An A – Z hands-on guide to IBM WebFacing Tool V6.0.1 Advanced Edition lab

By
Michael J. Sandberg

WebSphere Development Studio Client Version 6.0.1 updates
provided by
Jim Bainbridge

Special thanks

Micheal would like to give a special thanks to Larry Schweyer, Maha Masri, Satish Gungabeesoon, Claus Weiss, and Tina Tang for their help with the original content and reviews of this lab.

Table of contents

Introduction to this tutorial

You can use the IBM® WebFacing Tool to create Web front ends to IBM System i applications that use data description specification (DDS) for their green-screen transactions. You choose a Web style for your green-screen application, generate a set of JavaServer Pages™ (JSP™) and XML files that interact with the logic of your program, and then easily test your application in the IBM WebSphere® test environment of the workbench. When you are ready to deploy your application, you can generate standard Java™ 2 Platform, Enterprise Edition (J2EE™) Web archive (WAR) and Enterprise archive (EAR) files that can be installed on a WebSphere Application Server.

The IBM WebFacing Tool is ideal for applications that you want to deploy broadly over a corporate intranet or the Internet, where rapid deployment takes precedence over customizing the look and feel of each page. With the IBM WebFacing Tool, you can continue to deliver your application as a 5250 application and use the same Integrated Language Environment (ILE) and non-ILE programs to deliver the application through the Web. Note that there are some restrictions, described in the IBM WebFacing Tool documentation, on which DDS keywords can be converted to generate sets of JSP and XML files.

IBM WebSphere Development Studio Client for iSeries CODE Designer has been enhanced to work with the IBM WebFacing Tool. A new “Web Settings” section helps you specify how your DDS fields are handled by a browser after they are converted using the IBM WebFacing Tool, without affecting their appearance on the green-screen.

Tutorial purpose

Before you begin, take a minute to read the learning objectives for the tutorial.

Chapters

Work on the chapters in sequence. The pictures in the chapters show similar tasks. Some of the names and icons might be different from the environment in which you work when you complete the chapters.

Prerequisites

You also learn about the prerequisite knowledge for the tutorial before beginning. This section describes the type of knowledge that is required to benefit from this tutorial.

Learning objectives

In this tutorial, you convert DDS screens in an existing 5250 order entry application to a Web-browser user interface. While creating a browser user interface, you learn how to use the IBM WebFacing Tool. You also learn how to customize the output of the conversion process using the IBM WebFacing Tool and CODE Designer. You learn how to run the WebFacing application in the WebSphere test environment that is part of WebSphere Development Studio Client. Finally, you work through a series of advanced sections that shows how you can leverage and extend the IBM WebFacing Tool.

The tutorial is broken into chapters, each with their own specific learning objectives. You can choose to complete all chapters, or choose to complete only a few, depending on your learning goals. Each chapter is composed of several exercises. These exercises must be completed to achieve the chapter's goals.

“Verifying the latest fixes for the IBM WebFacing Tool” shows how to verify that you are working with the latest server, client tools, and environment for the IBM WebFacing Tool.

“Reviewing the 5250 order entry application” helps you learn about the 5250 order entry application before you start and how to use the IBM WebFacing Tool to convert it. This allows you to become familiar with the 5250 screens and the behavior of the application. After the conversion you can then easily compare the 5250 and the Web user interface.

“Creating the WebFacing project” explains that before you can use the IBM WebFacing Tool to reface your application, you need to create a WebFacing project. You also learn that this project is a complete Web project with the directory structure and the files needed to conform to the J2EE standard of a Web application. It also contains additional information unique to WebFacing like the source to be converted as well as runtime information.

“Converting selected source members” helps you learn about the conversion option, how to select one or multiple files for conversion, and how to work with the conversion logs to check the results of the WebFacing conversion.

“Running the WebFacing application” explains how to run the Web application in the WebSphere test environment. You also learn about the WebSphere test environment. You see the results of the conversion results and have the opportunity to try the converted interface as you did in “Reviewing the 5250 order entry application.”

“Changing the user interface” shows you how to change certain aspects of the user interface related to the conversion process.

“Changing the style of the Web user interface” explains how to create a new style, change style settings, specify the new style to be used for the project, and refresh the project with the new style.

“Command key rules and labels” explains how to add your own command key recognition rules to the WebFacing conversion properties. You also learn how to add command key labels that have missing labels in the DDS source.

“Working with more style properties” show you how to apply changes to windows, the look of push buttons, and subfiles.

“Adding authentication” explains how to secure your application by forcing authentication before the application is called.

“Enhancing index.jsp page” explains how to take the plain index.jsp page generated by the IBM WebFacing Tool and add color and some pictures to make the input page of the Web application more interesting. You learn how to use the Page Designer tool and some other related Web tools.

“Export/Import IBM WebFacing Tool projects in WebSphere Development Studio Client” is a step-by-step example on how to move your IBM WebFacing Tool into and out of WebSphere Development Studio Client for iSeries.

“Exporting to WebSphere Application Server Express for iSeries 6.0.1” describes how to export files created by the IBM WebFacing Tool to a remote WebSphere Application Server, to publish and deploy your application as you would when your application is ready for production.

“IBM WebFacing Tool deployment options” discusses the options for deploying, and IBM WebFacing Tool Application. It covers partial deployment, JSP precompile options, and configuring the pre-touch attributes.

“Creating a WebFacing portlet project” is a simple example of creating an IBM WebFacing Tool portlet project in WebSphere Development Studio Client. As part of this section, you use the WebSphere Portal test environment in WebSphere Development Studio Client for iSeries to run the IBM WebFacing Tool Portlet without having to install WebSphere Portal server on your System i model.

“Testing system screen support” is a simple example of creating an IBM WebFacing Tool project with system screen support. This section also discusses what system screens are officially supported.

“Invoking a Java Application from a WebFaced Web page” shows how to incorporate Java applications into your WebFacing application to extend and enhance the functionality of your IBM WebFacing Tool application.

“Invoking a System i program from a WebFaced Web page” explains how to incorporate other System i programs into your IBM WebFacing Tool application using the Web Interaction Wizard to extend and enhance the functionality of your IBM WebFacing Tool application.

“Adding a pop-up calendar” incorporates publicly available Java Script code to create a pop-up calendar for a date field. The pop-up calendar then takes the date selected and places the information into the date field.

“Advanced Web Settings example” is a series of advanced short examples that demonstrate the flexibility and power available in Web Settings.

“Configuring the WebFacing server” teaches you how to use the WebFacing server configuration Data Area to configure the WebFacing server for up to 16 interactive subsystems

Be sure to read the prerequisite knowledge for the tutorial before beginning. This section describes the type of knowledge you need to benefit from this tutorial.

Prerequisite knowledge

To complete this tutorial end-to-end, you must already have a working knowledge of the following:

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

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

- DDS
- Servlets and JavaServer Pages (to understand the generated output of the IBM WebFacing Tool)

If you do not have current knowledge of these technologies and concepts, you might be interested in learning more about these after you complete this tutorial. You can find additional information on these topics in the Help system that is part of WebSphere Development Studio Client for iSeries. There are also plenty of resources on the Internet at our product Web page: **ibm**.com/software/awdtools/wdt400.

In the next chapter, you use a 5250 emulation screen to start the sample Order Entry application. In this application, you are prompted with the first screen for a customer number. With the use of help panels, you select a customer from a list of valid customers, and then go through the steps of filling an order by specifying the quantity ordered. After all parts and quantities for this order have been decided, you complete the order by pressing a specific command key. The application then prompts for a new customer number to handle the next order.

When you are ready to begin, start with “Reviewing the 5250 order entry application.”

Verifying the latest fixes for the IBM WebFacing Tool

When starting a WebFacing project, it is important to have the latest IBM WebFacing Tool fixes for both the client and the server. This example shows you how to ensure you are working with the latest version of the tool. If you do not have Internet connectivity, you can review the steps but are unable to complete the steps yourself. To accomplish these learning objectives, a few steps are involved. They are:

- Exercise 0.1: Verify the WebSphere Development Studio Client for iSeries version (IBM WebFacing Tool Client)
- Exercise 0.2: Verify IBM WebFacing Tool client fixes
- Exercise 0.3: Verify IBM WebFacing Tool server PTFs

Length of time

This chapter takes approximately 10 minutes to complete.

Exercise 0.1 Verify the WebSphere Development Studio Client for iSeries version

To determine your WebSphere Development Studio Client version, you must:

1. Start WebSphere Development Studio Client for iSeries. This can be accomplished by clicking: **Start > All Programs > IBM Rational > IBM WebSphere Development Studio Client Advanced Edition for iSeries V6.0 > WebSphere Development Studio Client Advanced Edition for iSeries** (see Figure 1)**.**

