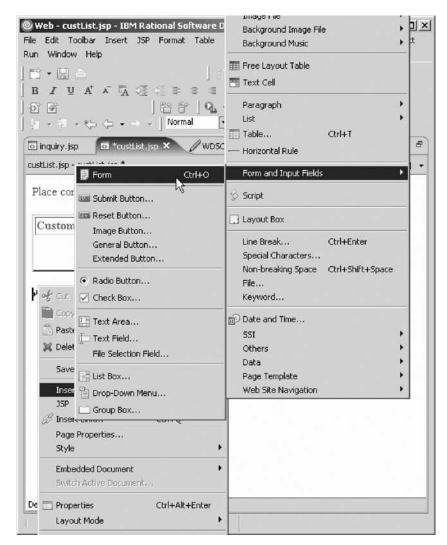
Now insert a form on the page.

Inserting a form on the page

An html form is needed to send back data with a request that you want to send the selected customer number back. You need a push button to send the request.

To insert a form on the page:

1. Place your cursor below the table in the Page Designer (if not already), right-click and then click **Insert** > **Form and Input Fields** > **Form** on the pop-up menu.



2. Place your cursor inside the dotted rectangle, which represents the form that you just created.

	tList.jsp 🗙 🦉 WDS	CSRV.RPGLE				đ
ustList.jsp - custList.jsp *			•	FORM ·	Standard	
Place content here.						
Customer Number	Customer Name	Customer P	hone N	umber		
J						
	2					

Next you want to access the Palette view again to select an **iSeries Button Component**.

Adding the iSeries button component

To add the iSeries Button Component:

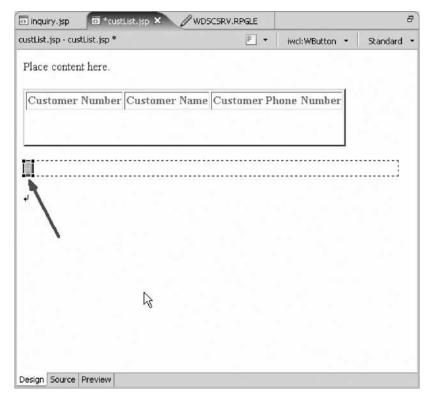
1. Navigate back to the Palette view. Click the **Palette** tab at the right of the workbench.

Palette X Snippets	- 8
HTML Tags	
🕒 Form Tags	
JSP Tags	
iSeries Web Component	> *
Form	
Emple Table	
Button	
Check Box N	
Combo Box Insert a But Component	ton web
By Hyperlink	
🖅 Image	
image Button	
T Label	
Radio Button Group	
E Selection Box	
Table	
Text Area	
Struts HTML Tags	
D Struts Bean Tags	
Struts Logic Tags	
🕒 Page Template	
🐣 Web Site Navigation	
🕒 Data	

2. Expand iSeries Web Components if not already.

3. Select **Button** then drag it to the form and drop it.

You have placed the iSeries Button inside the form on the page.



Next you complete the iSeries Button properties.

Adding iSeries button properties

To add iSeries Button properties:

1. The Properties pane for the button should automatically show up; if not, right-click the button and click **Properties** on the pop-up menu.

Properties ×	Quick Edit Servers Console Problems	
BODY	Name:	
FORM iwd:WButton	Label: Get Customer Number Button type: Submit	
• Styles	Initial state: Disabled Hidden Tool tip:	
	Access key: Tab index:	
	R	

- 2. In the **Name** field, type display.
- In the Label field, type Get Customer Number.
 You have finished creating and modifying the iSeries Web Components. Let's add a heading to this page.

Adding a heading to the page

Give the page a meaningful heading.

To add a heading to this page:

- 1. Select the text Place content here.
- 2. Replace it with the text Customer List.
- 3. In the Properties pane at the bottom of the workbench, select Heading 2.

Cust			
Custo Num		Customer Name	Customer Phone Number
2			
, Get (Tustom	er Numher	r;
Get (esign Si			·····;
esign S	ource		t »3 =
esign S	ource	Preview	t »3 =
esign S	ource ties ×	Preview Quick Edi Quick Edi	

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

inquiry.jsp			1
ustList.jsp - custList.jsp *	•	H2 •	Standard
Customer List			
Customer Number Customer Name Customer	Phone N	umber	
Get Customer Number			
\mathbf{k}			
esign Source Preview			

4. If you haven't done so already, save and close custList.jsp.

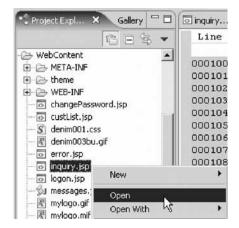
You have created a JSP file for the Customer List Web page, added an iSeries table component, selected properties for the table, inserted a form, added an iSeries button component, added iSeries button properties and added a heading to the Customer List Web page and now you are ready to begin "Exercise 3.4: Linking to the customer list page."

Exercise 3.4: Linking to the customer list page

Before you begin, you must complete "Exercise 3.3: Creating the customer list page" on page 22.

Now, let's modify the inquiry page. You will add a link beside the entry field so the end user can click this link (Find) to get the customer list page you created in the previous exercise.

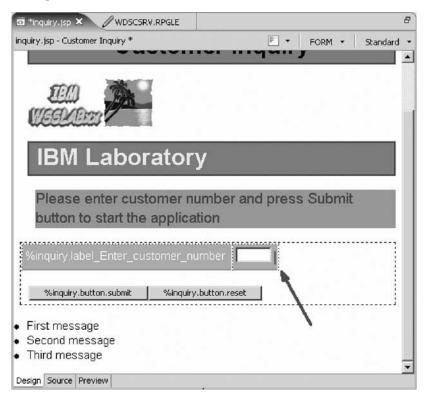
- 1. In the Project Explorer, right-click inquiry.jsp.
- 2. Click **Open** on the pop-up menu to edit the inquiry page named inquiry.jsp.



The Page Designer opens in the upper right pane of the workbench and shows the inquiry.jsp as the Web Interaction wizard created it and with the changes you made in Page Designer.

🗿 *inquiry.jsp 🗙 🖉 WDSCSR	₹V.RPGLE		ć
inquiry.jsp - Customer Inquiry *		MARC	QUEE - Standard
EEM) WEEKBaar			
IBM Labor Please enter cust	-	er and press S	ubmit
button to start the			-
%inquiry.label_Enter_cu	stomer_numbe	r 🗖	
%inquiry.button.submit	%inquiry.buttor	n.reset	
 First message 			
Second message			
 Third message 			
Design Source Preview			

- 3. Make sure that you are on the Design page in Page Designer. Click the **Design** tab.
- 4. Now put the cursor after the Enter_customer_number field.



5. Right-click the page and click Insert Link on the pop-up menu.



The Insert Link dialog opens.

ype • File		C FTP
E-mail	2	Brow
		Register as A
get:	*	

- 6. In the **URL** field, type custList.jsp.
- 7. In the **Link text** field, type Find.
- 8. Click OK.

The Customer Inquiry page after adding a link to customer list page (custList.jsp) should look like this:

🖾 *inquiry.jsp 🗙 🖉 WDSCSRV.RPGLE		Ð
inquiry.jsp - Customer Inquiry *	Ē ▼ A ▼	Standard 👻
3		^
Custome	er Inquiry	
EBM) WESLADIN		
IBM Laboratory		
Please enter customer num button to start the applicatio	an early and the set from the set of the set of the	t
%inquiry.label_Enter_customer_numl		
%inquiry.button.submit %inquiry.but	ton.reset	<i>₹</i>
Design Source Preview		

9. If you haven't done so already, save and close inquiry.jsp.

You linked the inquiry page to the customer list page and now you are ready to begin "Exercise 3.5: Creating another Web interaction."

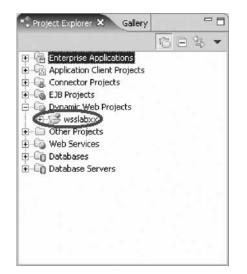
Exercise 3.5: Creating another Web interaction

Before you begin, you must complete "Exercise 3.4: Linking to the customer list page" on page 37.

Next you create another Web interaction to use the new input page and output pages that you just created.

To create another Web interaction:

1. Locate your project WSSLABxx in the Project Explorer.



2. Right-click on your project and click **New > Other** on the pop-up menu.

Project Expl X Gallery 🗖 🗖	WDSCSRV.RPG	GLE 🕄		N°1 □ E
18 B \$ -	Line 1	Column 1	Replace	A palette is not available.
Enterprise Applications Application Client Projects Connector Projects EJB Projects Dynamic Web Projects Symmetry Statement	000100 000101 000102 000103 000104		2+3 	avaliable,
⊕ Othe New			Project	
B Copy C Data D Data Paste X Delete			 D^o Java Project Deckage O Class 	
Refactor		Alt+Shift+T	G Interface	
Import Export			Source Folder	
Refresh			JSP File	
All Files (*.* Run Validation Specify iSeries Web Tools Run Debug	Run-time Configura	ation	Image Template File Image Template File Image File Image File	e
.classpath Profile Deploy Team Compare With			C Example	Ctrl+N

The Select page opens.