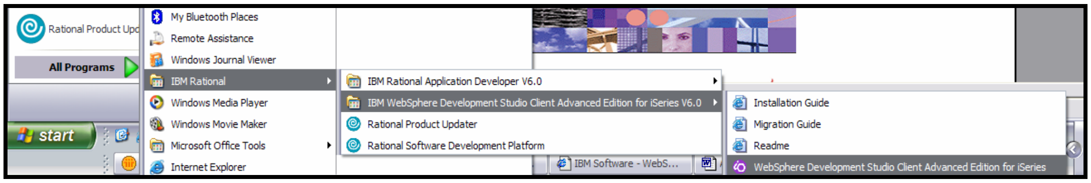
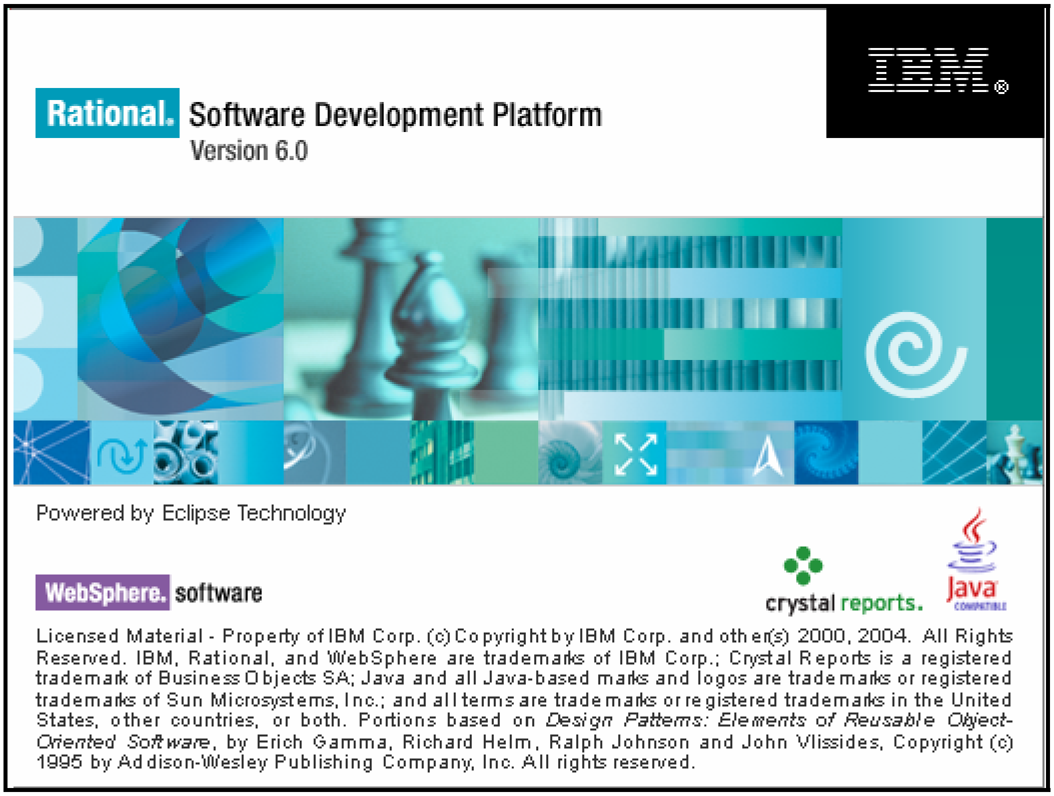


Figure 1. Determining the version of your WebSphere Development Studio Client

2. As WebSphere Development Studio Client for iSeries starts, its splash screen tells you what version of the tool you are running (see Figure 2).


IBM
Rational. Software Development Platform
Version 6.0
Powered by Eclipse Technology
WebSphere. software
crystal reports.
Java COMPATIBLE

Licensed Material - Property of IBM Corp. (c) Copyright by IBM Corp. and other(s) 2000, 2004. All Rights Reserved. IBM, Rational, and WebSphere are trademarks of IBM Corp.; Crystal Reports is a registered trademark of Business Objects SA; Java and all Java-based marks and logos are trademarks or registered trademarks of Sun Microsystems, Inc.; and all terms are trademarks or registered trademarks in the United States, other countries, or both. Portions based on *Design Patterns: Elements of Reusable Object-Oriented Software*, by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Copyright (c) 1995 by Addison-Wesley Publishing Company, Inc. All rights reserved.

Figure 2. The version provided for the tool you are running

Exercise 0.2 Verify IBM WebFacing Tool client fixes

In this step, you update WebSphere Development Studio Client with the latest fixes available for the version of the tool that you are running.

Run software updates

Run Software Updates in WebSphere Development Studio Client:

6. Ensure WebSphere Development Studio Client is running. WebSphere Development Studio Client runs after you complete the previous step. If it is not running, start it.
7. Ensure that there is a perspective open in WebSphere Development Studio Client. For Software Updates to run in WebSphere Development Studio Client, a perspective needs to be open. By default, the Remote Systems Explorer perspective is open. If there is not a perspective available, open one by clicking and selecting the Remote Systems Explorer perspective.
8. Run New Updates by clicking:
Help > Software Updates > IBM Rational Product Updater (see Figure 3).

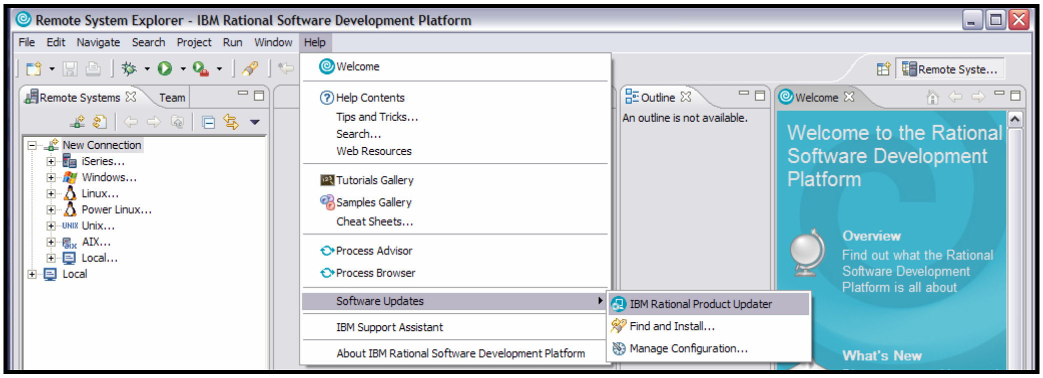
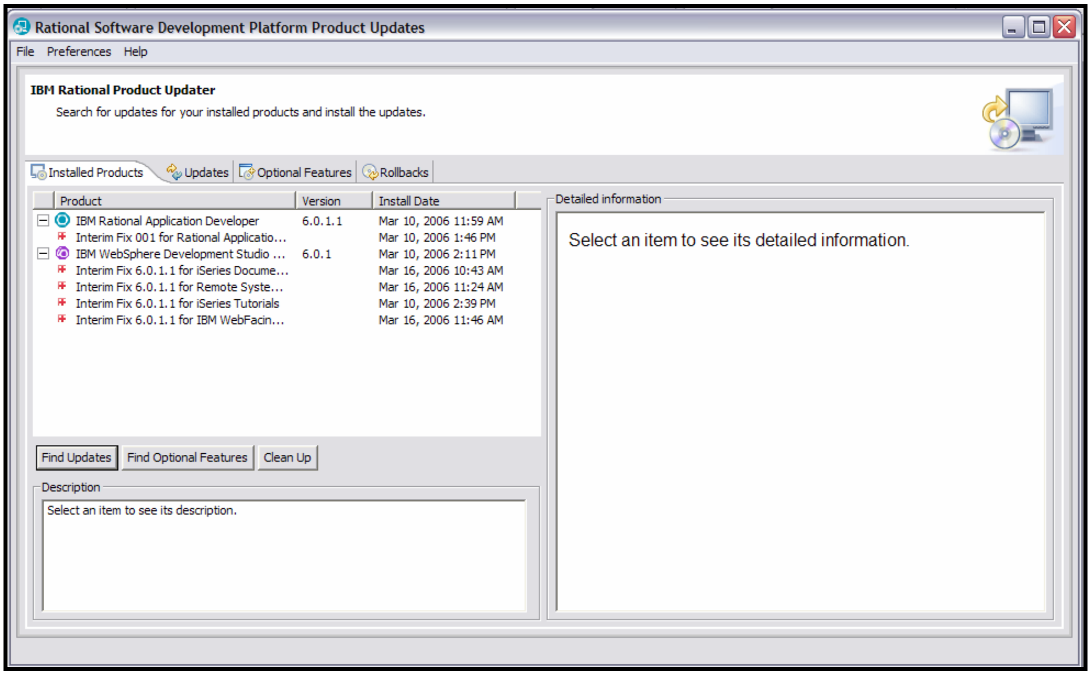


Figure 3. Run new updates

The Rational Software Development Platform Product Updates panel is displayed (see Figure 4).



IBM Rational Product Updater

Search for updates for your installed products and install the updates.

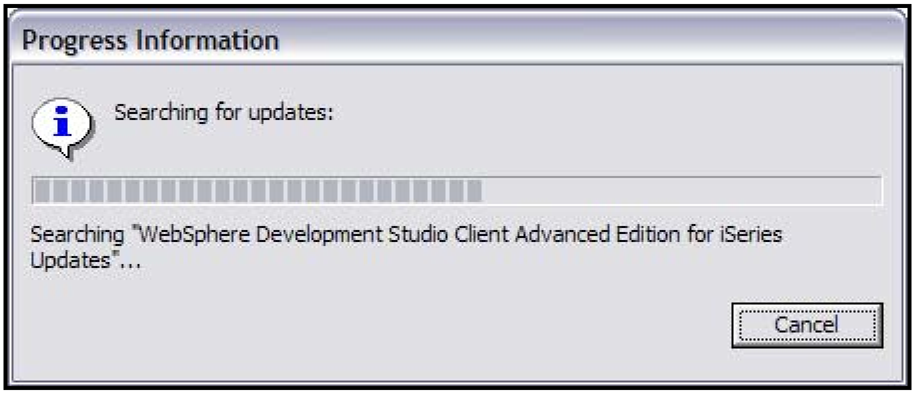
Product	Version	Install Date
IBM Rational Application Developer	6.0.1.1	Mar 10, 2006 11:59 AM
Interim Fix 001 for Rational Applicatio…		Mar 10, 2006 1:46 PM
IBM WebSphere Development Studio …	6.0.1	Mar 10, 2006 2:11 PM
Interim Fix 6.0.1.1 for iSeries Docume…		Mar 16, 2006 10:43 AM
Interim Fix 6.0.1.1 for Remote Syste…		Mar 16, 2006 11:24 AM
Interim Fix 6.0.1.1 for iSeries Tutorials		Mar 10, 2006 2:39 PM
Interim Fix 6.0.1.1 for IBM WebFacin…		Mar 16, 2006 11:46 AM

Select an item to see its description.

Select an item to see its detailed information.

Figure 4. Rational Software Development Platform product updates panel displayed

9. Click the **Find Updates** button (see Figure 5). This starts a search for updates.

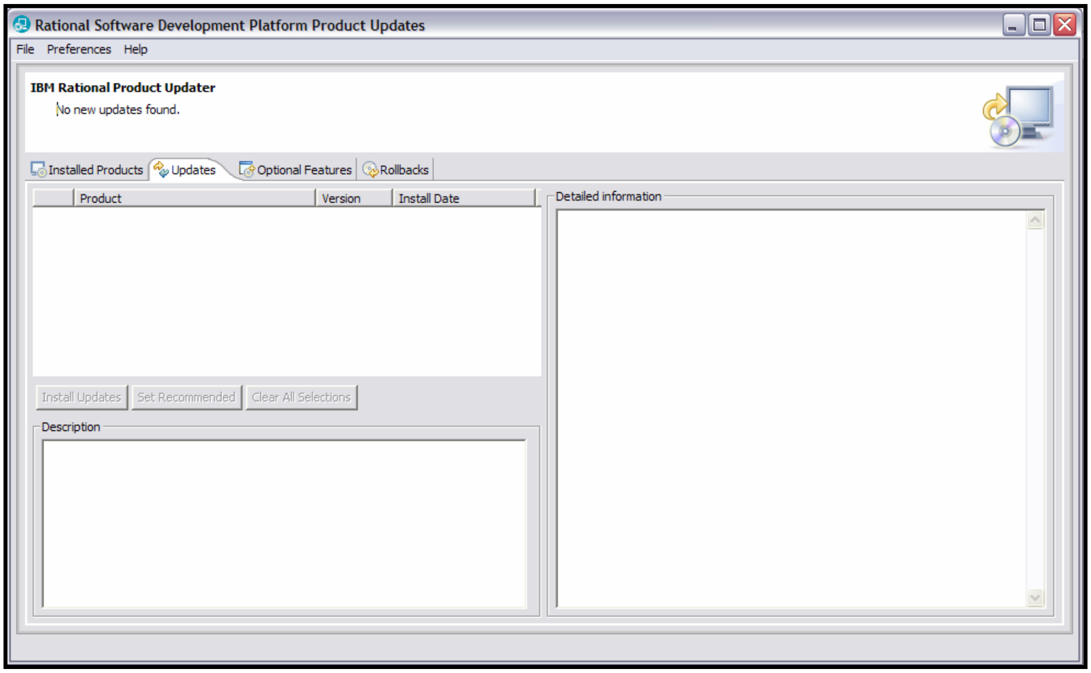


Searching for updates:

Searching "WebSphere Development Studio Client Advanced Edition for iSeries Updates"…

Figure 5. Searching for updates

The Rational Software Development Platform Product Updates panel is then displayed, and any available updates are listed. If updates are available, select the **Install Updates** button (see Figure 6). This installs the updates.



No new updates found.

Figure 6. Rational Software Development Platform updates panel

Exercise 0.3 Verify IBM WebSphere Development Studio Client, WebFacing Tool and CODE server PTFs

Making sure that you have the latest PTFs installed on your System i model is very important to avoid encountering errors in your IBM WebFacing Tool application. It is simple and fast to check IBM WebFacing Tool and CODE PTFs on your server.

Verify PTFs

Verify that you have the necessary PTFs on your System i model:

1. Switch to the Remote Systems Explorer perspective. To open a WebFacing perspective, click **Window > Open Perspective > Remote System Explorer** from the workbench menu (see Figure 7).

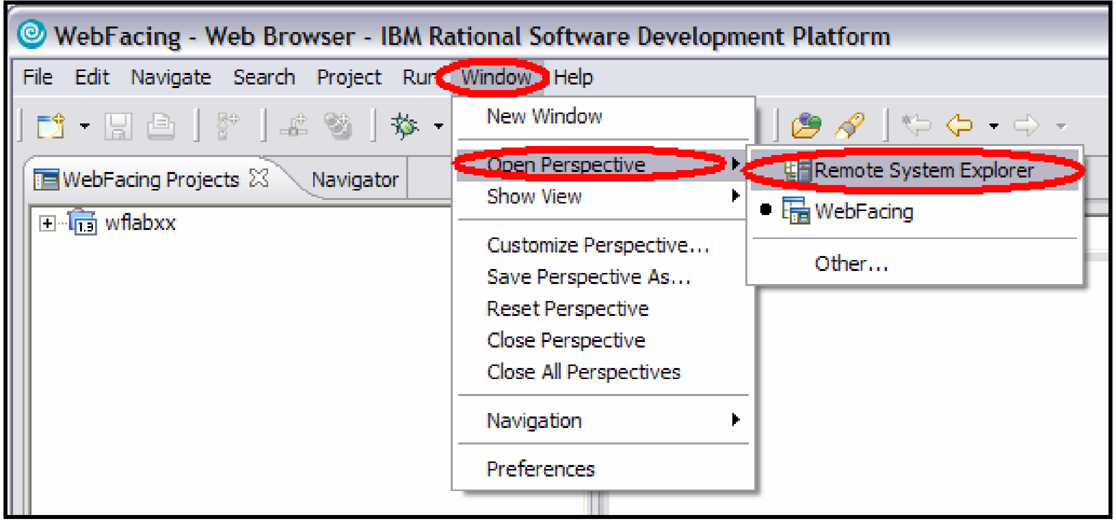
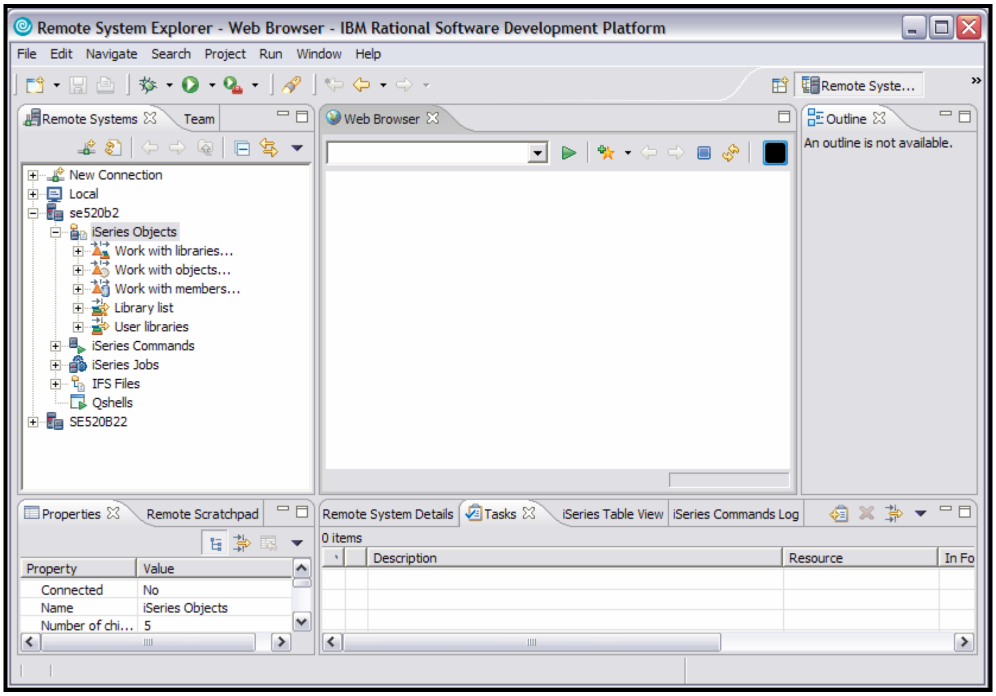


Figure 7. Open WebFacing perspective

2. The workbench now displays the Remote Systems Explorer perspective with the WebFacing Projects view open in the left pane of the workbench (see Figure 8).



An outline is not available.

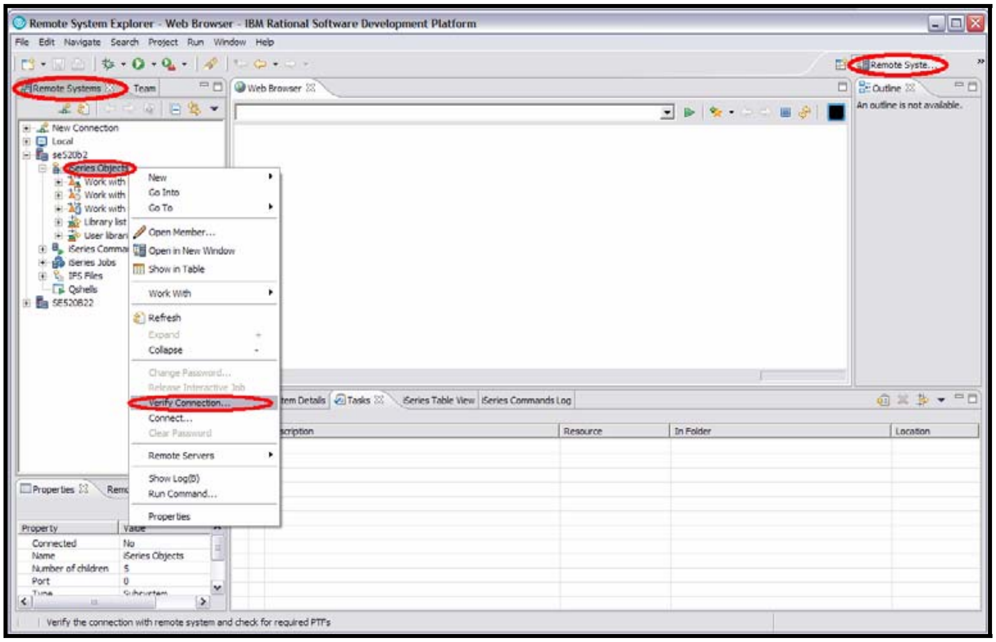
Property	Value
Connected	No
Name	iSeries Objects
Number of chi...	5

0 items

Description	Resource	In Fo

Figure 8. The Remote Systems Explorer perspective with the WebFacing Projects view

3. Expand your server information in the left-hand pane. Right-click **iSeries Objects**, and then select **Verify Connection... (see Figure 9).**



An outline is not available.