/izards:			
Web CSS File Dynamic Web Proj Faces JSP File Filter HTML/XHTML File JavaScript File J3P File Life-cycle Listener Page Template File Servlet Static Web Project	8	R	
Web Interaction Show All Wizards.	>		• (?)

- 3. In the left pane of the Select page, select Web.
- 4. In the right pane of the Select page, select Web Interaction.
- 5. Click Next.

The Web Interaction wizard opens.

Web Interaction -	New	2
Citra na fadi ta sa sana ang sa	Location for your Web Interaction location for the resources created for this Web Interaction.	031
Destination folder:	/wsslabxx/WebContent	Browse
Web Interaction name	getCustNumber	
lava package prefix:	com.ibm.wsslabxx	
Use error page	/error.jsp	Browse
Invalidate session a	after the interaction occurs	
Terminate iSeries c	onnection	
	< Back Next >	Finish Cancel

- 6. In the **Web Interaction name** field, type getCustNumber.
- 7. Click Next.

As you probably remember from the previous tutorial, this next page is where you specify the input and output pages.

Generate input page	Page preview:	
Use input pages:		
Custerse	Add	
	Preview	
	Rename	
Generate output pages Use output pages:		
(inquiry.)so	Add	
	Remove	
	Preview	
	Rename(G)	

- 8. Leave the Use input pages radio button selected and click Add.
- 9. Select custList.jsp on the Input page dialog. Click OK.

This is the subfile that contains the list of customers. It is the input page because from here you fill the customer number field with data from the selected table record.

- 10. Leave the Use output pages radio button selected and click Add.
- 11. Select inquiry.jsp on the Output JSP dialog. Click OK.
- 12. Click Next.

You have now defined your input and output pages.

Adding an iSeries ILE program to the interaction

Note: Some screens in this section will look different if you choose to use the GETDATAS service program instead of the GETDATA program to handle the Web interaction.

Next you will select the RPG service program you will use to handle your Web interaction.

ise an iSeries program or procedure ogram call definitions	C Use a Java bean	○ No program	n call
- ∰ / - ∰ (CSTRUC - ■ GETDATA	Cite Program Program alias: Program object: Library: Program type: Entry point: CCSID of entry point: Return type: Parse order: Thread safe: Source location: Associate this program	GETDATA GETDATA WSSLABXX "PGM Void CSTRUC FEEDBACK False	Browse V WEWJ
Import PCML Synchronize All			OK Cancel

To add a program to the Web interaction:

- 1. Leave the Use an iSeries program or procedure radio button selected.
- 2. Click Add Program.
- Click Browse to browse for the service program you will use. If you don't see your connection then you will need to create a connection. Follow the steps under the next section called Creating a connection. If you do see your connection then skip the next section and proceed to the section Selecting the service program.

Creating a connection

To create a connection:

- 1. In the list in the left pane, expand the New Connection list item by clicking on the plus sign + beside its icon.
 - The Name personal profile page opens.
- Accept the default profile. Click Next. The Remote iSeries System Connection page opens.

Perine connection information Parent profile: msteckh7tp Connection name 15400a 1 Description: Verify host name	x
Connection name Host name: Description:	=
Host name: 5400a I Description:	•
Verify host name	•
< Back Next > Finish Canc	. 1

- 3. Leave the default value for the **Parent Profile** field.
- 4. In the **Host Name** field, type the name of the iSeries server (Same server name as you specified for the runtime).
- 5. Click **Finish** to create a connection.

Selecting the service program

Now you select the RPG service program.

- 1. Expand your connection.
- 2. Expand *LIBL.
- 3. If you see a sign-on dialog, type your user ID and password.
- 4. If you don't see the WSSLABxx library, click **Work with libraries** and type WSSLABxx in the dialog, then click **OK**.
- 5. Expand the WSSLABxx library and select the WDSCSRV service program.

DSCSRV	
E- 🕅 QHLPSYS.*lib.prod-sys	
⊡ - M QUSRSYS.*lib.prod-sys	
庄 航 QGPL.*lib.prod-usr	
⊕ WFLABXX.*lib.test-usr	
E- 🕅 WSSLABXX.*lib.prod-usr	
DEMOR.*pgm.rpgle	
DEMORES.*pgm.rpgle	
🕀 💽 GETDASOL.*pgm.rpgle	
GETDASOLV.*pgm.rpgle	
GETDATA.*pgm.rpgle	
🔁 💽 GETDSSOLC.*pgm.cblle	
🔄 💽 READDTAQ.*pgm.rpgle	
😟 💽 WRTDTAQ.*pgm.rpgle	
GETDATAS.*srvpgm.rpgle	
🕀 腔 GETDATASC.*srvpgm.cblle	
🕀 🔊 GETDSSOL.*srvpgm.rpgle	
ORDENTR.*srvpgm.rpgle	
🕂 🔛 WDSCSRV.*srvpgm.rpgle	

6. Click OK.

Most of the fields have now been filled in for you. However you must fill in the values for the program alias and entry point fields.

Add Program		
Program alias:	getCustNumber	
Program object:	WDSCSRV	Browse
Library:	WSSLABXX	•
Program type:	*SRVPGM	•
Entry point:	readc	Browse
CCSID of entry point:		
Return type:	void	•
Parse order:		
Thread safe:	false	•
Source location:		▼ View
T Acception this provin	am with the interaction	

- 7. In the **Program alias** field, type getCustNumber.
- 8. In the Entry point field, type readc (case sensitive).
- 9. Click OK.

Adding a parameter

Let's add a parameter to the service program interface definition that you just added. You might remember the procedure readc returns the customer number as a parameter.

To add a parameter:

- 1. Under Program call definitions, select the getCustNumber program.
- 2. Click Add Parameter.
- 3. In the Parameter name field, type cnumber.
- 4. Leave character in the Data type list selected
- 5. In the **Length** field, type 7.
- 6. Select **Output** from the **Usage** list.

Your wizard should now look like this:

- 📑 getCustNumber -		Structure	
GSTRUC			
	Data type: Character Structure name: Length: 7 Precision: Count: Usage: Initial value: Advanced		R
	Specify database reference field	Specify	Synchronize
	Show database field definition	Show	

7. Click OK.

You have finished adding program parameters. Now you need to define the binding between the output parameter and the customer number field on the inquiry page.

8. Click Next.

put page fields:	Link	Unlink	Input parameters:	
CustList.jsp FORM:null CUSTLIST	Fields	Parameters		
roperty Value			Property	Value

- 9. You have no input parameters, so you don't need to modify this page.
- 10. Click Next again.
- 11. Under **Output parameters**, select cnumber.

Output parameters:			Unlink	Output page fields:		
Crumber		Parameters	Fields /inquiry.jsp-CUSTNOI	Inquiry.sp FORM:myForm submit reset Simple Table Usbal CUSTIN	USTNOL	
Property	Value				Value	1
	character(7)	_		Property		_
Data Type:	cnumber			Name in Session: Restore from Session:	CUSTNOI false	-
Name in Session:				Rescore more session:	1050	
Name in Session: Save to Session:	false					_
Name in Session:		-				

- 12. Under **Output page fields**, expand /inquiry.jsp > FORM:myForm > Simple Table and select CUSTNOI.
- 13. Click Link in the middle of the page.
- 14. Click **Finish** to link the output field to the output parameter.

This tells the wizard to generate code to move the data from the parameter cnumber to the customer number field on the inquiry page.

You have created an iSeries connection, selected the service program, added an output parameter and mapped an output parameter to an output field.

Module recap

You have completed Chapter 3, "Module 3. Creating the customer list input page," on page 19. You have learned how to:

- · Start the product
- Set the workspace location
- Check the Web perspective is open
- Create a JSP file
- Add an iSeries table component
- Select properties for the table
- Insert a form on the page
- Add the iSeries button component
- Add iSeries button properties
- Add a heading to the page
- Link the inquiry page to the customer list page
- Create another Web interaction to use the new input page and output pages

- Begin the process to add a program to the Web interaction
- Create an iSeries connection
- Select the service program
- Add an output parameter
- · Map an output parameter to an output field

Now that you have created a new input page from a subfile, linked that new input page to the customer inquiry input page and added the iSeries business logic (service program) and additional parameter to the existing Web application in order to process the request to get a customer number, you are ready to begin Chapter 4, "Module 4. Creating the update record output page," on page 53.

Chapter 4. Module 4. Creating the update record output page

This module teaches you how to create a new output page for your existing Web application. This page will allow end users to update customer information and will be generated from the Web interaction wizard. You will also learn how to handle errors during the update record process and to link the customer details page to the update record page.