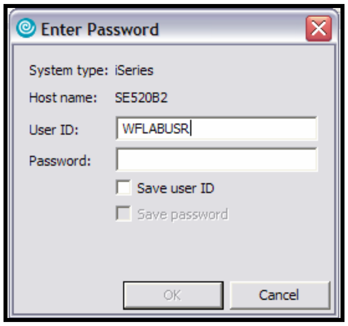
Property	Value
Connected	No
Name	iSeries Objects
Number of children	5
Port	0
Type	Subsystem

scription	Resource	In Folder	Location

Verify the connection with remote system and check for required PTFs

Figure 9. Expand your server information in the left-hand pane

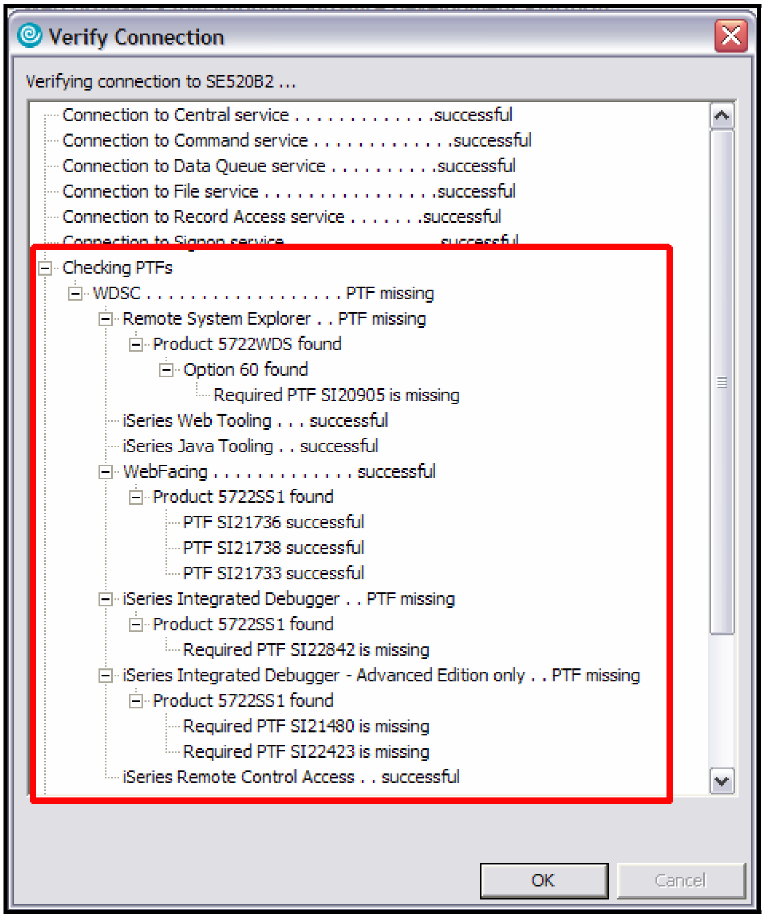
4. If prompted, enter your user ID and password and click **OK** (see Figure 10).



System type: iSeries
Host name: SE520B2
User ID: WFLABUSR
Password:
□ Save user ID
□ Save password

*Figure 10. Enter user ID and password, and click **OK**.*

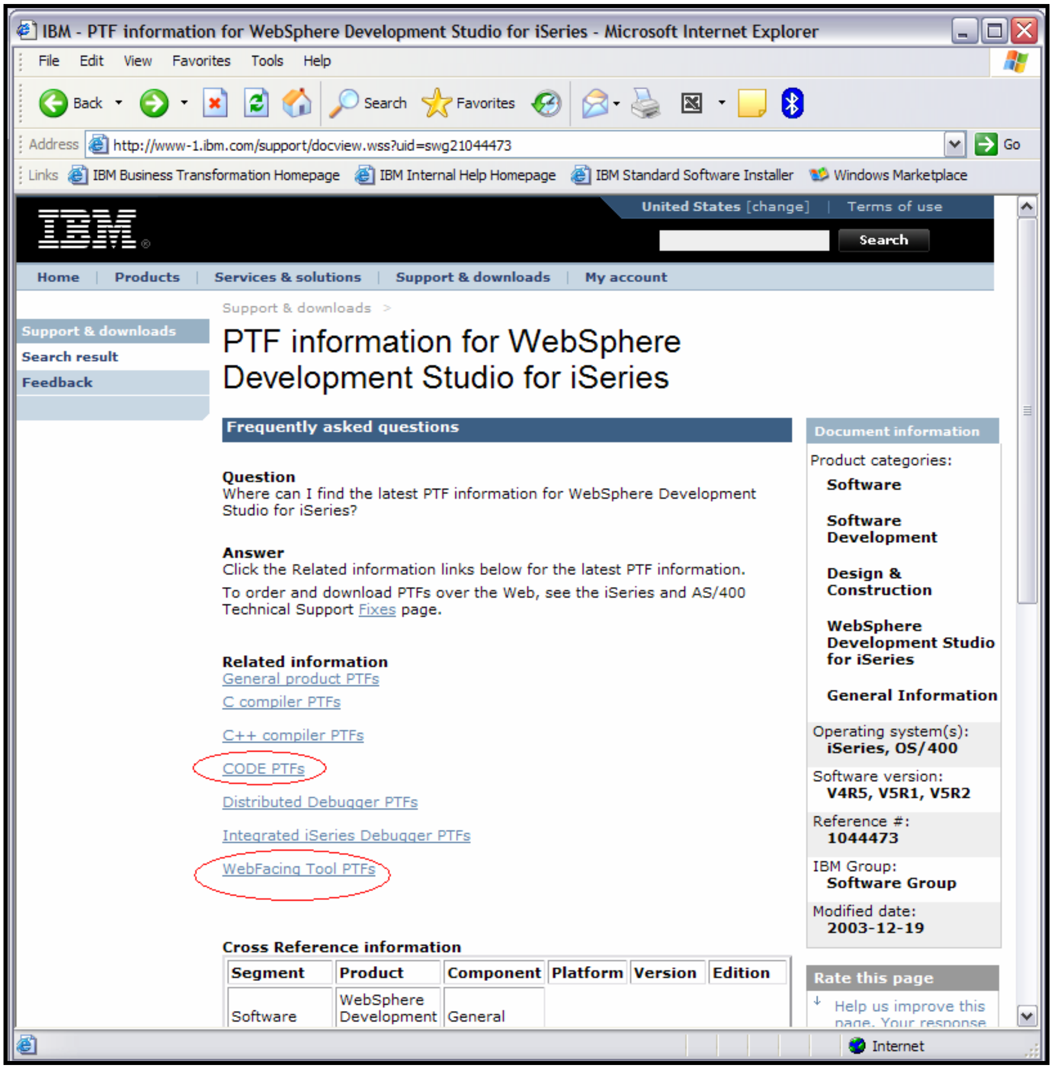
5. The Verify Connections panel then appears. As part of the verification process, the host server is scanned to determine if all required WebSphere Development Studio Client PTFs are installed. The results are then displayed in the Verify Connections panel. Make note of any missing PTFs that need to be installed, and click **OK** (see Figure 11).



```
Verifying connection to SE520B2 ...
Connection to Central service . . . . . . . . . . . . .successful
Connection to Command service . . . . . . . . . . . . .successful
Connection to Data Queue service . . . . . . . . . .successful
Connection to File service . . . . . . . . . . . . . . . .successful
Connection to Record Access service . . . . . . .successful
Connection to Signon service                     successful
Checking PTFs
  WDSC . . . . . . . . . . . . . . . . . . PTF missing
    Remote System Explorer . . PTF missing
      Product 5722WDS found
        Option 60 found
          Required PTF SI20905 is missing
    iSeries Web Tooling . . . successful
    iSeries Java Tooling . . successful
    WebFacing . . . . . . . . . . . . . successful
      Product 5722SS1 found
        PTF SI21736 successful
        PTF SI21738 successful
        PTF SI21733 successful
    iSeries Integrated Debugger . . PTF missing
      Product 5722SS1 found
        Required PTF SI22842 is missing
    iSeries Integrated Debugger - Advanced Edition only . . PTF missing
      Product 5722SS1 found
        Required PTF SI21480 is missing
        Required PTF SI22423 is missing
    iSeries Remote Control Access . . successful
```

Figure 11. The Verify Connections panel

6. Next, you need to check the IBM Support Web sites for any additional PTFs that might be needed. The “PTF information for WebSphere Development Studio for iSeries” support site can be found at: **ibm.com**/support/docview.wss?uid=swg21044473. When there, click the **CODE PTFs** list item in the main panel (see Figure 12).


IBM - PTF information for WebSphere Development Studio for iSeries - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
http://www-1.ibm.com/support/docview.wss?uid=swg21044473
Go
Links
IBM Business Transformation Homepage
IBM Internal Help Homepage
IBM Standard Software Installer
Windows Marketplace
IBM
United States [change]
Terms of use
Search
Home
Products
Services & solutions
Support & downloads
My account
Support & downloads
Search result
Feedback
Internet

Support & downloads >

PTF information for WebSphere Development Studio for iSeries

Frequently asked questions

Question
Where can I find the latest PTF information for WebSphere Development Studio for iSeries?

Answer
Click the Related information links below for the latest PTF information.

To order and download PTFs over the Web, see the iSeries and AS/400 Technical Support Fixes page.

Related information
General product PTFs
C compiler PTFs
C++ compiler PTFs
CODE PTFs
Distributed Debugger PTFs
Integrated iSeries Debugger PTFs
WebFacing Tool PTFs

Cross Reference information

Segment	Product	Component	Platform	Version	Edition
Software	WebSphere Development	General			

Document information

Product categories:
Software
Software Development
Design & Construction
WebSphere Development Studio for iSeries
General Information

Operating system(s):
iSeries, OS/400

Software version:
V4R5, V5R1, V5R2

Reference #:
1044473

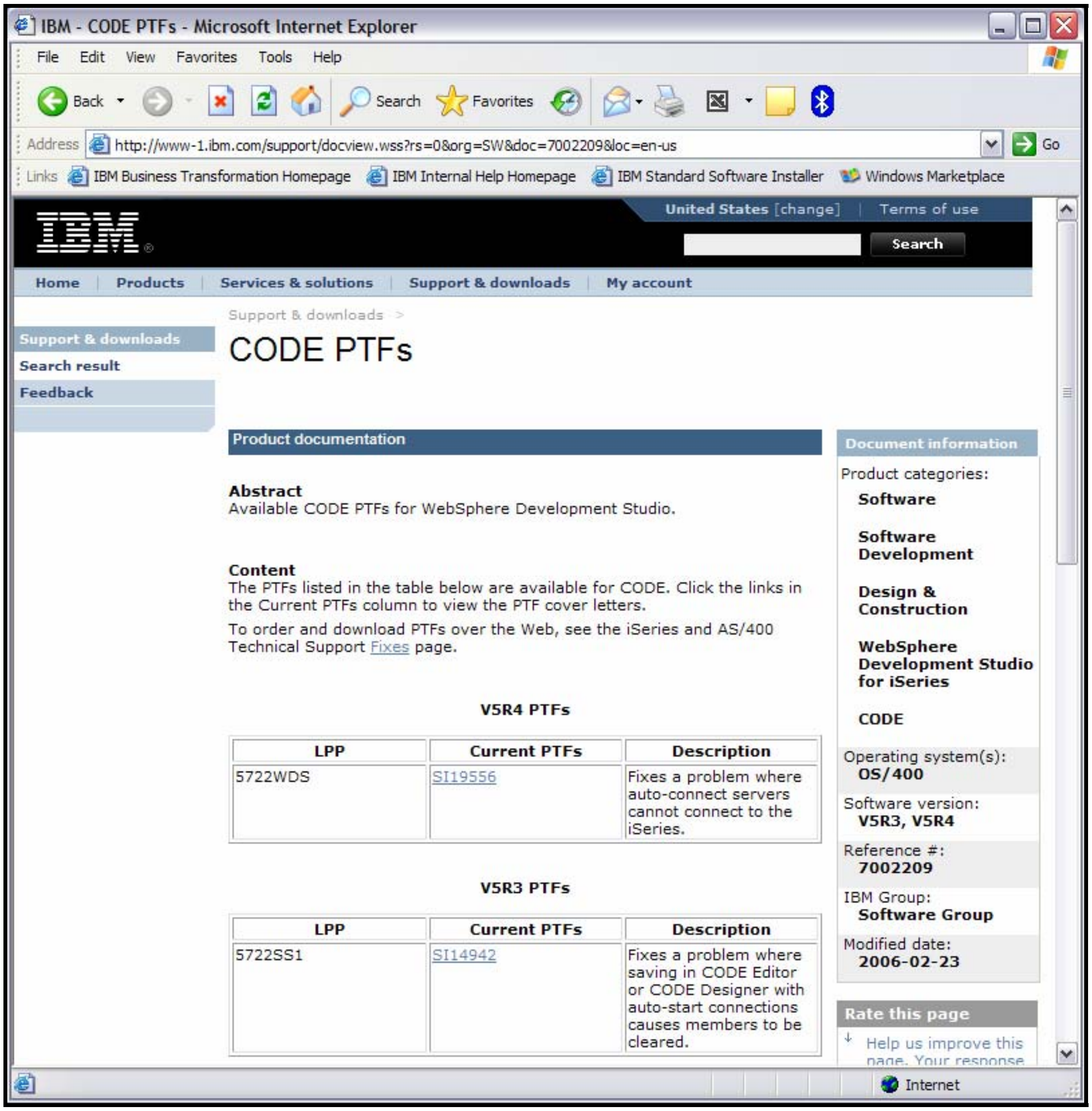
IBM Group:
Software Group

Modified date:
2003-12-19

Rate this page
Help us improve this page. Your response

*Figure 12. Click the **CODE PTFs** list item in the main panel.*

The subsequent panel then displays CODE PTFs categorized based on operating system version and release (see Figure 13).


IBM - CODE PTFs - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://www-1.ibm.com/support/docview.wss?rs=0&org=SW&doc=7002209&loc=en-us
Go
Links
IBM Business Transformation Homepage
IBM Internal Help Homepage
IBM Standard Software Installer
Windows Marketplace
United States [change]
Terms of use
Search
Home
Products
Services & solutions
Support & downloads
My account
Support & downloads
Search result
Feedback
Internet

Support & downloads >

CODE PTFs

Product documentation

Abstract
Available CODE PTFs for WebSphere Development Studio.

Content
The PTFs listed in the table below are available for CODE. Click the links in the Current PTFs column to view the PTF cover letters.

To order and download PTFs over the Web, see the iSeries and AS/400 Technical Support Fixes page.

V5R4 PTFs

LPP	Current PTFs	Description
5722WDS	SI19556	Fixes a problem where auto-connect servers cannot connect to the iSeries.

V5R3 PTFs

LPP	Current PTFs	Description
5722SS1	SI14942	Fixes a problem where saving in CODE Editor or CODE Designer with auto-start connections causes members to be cleared.

Document information

Product categories:
Software
Software Development
Design & Construction
WebSphere Development Studio for iSeries
CODE

Operating system(s):
OS/400

Software version:
V5R3, V5R4

Reference #:
7002209

IBM Group:
Software Group

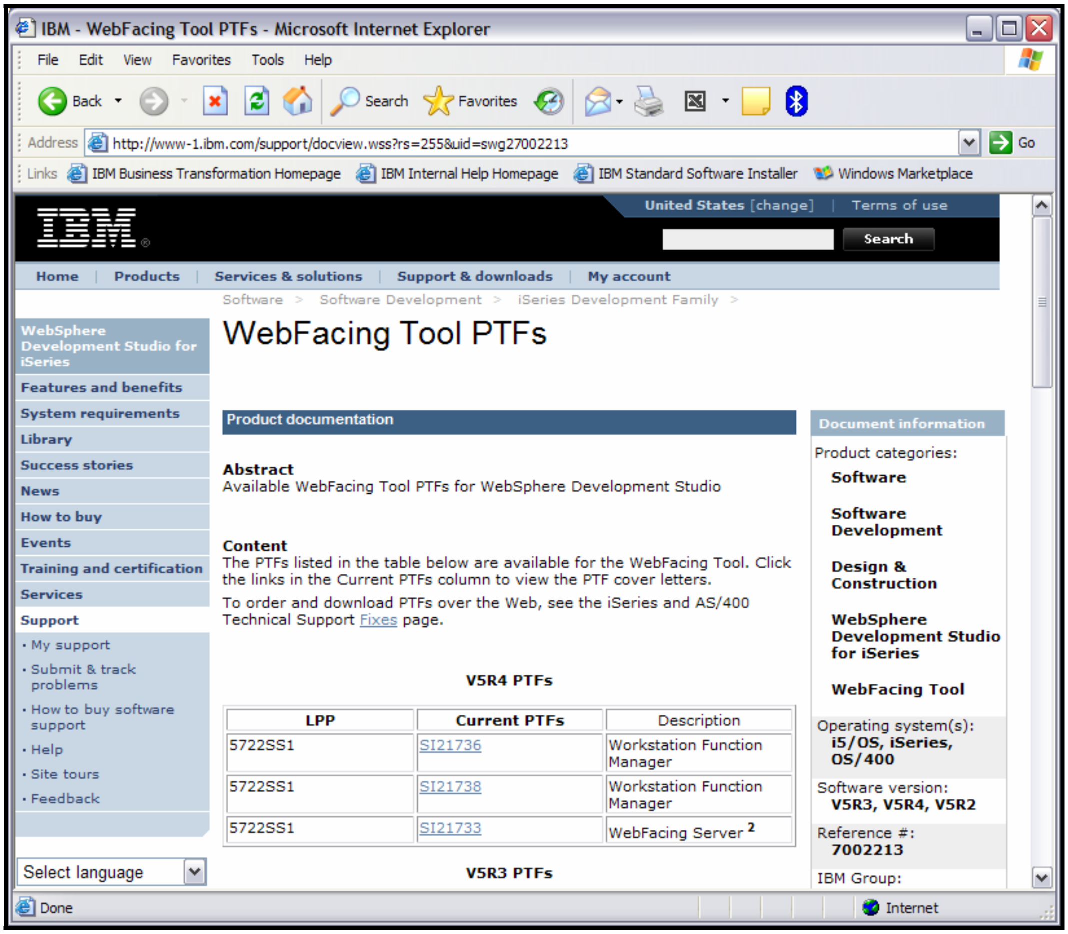
Modified date:
2006-02-23

Rate this page

Help us improve this page. Your response

Figure 13. Display of ***CODE PTFs***

7. Click the **Back** button, and select the WebFacing Tool PTFs list item in the main panel. The subsequent panel then displays WebFacing Tool PTFs categorized based on operating system version and release (see Figure 14).


IBM - WebFacing Tool PTFs - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www-1.ibm.com/support/docview.wss?rs=255&uid=swg27002213 Go
Links IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer Windows Marketplace
IBM
United States [change] | Terms of use
Search
Home | Products | Services & solutions | Support & downloads | My account
Software > Software Development > iSeries Development Family >
WebSphere Development Studio for iSeries
Features and benefits
System requirements
Library
Success stories
News
How to buy
Events
Training and certification
Services
Support
· My support
· Submit & track problems
· How to buy software support
· Help
· Site tours
· Feedback
Select language
Done
Internet

WebFacing Tool PTFs

Product documentation

Abstract
Available WebFacing Tool PTFs for WebSphere Development Studio

Content
The PTFs listed in the table below are available for the WebFacing Tool. Click the links in the Current PTFs column to view the PTF cover letters.

To order and download PTFs over the Web, see the iSeries and AS/400 Technical Support Fixes page.