In this module, you will:

- Create the update record page
- Add a program
- · Add parameters
- Design the output page
- Link the customer details page to the update record page
- Create an error page

Exercises

The exercises in this module must be completed in order.

- "Exercise 4.1: Generating the output page from the Web Interaction wizard"
- "Exercise 4.2: Linking to the update record page" on page 65
- "Exercise 4.3: Creating the update error page" on page 70

Time required

This module will take approximately 25 minutes to complete.

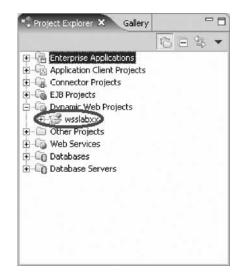
Exercise 4.1: Generating the output page from the Web Interaction wizard

You will create the update record page using the Web interaction wizard.

Creating the update record page

To create the update record page:

- 1. Open the Web perspective.
- 2. Locate your Web project in the Project Explorer.



3. Right-click your project and click **New > Other** on the pop-up menu.

Project Exp	X Gallery 🗖 🗖	WDSCSRV.RPGL	EX	٦	Nº1 - 0
	俗日生 -	Line 1	Column 1	Replace	A palette is not
F - Entern	rise Applications		+	2 + 3	available.
	tion Client Projects	000100	*		
+- Connec	방송 귀엽 이 가 있는 것 같은 것 같은 것 같은 것 같이 있는 것 같이 있다.	000101	* WDSCSRV		
EJB Pro	and the second se	000102	#		
	ic Web Projects	000103	* This progra		
E 😂 wa	slahxx	000104	* allows th	e user to wo	
🗄 🗀 Othe	New		,	Eg Project	
😟 - Co Web	Сору			🔎 Java Project	
🗄 🗋 Data 🖞	Paste			HP Package	
	🕻 Delete				
				Class	
	Refactor		Alt+Shift+T		
	Import			Source Folder	
	Export		•	5° HTML/XHTML File	
	Refresh			JSP File	
Thumbna	Close Project			Faces JSP File	
All Files (*.*	Run Validation			Page Template Fi	e
	A CONTRACTOR OF			🐨 JavaScript File	
1.11	Specify iSeries Web Tools	Run-time Configurat	ion	- Maria - Secondaria	
	Run		,	S CSS File	
	Debug		,	Image File	
.classpath	Profile		•	And the second second second	
lassbar	Deploy			Example	
	Team			He other	Children
	Compare With		•	📑 Other	Ctrl+N

The Select page opens.

Series Web Interaction Wizard			
lizards:			
-) Ce Web		 	
CSS File			
	ct		
Faces JSP File			
Image File			
- 🐼 JSP File			
Life-cycle Listener		N	
- Servlet		45	
Static Web Project			
Woh Discuss			
Web Interaction	>		-
Show All Wizards.			?

- 4. In the left pane of the Select page select Web.
- 5. In the right pane of the Select page, select **Web Interaction**.
- 6. Click Next.

The Web Interaction wizard opens.

Web Interaction	New	x
e 27 he and de la de la construction de la series de	Location for your Web Interaction location for the resources created for this Web Interaction.	233
Destination folder:	/wsslabxx/WebContent	Browse
Web Interaction name	custDispUpdate	
Java package prefix:	com.ibm.wsslabxx	
☑ Use error page	/error.jsp	Browse
🗖 Invalidate session	after the interaction occurs	
Terminate iSeries o	onnection	
	< Back Next > Finis	cancel

- 7. In the **Web Interaction name**, type custDispUpdate.
- 8. Click Next.

This next page is where you specify the input and output pages.

Web Interaction - custDispUpdate		×
Specify the Input and Output Pages for your We You can use previously designed input and output pag	b Interaction es for this Web interaction or you can have the wizard generate them.	600
Generate input page Use input pages:	Page preview:	
Generate output pages Use output pages: (ToustDispUpdeteResults.jsp)	Rename Add	
	Remove Proview Rename(C)	
Generate input/output pages using a page templat	te	
	< Back Next > Finis	6 Cancel

You want to get from the current results page to the update page. So you use the result page as an existing input page and let the Web Interaction wizard generate an update page for you.

- 9. Leave the **Use input pages** radio button selected and click **Add**.
- 10. Select result.jsp on the Input dialog and click OK.

This is the page your original Web Interaction generated; you are now linking it with another Web Interaction.

- 11. Click the **Generate output JSP** radio button to tell the Web Interaction wizard to make an output page.
- 12. Click Next.

On the next page, you will need to add another program instance and define the input and output parameters for this Web Interaction.

You will notice that the structure and the program from your previous Web Interaction are visible, and available for your use.

Adding a program

To add a program:

1. Click Add Program.

The service program has several procedures that we have already coded for you. One of these procedures is dispUpdate. It gets the customer detail data from a database file given a customer number.

2. To invoke procedure dispUpdate fill the dialog as shown and click **OK** when finished.

Add Program	\frown	
Program alias:	custDispUpdate	
Program object:	WDSCSRV	Browse
Library:	WSSLABXX	•
Program type:	*SRVPGM	•
Entry point:	dispUpdate	Browse
CCSID of entry point:		
Return type:	void	• N
Parse order:		\/\
Thread safe:	false	· ·
Source location:		▼ View
Associate this program	m with the interaction	

3. Click **OK** when finished.

You will see that the program has been added under the **Program call definitions**.

Adding parameters

To add parameters:

- 1. Click Add Parameter.
- 2. The input parameter is customer number. To define this parameter fill in details as shown:

Use an iSeries program or procedure rogram call definitions Constructions Construction Construct	Add Parameter Parameter name: Data type: Character Structure name: Length: Precision: Count:	C No program cal	
	Usage: Initial value: Advanced Specify database reference field Show database field definition	Spedfy	Synchronize
Import PCML Synchronize All			

3. Click **OK** when finished.

Now you need to add the structure you defined earlier.

- 4. Click Add Parameter.
- 5. The procedure returns a structure with detail data. To define this parameter fill in the details as shown:

6. Click OK.

Your wizard page should now look like this:

se an iSeries program or procedure	C Use a Java bean	C No program call	
gram call definitions	Add Program Add Parameter Add	Structure	
CSTRUC CustDispUpdate	Add Parameter Parameter name: Data type: Character Structure name: CSTRUC Length: Precision: Count: Usage: Initial value: Initial value:		
	Advanced Specify database reference field	Specify	Synchronize
	Show database field definition	S[20W)).	
nport PCML Syncheonize All			

You can see the new parameters you have defined for this interaction.

7. Click Next.

The input parameter page opens.

iput page fields:			Un/nk	Input parameters:	
FORM:myFi		Fields Uresuit.jsp-CSTRUC_CUSTNO	Paramete	Coumber	
			k⊊		
			18	Property	Value
operty	Value	1	18	Data Type:	character(7)
operty Name in Session: Save to Session:	Value CSTRUC_CUS false		18		

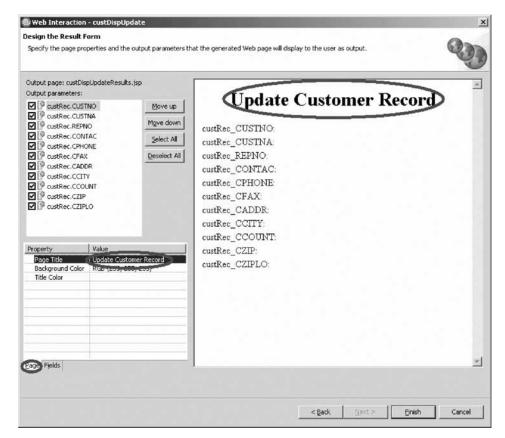
Now you will link the input page field to the program parameter.

- 8. Under Input page fields, select the CSTRUC_CUSTNO parameter.
- 9. Under Input parameters, select the cnumber parameter.
- 10. Click the **Link** button in the middle of the page.

You have just linked a field on the input page to the input parameter of our program.

11. Click Next.

Now its time to design the output page.



First you will change the output page title.

Designing the output page

To design the output page:

- 1. In the Properties table, select the **Page** tab.
- Change the value of the Page Title property to Update Customer Record. Now you need to change the field labels to make sense and to make the fields editable.

By default the output data is displayed as a label when the Web Interaction wizard generates the html for the page. Since you want to allow changes to the data you need to change all field attributes from Input Style: Label to Input Style: Text Entry.

3. Click the Fields tab.

Output parameters:	dateResults.jsp		Update Customer Reco	hard	-
PostRec.CUSTNO PostRec.CUSTNO PostRec.CUSTNO PostRec.CUSTNO PostRec.CUSTNO PostRec.CUSTNO PostRec.CONTAC PostRec.CONTAC PostRec.CONTAC PostRec.CADDR PostRec.CADDR PostRec.CCIV PostRec.CCIV PostRec.CCIV PostRec.CCIV PostRec.CCIV PostRec.CCIV	Value	Move up Mgve down Select All Deselect All	Customer Number: custRec_CUSTNA: custRec_CEPNO: custRec_CONTAC: custRec_CPHONE: custRec_CFAX: custRec_CADDR: custRec_CCITY: custRec_CCOUNT: custRec_CCOUNT: custRec_CZIP:		
Component Name: Label: Data Type: Input Style: Input Choices: Size: Name in Session: Save to Session: Message Controller:	CustRec_CU Customer Nu character(T) Text Entry 7 custRec_CU faise faise		custRec_CZIPLO:	R	

- 4. Under Output parameters, select custRec.CUSTNO.
- 5. In the Properties table, change the value of the Label property to Customer Number.
- 6. Select **Text Entry** from the **Input Style** list.
- 7. Repeat for the following by changing their Label fields to their respective names in the table below, and by changing all their Input Style fields to Text Entry.
- 8. Make sure that the length of the field shown in the Size and Data Type fields are the same.

CUSTNA	Customer Name
REPNO	Representative Number
CONTAC	Contact
CPHONE	Phone Number
CFAX	Fax Number
CADDR	Address
CCITY	City
CCOUNT	Country
CZIP	Zip

Your page should now look like this:

Specify the page propert	ies and the output p	arameters ti	nat the generated Web page will display to the user as output.
Dutput page: custDispUp Dutput parameters: Dutput parameters: Dutput parameters: Dutput parameters: Dutput page: CUSTNO Dutput page: CUSTNO	Mg De	love up we down elect All select All	Update Customer Record Customer Number: Customer Name: Representative Number: Contact: Phone Number: Fax Number:
Property	Value	1 1	Address: City:
Component Name: Label: Data Type: Input Style: Input Choices: Size: Name in Session: Save to Session: Message Controller:	custRec_CZIP Zip character(10) Text Entry 10 custRec_CZIP false false		Country:Zip:
age Fields			

- 9. You don't want to allow the **custRec.CZIPLO** field to be updated. Clear it's check box.
- 10. Click Finish.
- 11. Navigate to your new page custDispUpdateResults.jsp and double-click the file.

The Page Designer opens.

12. Select the text field for the Customer Number.

The customer number field needs to be protected so no changes can be made to it.

Project Ex 🛛 Gallery 🖵 🗖	WD5CSRV.RPGLE *custDispUpdateResults.jsp ×
6日年 -	custDispUpdUntitled * 📄 🔹 iwd:WTextEntry 🔹 Standard 👻
WebContent WebContent WEB-INF Content WEB-INF Content Content	Update Customer Record %custDispUpdateResults.label_Customer_Number %custDispUpdateResults.label_Customer_Name % custDispUpdateResults.label_Representative_Number % custDispUpdateResults.label_Contect Design_Source Preview
Page Data 23 "4 C	Properties 🛛 Quick Edit Servers Console Problems
Page Data X *4	BODY Name: custRec_CUSTNO
E 🚱 Scripting Variables	FORM Input type: O Text O Password O Hidden
⊞-@ Struts	Size:
	TD Initial state: Disabled Read-only
	Initial value:
	· Data Orientation: (Default)

13. In the Properties pane, set the field to **Read-only**. Select the **Read-only** check box.

You don't want to allow the user to change the customer number.

14. Save and close the JSP.

You have created the update record page, added a program, added parameters and designed the output page and now you are ready to begin "Exercise 4.2: Linking to the update record page."

Exercise 4.2: Linking to the update record page

Before you begin, you must complete "Exercise 4.1: Generating the output page from the Web Interaction wizard" on page 53.

Now let's modify the customer details page so that you can link to the update record page.

Instead of having the default Submit and Reset buttons you want to add a link to return to the inquiry page and you want to add a push button to submit a request for invoking the dispUpdate procedure. You did create the Web interaction to generate this request in the previous exercise.

To link to the update record page:

1. In the Project Explorer, drill down to your result.jsp page and double-click.

WDSCSRV.RPGLE					
suic, jsp - Unaciea ·		-	Р×	Standard	-
Cus	tomer Details				
%result.label_Customer_Number					
%result.label_Customer_Name	CSTRUC_CUSTNA				
%result.label_Representative_Number	CSTRUC_REPNO				
%result.label_Contact	CSTRUC_CONTAC				
%result.label_Phone_Number	CSTRUC_CPHONE				
%result.label_Fax_Number	CSTRUC_CFAX			1	
%result.label_Address	CSTRUC_CADDR				
%result.label_City	CSTRUC_CCITY				
%result.label_Country	CSTRUC_CCOUNT				
%result.label_Zip	CSTRUC_CZIP				
Back to Customer Inquiry					
First message					
Sonond-manage					ļ

On the page you will see two push buttons at the bottom of the page, delete both.

- 2. Select the **Submit** button and press the **Delete** key.
- 3. Select the **Reset** button and press the **Delete** key. Now you add the link back to the inquiry page.
- 4. Type Back to Customer Inquiry.
- 5. Highlight the text Back to Customer Inquiry, right-click and click **Insert Link** on the pop-up menu.

WDSCSRV.RPGLE	*result.isp 🗙		
result.jsp - Untitled *	of Cut		Standard
3	Paste		-
	💢 Delete		
	Save		ils
na n	Insert		•
%result.label_Custom	JSP		•
%result.label_Custom	💋 Insert Link	Ctrl+Q	
%result.label_Represe	Page Properties Style	~	•
%result.label_Contact %result.label_Phone_	Embedded Document		•
%result.label_Fax_Nu %result.label_Address	Properties	Ctrl+Alt+Enter	
	Layout Mode		
%result.label_City %result.label_Country	→] Run to Line Add to Snippets	Ctrl+R	
%result.label_Zip	Edit Events		
Back to Customer Inqu	uy		-
 First message Second message 			

The Insert Link dialog opens.

6. In the **URL** field, type inquiry.jsp

File	C HTTP	C FTP	
C E-mail	C Others		
.; inquiry.j	sp		Browse
		Registe	er as Alia
text:			
get:	•		

7. Click OK.

Now you add the update push button that when clicked, will send a request to fetch the customer data and display the custDispUpdate page.

8. Put your cursor right before the **Back to Customer Inquiry** link and press **Enter** on your keyboard.

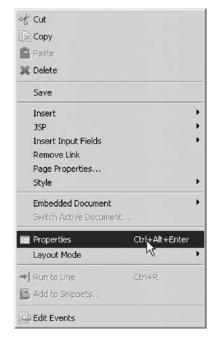
esult.jsp - Untitled *			FORM ·	Standard	
Cu	stomer Details	5			100
%result.label_Customer_Number					
%result.label_Customer_Name	CSTRUC_CUSTNA				
%result.label_Representative_Numbe					
%result.label_Contact	CSTRUC_CONTAC			1	
%result.label_Phone_Number	CSTRUC_CPHONE				
%result.label_Fax_Number	CSTRUC_CFAX				
%result.label_Address	CSTRUC_CADDR				
%result.label_City	CSTRUC_CCITY				
%result.label_Country	CSTRUC_CCOUNT			1	
%result.label_Zip	CSTRUC_CZIP				
ე∢	***************************************				
Back to Customer Inquiry					
First message					
Second message					
Third message					

- 9. In the right pane, click the **Palette** tab and expand **iSeries Web Components** if not already expanded.
- 10. Click and drag the **Button** component to the location shown.

WDSCSRV.RPGLE 0 *result.jsp X		□ Palette 🛛 "2 🖓
result. jsp - Untitled *	▼ iwcl:WButton ▼ St	tandard 👻 HTML Tags
%result.label_Address	CSTRUC_CADDR	Form Tags SP Tags
%result.label_City	CSTRUC_CCITY	iSeries Web Components
%result.label_Country	CSTRUC_CCOUNT	Form
%result.label_Zip	CSTRUC_CZIP	PT Simple Table
Back to Customer Inquiry First message Scond message Third message Design Source Preview		Check Box Combo Box Hyperlink Image Image Button Cabel Radio Button Group

The Properties pane in the bottom left of the workbench should automatically appear.

11. If not, right-click the button and click **Properties** on the pop-up menu.



12. Fill in the Button's details as shown below:

Properties ×	Quick Edit Servers Console Problems	□ ▼ □ ₽
BODY FORM P A iwd:WButton · Events · Styles	Name: Label: Update Customer Record Button type: Submit Initial state: Disabled Hidden Tool tip:	
	R	

Your Customer Details page should now look like this:

WDSCSRV.RPGLE 🖸 *result.jsp 🗙					ć
esult.jsp - Untitled *		F •	Р •	Standard	
2					
Custor	mer Details	5			
,	. <u></u>				
%result.label_Customer_Number					
%result.label_Customer_Name	CSTRUC_CUSTNA				
%result.label_Representative_Number	CSTRUC_REPNO				
%result.label_Contact	CSTRUC_CONTAC				
%result.label_Phone_Number	CSTRUC_CPHONE				
%result.label_Fax_Number	CSTRUC_CFAX				
%result.label_Address	CSTRUC_CADDR				
%result.label_City	CSTRUC_CCITY				
%result_label_Country	CSTRUC_CCOUNT				
%result.label_Zip	CSTRUC_CZIP				
		8			-
ب Update Customer Record				1	
Back to Customer Inquiry					10
Design Source Preview		_			1

13. Save and close the file.

You have linked the customer details page to the update record page and now you are ready to begin "Exercise 4.3: Creating the update error page."