V5R4 PTFs

LPP	Current PTFs	Description
5722SS1	SI21736	Workstation Function Manager
5722SS1	SI21738	Workstation Function Manager
5722SS1	SI21733	WebFacing Server [2]

V5R3 PTFs

Document information

Product categories:
Software

Software Development

Design & Construction

WebSphere Development Studio for iSeries

WebFacing Tool

Operating system(s):
i5/OS, iSeries, OS/400

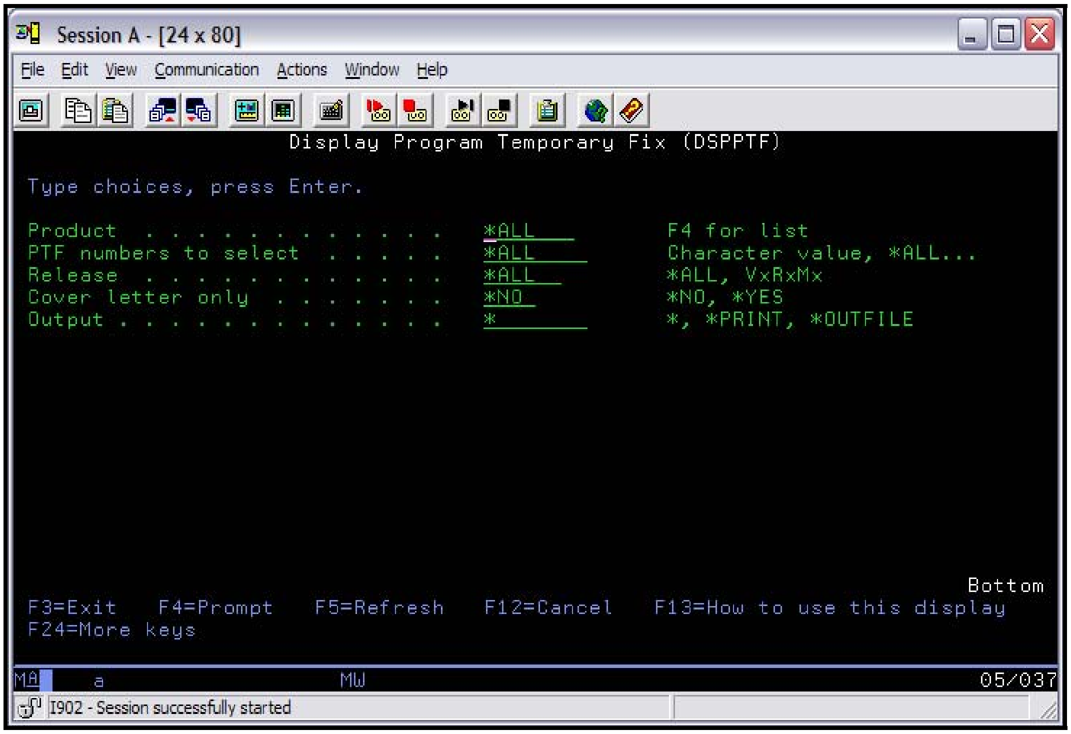
Software version:
V5R3, V5R4, V5R2

Reference #:
7002213

IBM Group:

Figure 14. Select the WebFacing Tool PTFs list item.

8. When you have the most current list of PTFs, you can check your System i model to see if they are installed. This can be done using the **DSPPTF** (Display Program Temporary Fix) command (see Figure 15).



```
                         Display Program Temporary Fix (DSPPTF)

 Type choices, press Enter.

 Product  . . . . . . . . . . . .   *ALL          F4 for list
 PTF numbers to select  . . . . .   *ALL          Character value, *ALL...
 Release  . . . . . . . . . . . .   *ALL          *ALL, VxRxMx
 Cover letter only  . . . . . . .   *NO           *NO, *YES
 Output . . . . . . . . . . . . .   *             *, *PRINT, *OUTFILE

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
MA     a                      MW                                          05/037
I902 - Session successfully started
```

*Figure 15. Use the **DSPPTF** command to check for installation of PTFs on System i model*

Recap

You have completed the “Verifying the latest fixes for the IBM WebFacing Tool” exercise. You now have the information to understand how to:

- Determine your version of WebSphere Development Studio Client for iSeries (IBM WebFacing Tool Client)
- Verify IBM WebFacing Tool client fixes
- Verify IBM WebFacing Tool server PTFs

Ensuring you are working with the most up-to-date version of the tools is very important to use the latest functions and to avoid problems. As you have seen in this section, it is easy and fast, so there is no excuse to be working without the latest fixes.

Reviewing the 5250 order entry application

In this chapter, you are going to explore the application by invoking it from a 5250 session and working with the applications green-screens and command keys. This also tests your user profile setup. If you can run this application directly without changing the job environment on the System i model, the WebFacing conversion and the WebFacing server does not have problems during the remaining exercises later.

Note: If you have problems, make sure the library list for your job is set up correctly to include library WFLABXX at job start up.

You learn about the 5250 application before you start to use the IBM WebFacing Tool to convert it. This allows you to become familiar with the 5250 screens and the behavior of the application. After the conversion with the IBM WebFacing Tool, you can compare the 5250 and the Web user interface.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 1.1: Starting the 5250 application
- Exercise 1.2: Accessing the online help
- Exercise 1.3: Selecting a customer
- Exercise 1.4: Filling the order
- Exercise 1.5: Completing the order

The exercises in this chapter must be completed in order. Start with “Exercise 1.1: Starting the 5250 application” when you are ready to begin.

Length of time

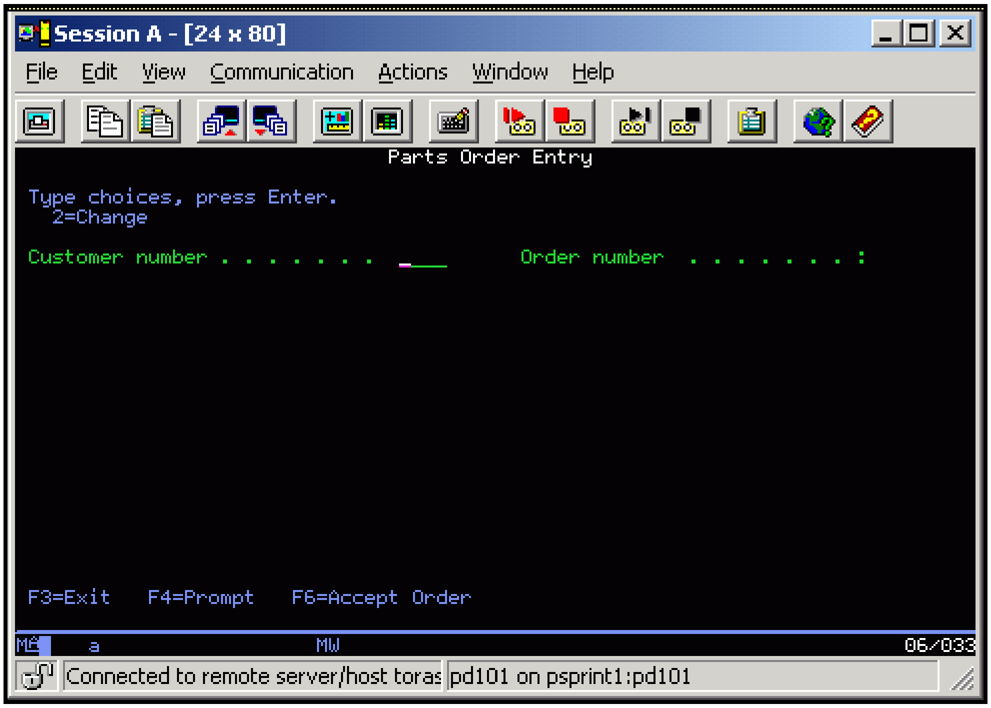
This chapter takes approximately five minutes to complete.

Exercise 1.1: Starting the 5250 application

You need a 5250 emulator on your workstation to start the order entry application. To start the 5250 application:

1. Start a 5250 emulation session.
2. In the **User ID** field, type your User ID.
3. In the **Password** field, type your password.
4. On the command line of the 5250 screen, invoke the **Order Entry Application: CALL ORDENTR**.

   The application starts and shows the first panel. This first screen prompts for a customer number. You have a choice of keying in the customer number directly, or you can press command key **F4** to display a list of existing customers (see Figure 16).


Session A - [24 x 80]
File
Edit
View
Communication
Actions
Window
Help

```
                              Parts Order Entry

 Type choices, press Enter.
   2=Change

 Customer number . . . . . . . _____       Order number  . . . . . . . :




















 F3=Exit    F4=Prompt    F6=Accept Order
```

MA   a   MW   06/033
Connected to remote server/host toras pd101 on psprint1:pd101

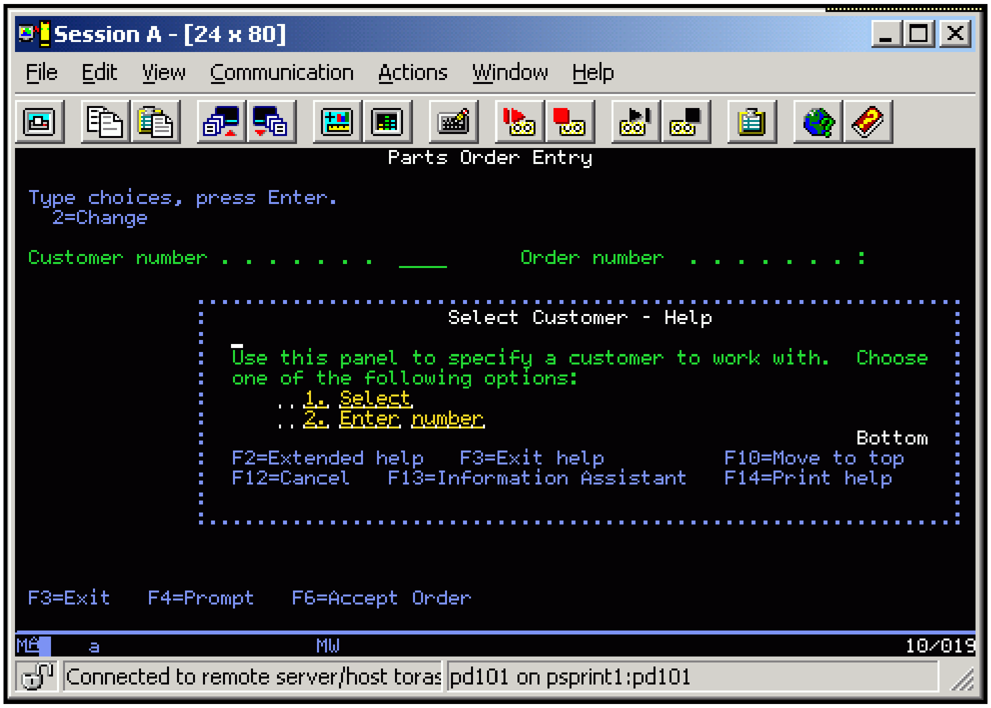
Figure 16. Parts Order Entry screen

Exercise 1.2: Accessing the online help

Before you begin the order entry process, try the help option. This application uses User Interface Manager (UIM) help. You also convert these UIM help panels with the IBM WebFacing Tool. To understand the results of the UIM help conversion, try the help application in the 5250 environment and then later in the browser.