Exercise 4.3: Creating the update error page

Before you begin, you must complete "Exercise 4.2: Linking to the update record page" on page 65.

Now let's create the error page that will be displayed if there is an error during the update process.

To create the update error page:

- 1. In the Project Explorer, expand your project if not already and drill down to the WebContent folder.
- 2. Right-click the WebContent folder inside your project.
- 3. Click **New** > **JSP File** on the pop-up menu. The New JSP File dialog opens.

ISP File Specify a name an	d location for the new JSP file.		4
Folder:	/wsslabxx/WebContent		Browse
File Name:	UpdateError.jsp		-
Markup Language:	HTML	•	
Options:	Create from page template Create as JSP Fragment Use XML Style Syntax		
Model:	None	•	
	Generate a new blank JSP page.	* *	
Configure advan	ced options		
	< Back Next >	Finish	Cancel

- 4. In the **File Name** field, type UpdateError.
- 5. Click Finish.

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

1
Standard

- 6. Select the words "Place content here" and change it to An Error Occurred During Update.
- 7. Highlight your new text.

This opens the Properties pane. It appears in the bottom right of the workbench.

WDSCSRV.RPG	LE 💿 *UpdateEr	ror.jsp 🖾	
UpdateError.jsp - I	UpdateError.jsp *		Standard 🝷
An Error	Occurred Du	ing Undate	
JAIL DITOL		ing optiate	
		1	
		/	
		/	
	1		
	1		
	1		
Design Source P	review		
🔲 Properties 🗙	Quick Edit Servers	Console Problems	
H3 Form	at: BDU	A A 2	144 144 1
Text	araph: Heading 3		
Paraç	raph: Heading 3 Heading 2		
Font:		Color:	
Style	Heading F	-	asses:
4			•

- 8. Select Heading 3 from the Properties pane.
- 9. Place your cursor below the heading you just created, and press the **Enter** key two times.
- 10. Type the text Back to Customer Inquiry.
- 11. Highlight the text and right-click. Now you will define the link.

WDSCSRV.RPGLE	🖸 *'UpdateErr	or.jsp 🗙			
UpdateError.jsp - Update	Error.jsp *	₹. •		Standard	•
An Error Occi	urred Du	ing Update			
		0 1			
ب					
₽ Back to Customer I	0.000				
Dack to Customer 1	of Cut				
	Cop	У			
	🖺 Pas	ie			
	36 Dele	ete			
Design Source Preview	Sav	e			
Properties 🛛 Qui	ck Edit Inse JSP	1977 A.		* ; * ;	
P Format:	B 💋 Inse	ert Link	Ctrl+Q		16

- 12. Click **Insert Link** on the pop-up menu. The Insert Link dialog opens.
- 13. In the URL field, type inquiry.jsp.

• File	C HTTP	C FTP
C E-mail	C Others	
; inquiry.j:	sp	Brow
		Register as A
text:		
get:	•	

14. Click OK.

Your UpdateError.jsp should now look like this:

WDSCSRV.RPGLE	🖬 *UpdateErro	or.jsp 🗙		
UpdateError.jsp - UpdateErr	ror.jsp *	₽. ▼		Standard 🔹
An Error Occur	red Dur	ing Upda	ıte	
ų.				
4			\mathbb{R}	
<u>Back to Customer Inc</u>	uny			
Design Source Preview				

15. Save and close the file.

You are now ready to use the Web Interaction Wizard to add the flow control parameter.

You have created an error page to handle records during the update record process.

Module recap

You have completed Chapter 4, "Module 4. Creating the update record output page," on page 53. You have learned how to:

- Create the update record page
- · Add a program
- Add parameters
- Design the output page
- Link the customer details page to the update record page
- Create an error page

Now that you have created the update record output page, you are ready to begin Chapter 5, "Module 5. Creating the update complete output page," on page 75.

Chapter 5. Module 5. Creating the update complete output page

This module teaches you how to create the customer update complete output page and how to invoke the RPG procedure that does the database update. You will also learn how to add an error page in case the RPG procedure can't complete the update successfully and to link the update page to the inquiry page.

In this module, you will:

- · Generate another output page
- Add a program
- Add a parameter
- Design the output page
- · Add a flow control error page
- Link the update complete page to the inquiry page
- · View the flow structure of your Struts-based Web application

Exercises

The exercises in this module must be completed in order.

- "Exercise 5.1: Generating another output page using the Web Interaction wizard"
- "Exercise 5.2: Adding the flow control error page" on page 85
- "Exercise 5.3: Linking the update complete page to the inquiry page" on page 88
- "Exercise 5.4: Visualizing the flow structure of your Web application" on page 90

Time required

This module will take approximately 15 minutes to complete.

Exercise 5.1: Generating another output page using the Web Interaction wizard

Now you will create the Web Interaction that will actually update the record on the iSeries. It will read the input fields from the custDispUpdateResults.jsp that you created in the previous exercise, pass the data in a structure as a parameter to the procedure doUpdate and display the new data in a Web page. In case of an error an error page gets displayed.

Generating another output page

To generate another output page:

1. Right-click your project and click New > Other on the pop-up menu.

Project Expl × Ga	llery 🗁 🗇 🖉 WDSCSRV.RP	GLE 🕄		Nº1 -0
C	🖹 🛬 👻 Line 1	Column 1	Replace	A palette is not
Connector Projects Connector Projects	s 000100 iects 000101 000102		am presents ε	available.
E- Othe New			Project	
e-Co Web e-Co Data e-Co Data Copy e-Co Data Copy Paste X Delete			 Java Project Package Class 	
Refactor		Alt+Shift+T	 G Interface 	
Import Export			Source Folder Source Folder Source Folder	
Thumbra Close Project	t		JSP File Faces JSP File	
2.2	on ies Web Tools Run-time Configur	ation	Page Template Fil	e
Run Debug Profile			CS5 File Timage File	
Team Compare W	ith		C Example	Ctrl+N

The Select page opens.

- 2. In the left pane, select **Web**.
- 3. In the right pane, select Web Interaction.

ielect a wizard		
iSeries Web Interaction Wizard		\square^{\diamond}
Wizards:		
 Web CSS File Dynamic Web Project Faces JSP File Filter HTML/XHTML File Image File JavaScript File JSP File Life-cycle Listener Page Template File Static Web Project Static Web Interaction Show All Wizards. 	ŀŝ	
< Back	Next > Finish	Cancel

4. Click Next.

5. In the **Web Interaction name** field, type custDoUpdate.

Web Interaction	- New	2
CTANNAS CLANARIES CANTANNS	Location for your Web Interaction	() ()
	Specify the name and loca	ation for the resources created fo
Destination folder:	/wsslabxx/WebContent	Browse,
Web Interaction name	custDoUpdate	
Java package prefix:	com.ibm.wsslabxx	
Use error page	/error.jsp	Browse
Invalidate session	after the interaction occurs	
Terminate iSeries o	onnection	
	< Back Next >	Finish Cancel

6. Click Next.

Now you define the input and output pages. You use the page from the previous exercise as the input page.

@Web Interaction - custDoUpdate		x
Specify the Input and Output Pages for your W You can use previously designed input and output pa	teb Interaction Iges For this Web interaction or you can have the wizard generate t	hem.
C Generate input page	Page preview:	
CustDispUpdateResults.jsp	Add	
	Remove	
	Proview	
	Rename	
Generate cutput pages		
C Use output pages:	Add	
	Remove	
	Preview	
	Rename(G)	
Generate input/output pages using a page temp	late	
	< Back Nex	Enish Cancel

- Leave the Use input pages radio button selected and click Add.
 You let the Web Interaction wizard generate the output page.
- 8. Select **custDispUpdateResults.jsp** on the Input page dialog. Click **OK**.

This is the page your original Web Interaction generated; you are now linking it with another Web Interaction.

- 9. Click the **Generate output JSP** radio button to tell the Web Interaction wizard to make an output page.
- 10. Click Next.

Now you will add a program.

Adding a program

To add a program:

- 1. Click Add Program.
- 2. Fill in the program details as shown.

Add Program	\sim	
Program alias:	custDoUpdate	
Program object:	WDSCSRV	Browse
Library:	WSSLABXX	•
Program type:	*SRVPGM	•
Entry point:	doUpdate	Browse
CCSID of entry point:		
Return type:	void	•
Parse order:		
		-
Thread safe:	false	×
Source location:		▼ View
Associate this progr	am with the interaction	

3. Click **OK** when finished.

You will see that the program has been added under **Program call definitions**. Now you need to add the structure you defined earlier.

Adding a parameter

The procedure doUpdate is provided to you in WDSCSRV. It receives a structure with the input data from the Web page and updates the customer record. It returns the string 'ok' in the parameter forward if the update completed successfully.

To add a parameter:

- 1. Click Add Parameter.
- 2. Define the structure for the customer data as an input/output parameter. Fill in the details as shown:

Use an Series program or procedure Program call definitions Import PCML Synchronize All Import PCML Use an Java bean Import PCML Use a Java bean Import PCML Import	e this page to define the input and output pa	rameters for your iSeries host program.			and
Show database field definition Show w	rogram call definitions	Add Program Add Parameter	Add Structure.	<u>.</u>	
					bynchronize
Import PCML Synchronize All Cancel		Show database field definition		Show,	
	Import PCML Synchronize All				Cancel

3. Click OK.

The feedback from the procedure to indicate success is a 20 character parameter. You use this parameter to decide which page to show next.

4. Fill in the parameter details as shown:

Use an iSeries program or procedure Program call definitions	C Use a Java bean Add Program Add Parameter Add Parameter Parameter name: Data type: Character Structure name: Length: 20	C No program cel	
·	Precision: Count: Usage: Initial value: Advanced Specify database reference field	Specfy	Synchronice
	Show database field definition	Showitt	ayinirinate.
Import PCML Synchronize All		success	OK Cancel

5. Click **OK** when finished.

Your wizard page should now look like this:

pecify the Input and Output Parameters Specify length.	for your iSeries Host Program		23
Use an ISeries program or procedure Program call definitions CustDoUpdate CustDoUpdate CustDoUpdate Groward Groward Groward Groward GeTDATA	Add Parameter Parameter name: Data type: Character Structure name: CSTRUC Length: Precision: Count: Usage: Initial value:	C No program cel	
	Advanced Specify database reference field Show database field definition	Specify	Synchroniste
Import PCML Synchronize All			OK Cancel

You can see the new parameters you have defined for this interaction.

6. Click Next.

The input parameter page opens.

nout page fields:			Unink	Input parameters:	
සා කි සා කි සා කි සා කි සා කි සා කි සා කි		/custDispUpdateResults.jspc /custDispUpdateResults.jspc /custDispUpdateResults.jspc /custDispUpdateResults.jspc /custDispUpdateResults.jspc /custDispUpdateResults.jspc /custDispUpdateResults.jspc /custDispUpdateResults.jspc /custDispUpdateResults.jspc	custRec.CUSTNA custRec.REPNO custRec.CONTAC custRec.CHONE custRec.CFAX custRec.CADDR custRec.CCDTY custRec.CCDIVT	CustRec.CUSTNA CustRec.REPNO CustRec.CONTAC CustRec.CONTAC CustRec.CADR CustRec.CADR CustRec.CCITY CustRec.CCITY CustRec.CCIP CustRec.CZIPLO	
				Property	Value
roperty Name in Session: Save to Session:	Value custRec_CZIP false			Data Type: Name in Session: Restore from Session: Get Browser Info.:	character(10) custRec_CZIP false false

To be able to generate the correct code, the Web Interaction wizard needs to know which field on the Web page is correlated to which parameter. You specify this correlation by linking the fields from the Web page to the correct parameter.

Now you will link the input page parameters.

- 7. Select an input page parameter under Input page fields.
- 8. Select the respective program input parameter under Input parameters.
- Click the Link button in the middle of the page You have just linked a field from the input page to the input parameter of your program.
- 10. Repeat for all remaining input page parameters.

Designing the output page

Now you design the output page.

To design the output page:

1. Click Next.

Output page: custDoUpdateRe: Output parameters:	ults.jsp		1
OustRec.CUSTNO OustRec.CUSTNA OustRec.CISTNA OustRec.CPTNE OustRec.CPHONE OustRec.CADDR OustRec.CADDR OustRec.CCIV OustRec.CCIV OustRec.CCIV OustRec.CCIP OustRec.CIP OustRec.CCIP OustRec.CIP OustRec.CIP	Move up Move down Select All Deselect All	custRec_CUSTNO: custRec_CUSTNA: custRec_CUSTNA: custRec_CONTAC: custRec_CPHONE: custRec_CFAX: custRec_CADDR: custRec_CCITY: custRec_CCOUNT: custRec_CCIP: custRec_CZIP: custRec_CZIP: custRec_CZIPLO:	
Title Color		forward.	

- 2. In the Properties table, click the **Page** tab.
- 3. Change the value of the Page Title property to Update Complete. Now change the field labels to be readable.
- 4. Click the **Fields** tab.

output parameters:	ateResults.jsp	Update Complete	1
Outraction O	Move up Move dow Select All Deselect A	vn Customer Number: L Customer Name:	
QustRec.CCITY QustRec.CCOUNT CustRec.CCOUNT CustRec.CZIP QustRec.CZIPLO P custRec.CZIPLO P forward		Phone Number: Fax Number: Address: City: Country:	
Property	Value	Zip:	
Component Name:	custRec_CZIP		
Label:	Zip abaratar(10)		
Data Type:	character(10)		
	Label		
Input Style:	10		
Input Choices:		-	
Input Choices: Size:	michDec (7710		
Input Choices: Size: Name in Session:	custRec_CZIP		
Input Choices: Size:	custRec_CZIP false false		

- 5. Select **custRec.CUSTNO** from the left.
- 6. Change the value of the Label property to Customer Number.
- 7. Using the table below, change the values of the remaining properties:

CUSTNA	Customer Name
REPNO	Representative Number
CONTAC	Contact
CPHONE	Phone Number
CFAX	Fax Number
CADDR	Address
CCITY	City
CCOUNT	Country
CZIP	Zip

Now, let's disable the values you don't want to be displayed.

As a final step, you choose not to use the CZIPLO field and not to display the forward field.

Output parameters: Update Complete CustRec.CUSTNA Move up Move down CustRec.CUSTNA CustRec.CUSTNA CustRec.CUSTNA Move down Select All CustRec.Contac Contact: Phone Number: Contact: Phone Number: Fax Number: Contact: Phone Number: Fax Number: Address: City: Country: CustRec.Contry: CustRec.C	CustRec.CUSTNO Move up CustRec.CUSTNA Move down CustRec.CONTAC Select All CustRec.CONTAC Select All CustRec.CONTAC Select All CustRec.CONTAC Select All CustRec.CONTAC Deselect All CustRec.COUNT Deselect All CustRec.COUNT Deselect All CustRec.COUNT Deselect All CustRec.CITY Contact: Phone Number: Country: CustRec.CITY CustRec.CITY Label: Zip Input Style: Label Input Style: Label Size: 10 Name in Session: custRec.CITP	OustRec.CUSTNO Move up Update Complete CustRec.CUSTNA Move down Customer Number: CustRec.CONTAC Select All Customer Number: CustRec.CADDR Deselect All Representative Number: CustRec.CADDR Contact: Phone Number: CustRec.CONT Contact: Phone Number: CustRec.CIPIO Fax Number: Contact: Property Valkec.CIPIO CustRec.CIPIO Component Name: custRec.CIPI Zip:	CustRec.CUSTNO Move up Update CustRec.CUSTNA Move down Customer Number: CustRec.CONTAC Select All Customer Name: CustRec.CPHONE Deselect All Representative Number: CustRec.CADRR Contact: Contact:	Complete
Property Value Zip: Component Name: custRec_CZIP Label: Zip	volue Zip: Component Name: custRec_CZIP Label: Zip Data Type: character(10) Input Style: Label Size: 10 Name in Session: custRec_CZIP	Property Value Zip: Component Name: custRec_CZIP	CustRec.CCUP Fax Number: CustRec.CZIP.0 Address: Forward City:	
Component Name: custRec_CZIP Label: Zip	Component Name: custRec_CZIP Label: Zip Data Type: character(10) Input Style: Label Size: 10 Name in Session: custRec_CZIP	Component Name: custRec_CZIP		
Label: Zip	Label: Zip Data Type: character(10) Input Style: Label Input Choices: Jabel Size: 10 Name in Session: custRec_CZIP		LAP.	
Data Type: character(10)	Input Style: Label Input Choices:	Label: Zip		
	Input Choices: Input Choices: Size: 10 Name in Session: custRec_CZIP	Data Type: character(10)	ata Type: character(10)	
	Size: 10 Name in Session: custRec_CZIP		n # Shula- Lahal	
	Name in Session: custRec_CZIP			
			nput Choices:	
			put Choices: ze: 10	
			put Choices: ze: 10 ame in Session: custRec_CZIP	
		Message Controller: raise	put Choices: 10 ame in Session: custRec_CZIP ave to Session: false	
		Message Controller: false	put Choices: ze: 10 ame in Session: custRec_CZIP	
		Message Controller: haise	put Choices: 10 ame in Session: custRec_CZIP ave to Session: false	

- 8. Clear their respective check boxes.
- 9. Click **Finish**.