1. Move the cursor to the **Customer number** field.
2. Press command key **F1**.

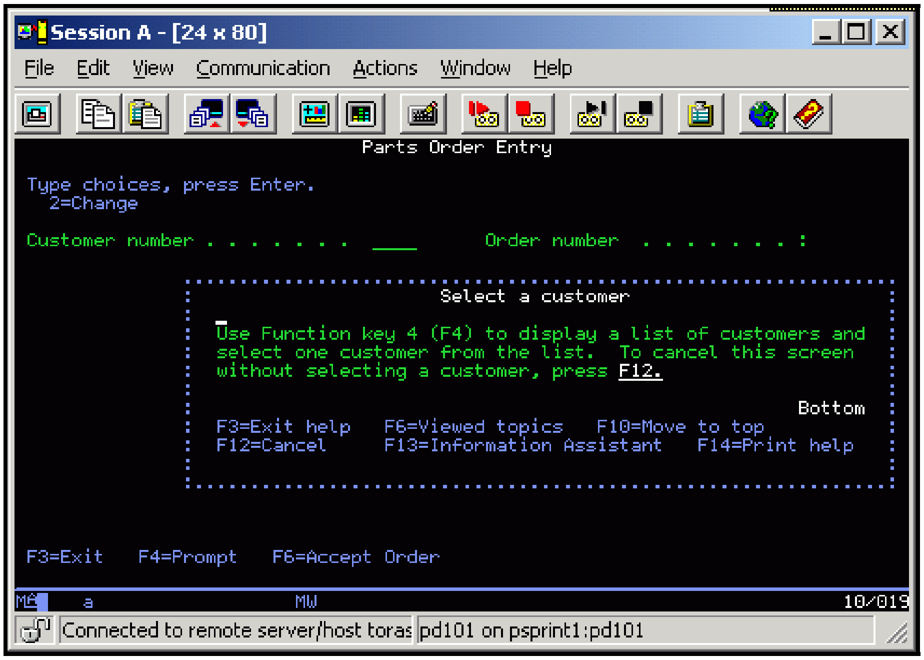
   The panel group help displays for the customer number (see Figure 17).



```
                             Parts Order Entry

 Type choices, press Enter.
   2=Change

 Customer number . . . . . . .  ____      Order number  . . . . . . . :

                   ................................................................
                   :                    Select Customer - Help                    :
                   :                                                              :
                   :  Use this panel to specify a customer to work with.  Choose  :
                   :  one of the following options:                               :
                   :      . . 1. Select                                           :
                   :      . . 2. Enter number                                     :
                   :                                                       Bottom :
                   :  F2=Extended help   F3=Exit help          F10=Move to top    :
                   :  F12=Cancel    F13=Information Assistant  F14=Print help     :
                   :                                                              :
                   ................................................................



 F3=Exit    F4=Prompt    F6=Accept Order
```

Figure 17. The panel group help displays for the customer number.

3. To see more help, move the cursor underneath the **1** in the help area and press the **Enter** key. Help for requesting a customer list displays as seen in Figure 18.



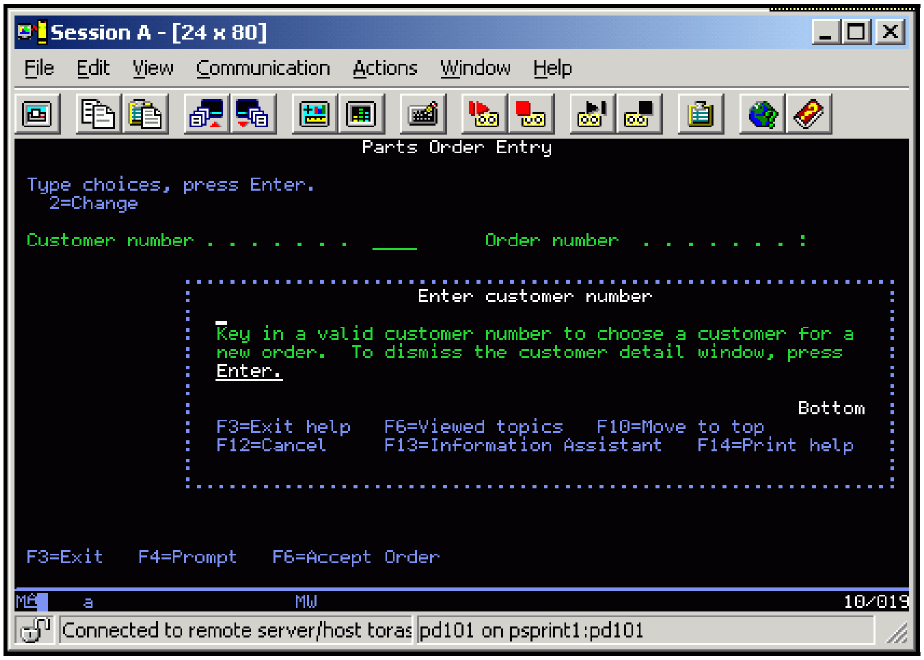
```
                              Parts Order Entry

 Type choices, press Enter.
   2=Change

 Customer number . . . . . . .  ____      Order number  . . . . . . . :

                .......................................................
                :                  Select a customer                  :
                :                                                     :
                :  Use Function key 4 (F4) to display a list of customers and
                :  select one customer from the list.  To cancel this screen
                :  without selecting a customer, press F12.           :
                :                                                     :
                :                                             Bottom  :
                :  F3=Exit help   F6=Viewed topics   F10=Move to top  :
                :  F12=Cancel     F13=Information Assistant   F14=Print help
                :                                                     :
                .......................................................


 F3=Exit   F4=Prompt   F6=Accept Order
```

MA a MW 10/019
Connected to remote server/host toras pd101 on psprint1:pd101

Figure 18. Help for requesting a customer list

4. Press command key **F12** to return to the help for the **Customer number** field.

5. To see help for the second choice, move the cursor underneath the **2** in the help area and press the **Enter** key to see detailed help for specifying a valid customer number (see Figure 19).


Session A - [24 x 80]
File Edit View Communication Actions Window Help

```
                               Parts Order Entry

 Type choices, press Enter.
   2=Change

 Customer number . . . . . . .  ____       Order number  . . . . . . . :

                        ...............................................
                        :              Enter customer number          :
                        :                                             :
                        :  Key in a valid customer number to choose a customer for a :
                        :  new order.  To dismiss the customer detail window, press  :
                        :  Enter.                                     :
                        :                                             :
                        :                                      Bottom :
                        :  F3=Exit help   F6=Viewed topics   F10=Move to top :
                        :  F12=Cancel     F13=Information Assistant   F14=Print help :
                        :                                             :
                        ...............................................

 F3=Exit   F4=Prompt   F6=Accept Order
```

MA a MW 10/019
Connected to remote server/host toras pd101 on psprint1:pd101

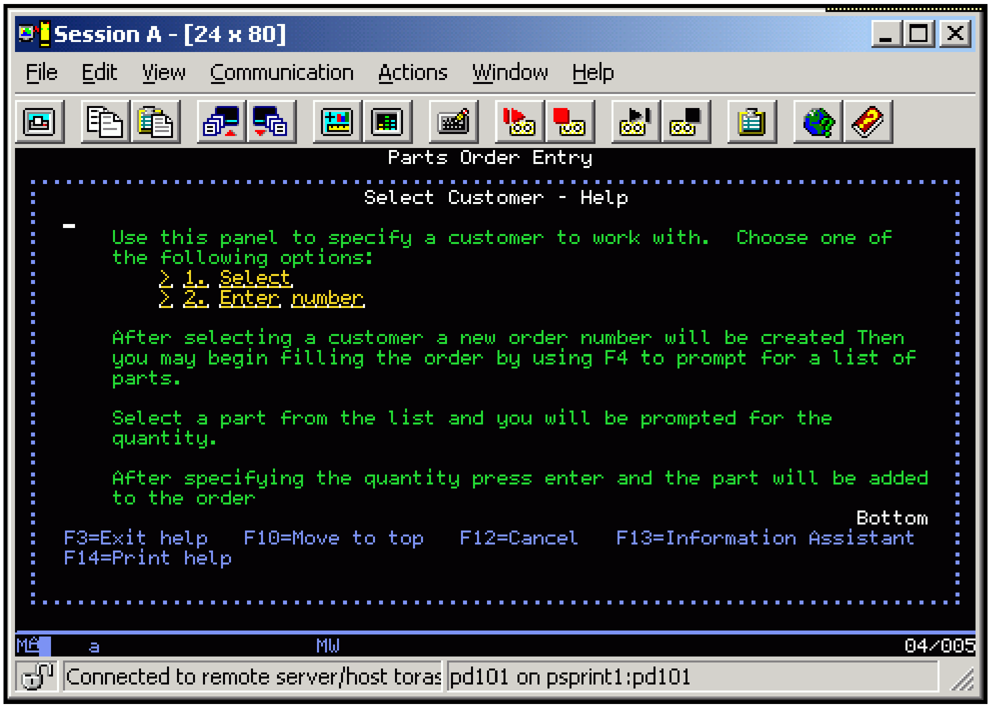
Figure 19. Detailed help for specifying a valid customer number

6. Press command key **F12** to leave the help for the **Customer number** field.

The main help panel in the Select customer - Help area shows.