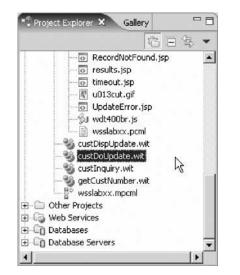
You have generated another output page, added a program, added a parameter and designed the output page and now you are ready to begin "Exercise 5.2: Adding the flow control error page."

Exercise 5.2: Adding the flow control error page

Before you begin, you must complete "Exercise 5.1: Generating another output page using the Web Interaction wizard" on page 75.

Now that the update complete page (custDoUpdateResults.jsp) is created, you can add the condition to this Web interaction to indicate which page to show next. If the update was successful, the update complete page is shown, otherwise the error page is shown. In order to add this feature you need to change the Web interaction you just created.

1. Drill down through your project and find the Web interaction file named **custDoUpdate.wit**.



2. Double-click the Web interaction file.

The Web Interaction wizard opens.

3. Click Next.

Next you add the error page to the possible output pages to be displayed. Now you can define when to display one or the other.

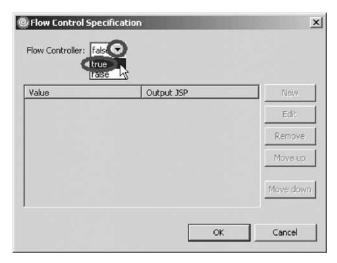
4. Click Add.

		ave the wizard generate them.	2.3
Generate input page	Page pr	eview:	
Use input pages:			
/custDispUpdateResults.jsp	Add		
2	Remove		
	Preview		
	1.16.00-01		
	Rename		
·			
Generate output pages			
Use output pages:			
/custDoLodateResults.jsp gUpdateError.jsp	Add.		
Duppaceerror.jsp	Remove		
	Preview	13	
	Remarme(C)		
Generate input/output pages using a page template			

- 5. Select UpdateError.jsp on the Output page dialog. Click OK.
- 6. Click **Next** until you reach the Map and Link the Output Parameters to the Output Fields page.

tput parameters:		Link	Unlink	Output page fields:	
CustRec.CUSTNO CustRec.CUSTNA CustRec.EPNO CustRec.EPNO CustRec.CHONE CustRec.CHONE CustRec.CADOR CustRec.CCITY CustRec.CCITY CustRec.CCITY CustRec.CCITY CustRec.CCIP CustRec.CZIP CustRec.CZIP CustRec.CZIP		Parameters sustRec.CUSTNO custRec.CUSTNA custRec.CUSTNA custRec.CONTAC custRec.CONTAC custRec.CADA custRec.CADA custRec.CCITY custRec.CCITY custRec.CCITY	Fields custDoLpdateResults.j custDoLpdateResults.j custDoLpdateResults.j custDoLpdateResults.j custDoLpdateResults.j custDoLpdateResults.j custDoLpdateResults.j custDoLpdateResults.j custDoLpdateResults.j custDoLpdateResults.j	E a custDoUpdateRe AlpdateError.jsp	
operty	Value	1			
Data Type:	character(20)			Property	Value
Name in Session:	forward				
	false 💦	2			
Save to Session:					
Save to Session: Flow Controller: Message Controller:	false	0			

- 7. Under **Output parameters**, select the forward parameter.
- 8. Select the value **true** for the Flow Controller list.



9. In the Flow Control Specification dialog, select **UpdateError.jsp** in the **Output JSP** field.

ок	and a standard state of the sta	New
The second	custDoUpdateResults.jsp	
*OTHER	/UpdateError.jsp	Edit
		Remove
		Move up

- 10. Click New.
- 11. Type 0K in the **Value** field and select **custDoUpdateResults.jsp** as the Output jsp
- 12. Click OK.

You return to the Web Interaction wizard.

13. Click Finish.

You have added a flow control error page and now you are ready to begin "Exercise 5.3: Linking the update complete page to the inquiry page."

Exercise 5.3: Linking the update complete page to the inquiry page

Before you begin, you must complete "Exercise 5.2: Adding the flow control error page" on page 85.

Next you edit the update complete page (custDoUpdateResults.jsp) to add the link to the inquiry page.

To link the update complete page to the inquiry page:

1. Navigate to your **custDoUpdateResults.jsp** file and double-click. The Page Designer opens.

ustDoUpdateResults.jsp - Untitled	i.	•	BODY -	Standard	1
8					1 al
Update Co	mplete				
%custDoUpdateResults label_Customer_Number	custRec_CUSTN	o			
%custDoUpdateResults.label_Customer_Name	custRec_CUSTN	A			
%custDoUpdateResults.label_Representative_Numb	er custRec_REPNC				
%custDoUpdateResults.label_Contact	custRec_CONTA	C			
%custDoUpdateResults.label_Phone_Number	custRtc_CPHO1	ΙE			
%custDoUpdateResults.label_custRec_CFAX	custRec_CFAX				
%custDoUpdateResults.label_Fax_Number	custRec_CADDI	2			
%custDoUpdateResults.label_City	custRec_CCITY				
%custDoUpdateResults.label_Country	custRec_CCOU1	T			
%custDoUpdateResults.label_Zip	custRec_CZIP				
%custDoUpdateResults.button.submit %custD	oUpdateResults.button.r	eset	\mathbf{P}		
				······	

- 2. Select and delete the **Submit** and **Reset** buttons.
- 3. Type Back to Customer Inquiry where the Submit and Reset buttons were shown.

	ł
F	P 🔹 🛛 Standard
omplete	
custRec_CUSTNO	······
custRec_CUSTNA	
custRec_REPNO	
custRec_CONTAC	
custRec_CPHONE	
custRec_CFAX	
custRec_CADDR	
custRec_CCITY	
custRec_CCOUNT	
custRec_CZIP	
	custRec_CUSTNO custRec_CUSTNA custRec_CUSTNA custRec_CONTAC custRec_CPHONE custRec_CPHONE custRec_CFAX custRec_CADDR custRec_CCITY custRec_CCOUNT

- 4. Highlight the text and right-click.
- 5. Click **Insert link** on the pop-up menu.

of Cut	PGLE		
Copy		F +	Standard
💼 Paste			1
💥 Delete			
Save	C.		
Insert JSP	; e Col	mplete	
Insert Input Fields		custRec_CUSTNO	1
💋 Insert Link Ctrl+Q	lame	custRec_CUSTNA	
Page Properties.	ve Numbe	r custRec_REPNO	
Style		custRec_CONTAC	
Embedded Document Switch Active Document	ber	custRec_CPHONE	
2	AX	custRec_CFAX	
Properties Ctrl+Alt-	FEnter	custRec_CADDR	
		custRec_CCITY	
→] Run to Line Ctrl+R	**********	custRec_CCOUNT	
Add to Snippets		custRec_CZIP	
Edit Events			
Back to Customer Inquiry			

The Insert Link dialog opens.

6. In the **URL** field, type inquiry.jsp.

File	C HTTP	C FTP
C E-mail	C Others	
JRL: inquiry.;	sp	Browse
		Register as Alias.
ink text:		ks.
Target:	•	18

- 7. Click OK.
- 8. Save and close the JSP file.

You have linked the update complete page to the inquiry page and are ready to begin "Exercise 5.4: Visualizing the flow structure of your Web application."

Exercise 5.4: Visualizing the flow structure of your Web application

Before you begin, you must complete "Exercise 5.3: Linking the update complete page to the inquiry page" on page 88.

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.

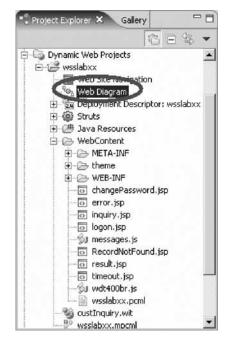
Now, that you know what Struts is, you can go ahead with this exercise.

You will now update the Web diagram from the previous Web Tools tutorial (Build a Web user interface for an RPG application using iSeries Web Tools) to show the new visual components of the Web application you just created.

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

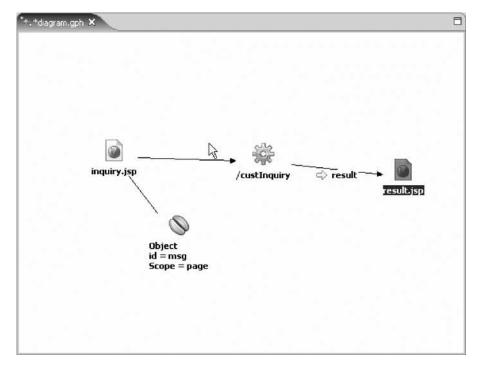
To create a Web diagram:

1. If you don't see Web Diagram in the editor, in the Project Explorer view, expand WSSLABxx.



2. Locate and double-click Web Diagram.

You will now see the Web Diagram in the editor.



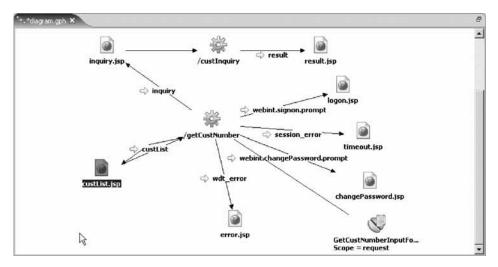
Now, let's update the Web diagram:

- 1. Locate custList.jsp under WSSLABxx and then WebContent in the Project Explorer.
- 2. Click custList.jsp and drag it to Web Diagram. You will notice the cursor

change from $oldsymbol{O}$ to $oldsymbol{B}$.

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

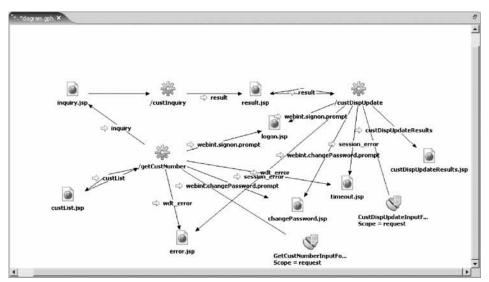
- Click Draw > Draw All From on the pop-up menu. You will now see the getCustomerNumber interaction.
- 5. Right-click getCustomerNumber and click Draw > Draw All From.



- 6. Right-click result.jsp.
- 7. Click **Draw** > **Selected** on the pop-up menu.
- Select the custDispUpdate interaction.
 You will now see the custDispUpdate interaction.
- 9. Right-click custDispUpdate interaction.
- 10. Click Draw All From on the pop-up menu.

You should now see the following graphical representation of the Web interaction so far.

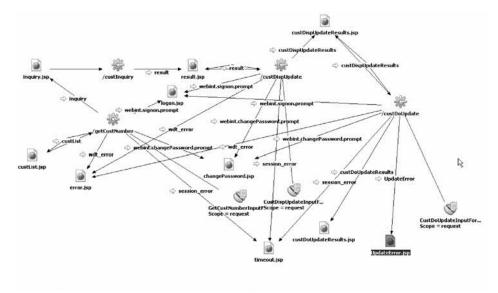
You may want to rearrange the objects in the editor if the objects are overlapping.



- 11. Right-click custDispUpdateResults.jsp.
- 12. Click **Draw** > **Selected** on the pop-up menu.
- 13. Select the custDoUpdate interaction.
- 14. Right-click custDoUpdate interaction.
- 15. Click Draw All From on the pop-up menu.

You now see the complete graphical representation of all the Web interactions that you created in all the previous exercises.

You may want to rearrange the objects in the editor if the objects are overlapping.



16. Save and close Web Diagram.

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

Module recap

You have completed Chapter 5, "Module 5. Creating the update complete output page," on page 75. You have learned how to:

- Generate another output page
- · Add a program
- Add a parameter
- Design the output page
- Add a flow control error page
- Link the update complete page to the inquiry page
- View the flow structure of your Struts-based Web application

Now that you have created the update complete output page, you are ready to begin Chapter 6, "Module 6. Running the Web application," on page 95.

Chapter 6. Module 6. Running the Web application

This module teaches you how to start the application server and test the updated Web application.

In this module, you will:

- Start the application server
- Test the customer list page
- · Test the update record and update complete pages

Note: To access iSeries data and resources you will need to have a job description setup and associated with your user profile on your iSeries system.

Exercises

The exercises in this module must be completed in order.

- "Exercise 6.1: Restarting the server"
- "Exercise 6.2: Testing the Web application" on page 96

Time required

This module will take approximately 10 minutes to complete

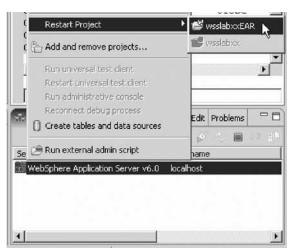
Exercise 6.1: Restarting the server

If the server is still running from the last time you ran the Web application, it needs to be restarted. The server will pick up changes in the Web application only when it is restarted. If the server is not started, then you must start the server before you can restart the project.

- 1. In the Web perspective locate the Servers view at the bottom of the workbench. Click the **Servers** tab.
- 2. Right-click the server and select Start on the pop-up menu.

The server will show Started status when the start operation completes.

3. Right-click the server and select Restart Project on the pop-up menu.



- 4. Then click your project on the pop-up menu.
- 5. Continue with "Exercise 6.2: Testing the Web application."

You have started the application server and now you are ready to begin "Exercise 6.2: Testing the Web application."

Exercise 6.2: Testing the Web application

Before you begin, you must complete "Exercise 6.1: Restarting the server" on page 95.

First you test the customer list page and then you test the update record and update complete pages.

Testing the customer list page

To test the customer list page:

1. Right-click inquiry.jsp and click Run > Run on Server on the pop-up menu.

Customer Inquiry	1
EEM WEST ABOUT	
IBM Laboratory	
Please enter customer number and press Submit button to	
Enter customer number	
Submit Reset	M

2. Click the **Find** link.

The Customer list page opens.

3. Select the first entry on the list.

++++ +	1 P = =	Select Action 🗾 Go	
Select ^	Customer Number \land	Customer Name 🔿	Custo
۲	0010100	Meridien Electronics Limited	206-
0	0010200	Royal Hardware Supplies	905-
0	0010300	Webster Appliances	619-
0	0010400	ProLine Building Supplies	905-
0	0010700	Universal Communications Ltd.	415-
0	0010800	Baker Electronics	818-
0	0010900	Village Telephone	707-
0	0011100	BelAir Communications incorp.	914-
0	0011300	Burnham Trading Inc	613-
0	0011400	Calderone Imports	407-
Page	1 of 8 D 1	Go Total: 71 Filtered: 71 Display	/ed: 10 S

4. Click the Get Customer Number button.

Customer Inquiry
EED WEST COTT
IBM Laboratory
Please enter customer number and pres
Enter customer number: 0010100
Submit Reset

5. Click the **Submit** button.

The Customer details page opens showing the details of the selected customer.

Customer Number:	0010100	
Customer Name:	Meridien Electronics Limited	
Representative Numb		
Contact:	Alfredo Bayonne	
Phone Number:	206-865-4027	
Fax Number:	206-865-4037	
Address:	10423 S.E. 30th Place	
City:	Bellevue, WA	
Country.	USA	
Zip:	98007	
Update Customer	Record	
Back to Customer Inqu	uiry	

Testing the update record and update complete pages

To test the update record and update complete pages:

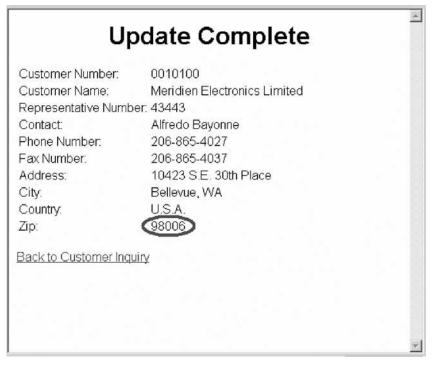
1. When the information is displayed, click the Update Customer Record button.

-

2. Change some of the fields and click the **Submit** button.

	0010100
Customer Name:	Meridien Electronics Limited
Representative Number:	43443
Contact:	Alfredo Bayonne
Phone Number:	206-865-4027
Fax Number:	206-865-4037
Address:	10423 S.E. 30th Place
Dity:	Bellevue, WA
Country:	U.S.A.
Zip:	98006

You should see the confirmation page with the new values.



3. Click the **Back to Customer List** link and you will return to the customer list page.

You have tested the update record page and the update complete page.

Module recap

You have completed Chapter 6, "Module 6. Running the Web application," on page 95. You have learned how to:

- Start the application server
- Test the customer list page
- Test the update record and update complete pages

Finish the tutorial by reviewing the materials in Chapter 7, "Summary," on page 101.

Chapter 7. Summary

In this tutorial, you learned how to create a new Web page containing a table component to list customers (custList.jsp), use an existing output page (result.jsp) to add an additional Web interaction, and create new Web pages to update customer records and confirm updates were completed on the iSeries (custDispUpdateResults.jsp and custDoUpdateResults.jsp) for an existing e-business RPG customer inquiry application. While you created these pages, you learned how to use the iSeries wizards in Development Studio Client to link pages together to form an application with multiple Web interactions. You then learned how to run the updated customer inquiry application in the WebSphere Test Environment that is part of Development Studio Client.

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

- Select an RPG service program to work with the Web interactions
- Create a Web input page with a table component using Page Designer
- · Create Web output pages using the Web interaction wizard
- Run the updated Web application in the WebSphere Test Environment

More information

For more information on Development Studio Client 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, 2005. 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 Rational 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