7. Press command key **F2** to get to the extended help.

   Here you see the extended help for the **Customer number** field (see Figure 20).



```
                           Parts Order Entry
 :                       Select Customer - Help                            :
 :                                                                         :
 :  -    Use this panel to specify a customer to work with.  Choose one of :
 :       the following options:                                            :
 :          > 1. Select                                                    :
 :          > 2. Enter number                                              :
 :                                                                         :
 :       After selecting a customer a new order number will be created Then:
 :       you may begin filling the order by using F4 to prompt for a list of
 :       parts.                                                            :
 :                                                                         :
 :       Select a part from the list and you will be prompted for the      :
 :       quantity.                                                         :
 :                                                                         :
 :       After specifying the quantity press enter and the part will be added
 :       to the order                                                      :
 :                                                                 Bottom  :
 :  F3=Exit help   F10=Move to top   F12=Cancel   F13=Information Assistant:
 :  F14=Print help                                                         :
 :                                                                         :
MA   a                  MW                                          04/005
```

Connected to remote server/host toras | pd101 on psprint1:pd101

*Figure 20. Extended help for the **Customer number** field.*

8. Press command key **F12** to leave the extended help for the **Customer number** field.

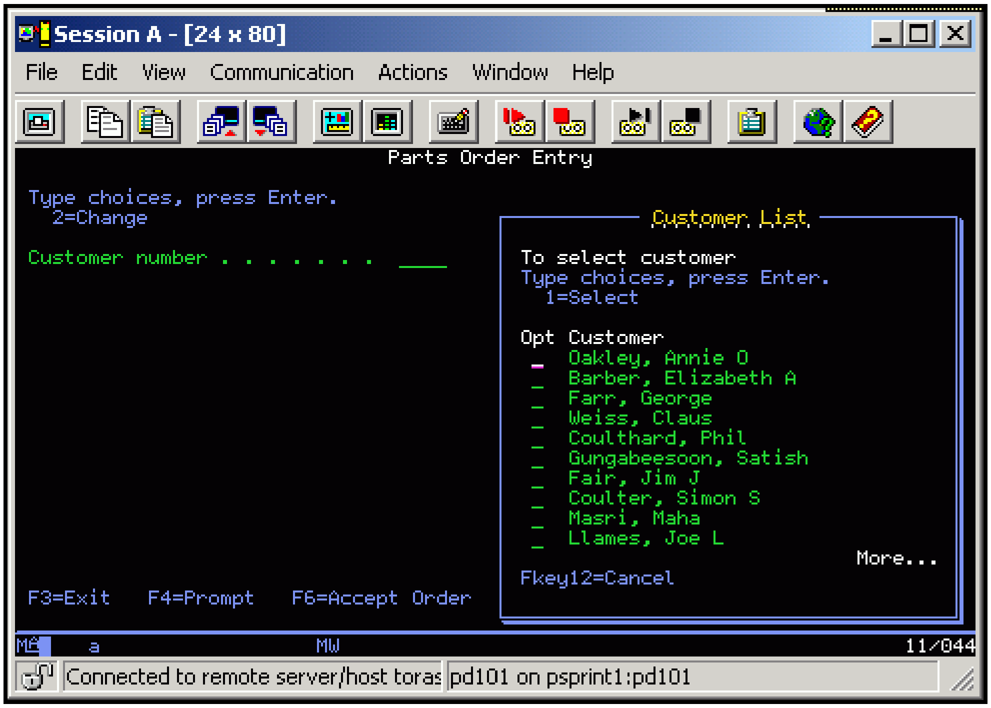
9. Press command key **F12** again to leave the help for the **Customer number** field.

Exercise 1.3: Selecting a customer

Now you have an understanding of how the help panels in this application work and look. Next, you need to identify the customer related order. You have the choice of specifying a valid customer number, or you can get a selection list of valid customers. Use the selection list because it shows a record format that uses the Window DDS keyword so you can see how the IBM WebFacing Tool converts a DDS window.

1. Press command key **F4** to display the customer selection list.

   A list of customers displays (see Figure 21).


Session A - [24 x 80]
File Edit View Communication Actions Window Help

```
                              Parts Order Entry

 Type choices, press Enter.
   2=Change                                   Customer List
                                         To select customer
 Customer number . . . . . . .  ____     Type choices, press Enter.
                                           1=Select

                                         Opt  Customer
                                          _   Oakley, Annie O
                                          _   Barber, Elizabeth A
                                          _   Farr, George
                                          _   Weiss, Claus
                                          _   Coulthard, Phil
                                          _   Gungabeesoon, Satish
                                          _   Fair, Jim J
                                          _   Coulter, Simon S
                                          _   Masri, Maha
                                          _   Llames, Joe L
                                                               More...
                                         Fkey12=Cancel
 F3=Exit    F4=Prompt    F6=Accept Order
```

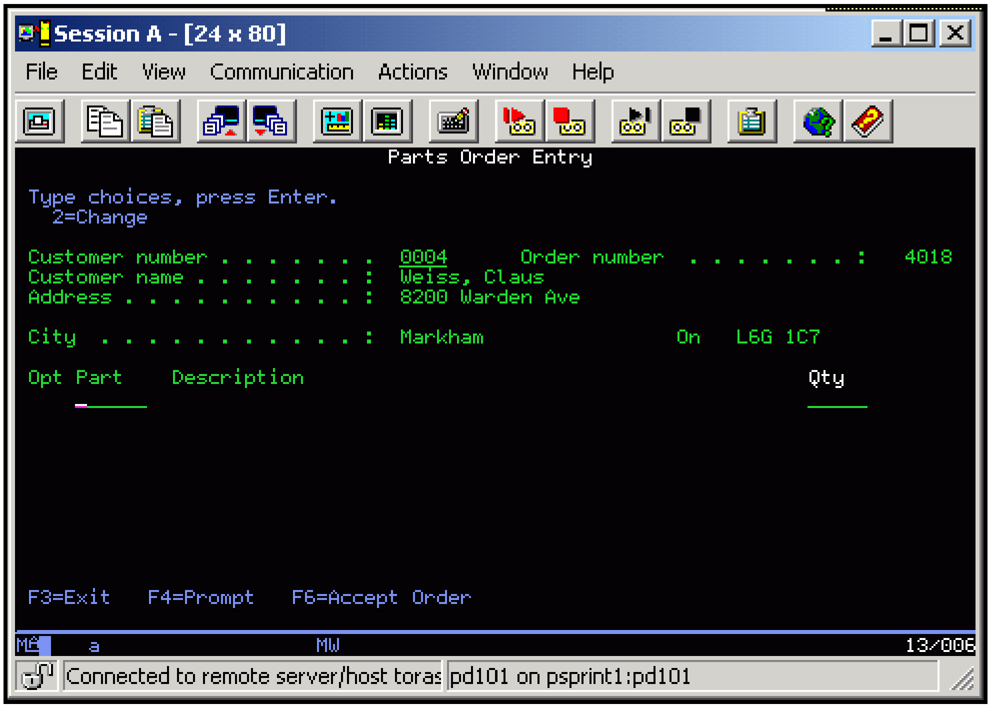
MA a MW 11/044
Connected to remote server/host toras
pd101 on psprint1:pd101

Figure 21. ***Customer List*** *displays*

2. In the **Options** field, type **1** to select a customer.

3. Press the **Enter** key to proceed.

   This returns you to the initial panel now filled with detailed data for the selected customer (see Figure 22).



```
                               Parts Order Entry

Type choices, press Enter.
  2=Change

Customer number . . . . . . .  0004      Order number  . . . . . . . :   4018
Customer name . . . . . . . :  Weiss, Claus
Address . . . . . . . . . . :  8200 Warden Ave

City  . . . . . . . . . . . :  Markham                 On   L6G 1C7

Opt Part     Description                                           Qty
    ______                                                         _____









F3=Exit    F4=Prompt    F6=Accept Order

MA      a                 MW                                              13/006
Connected to remote server/host toras pd101 on psprint1:pd101
```

Figure 22. Panel filled with detailed data for the selected customer

Notice that an order number has been assigned as well. You are now ready to order some parts.

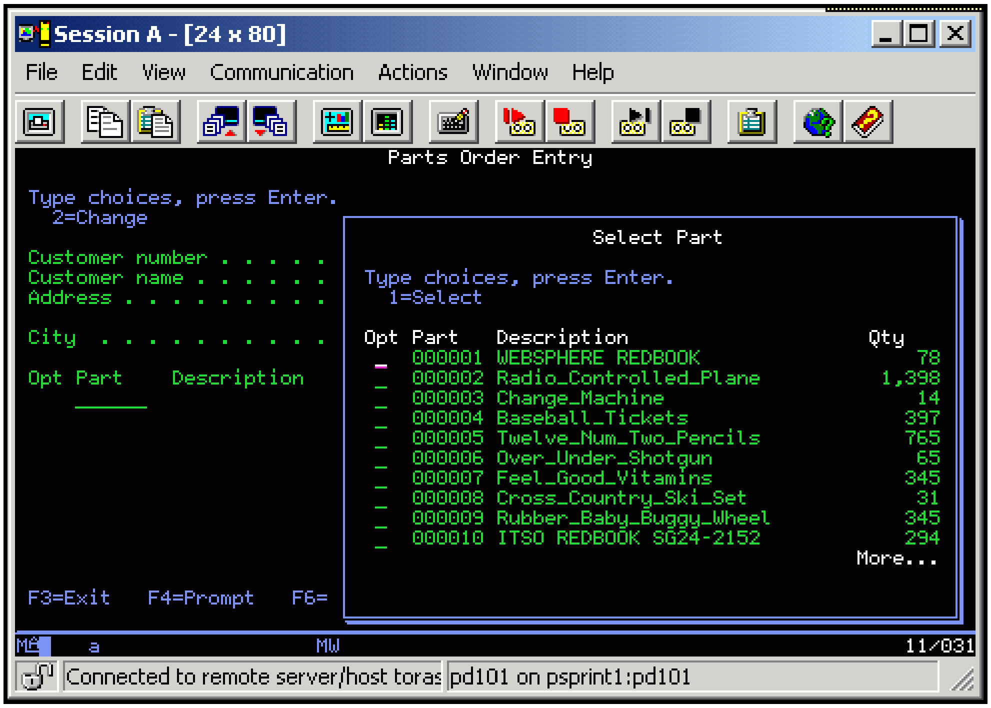
Exercise 1.4: Filling the order

The process of selecting parts is similar to the way you selected a customer in the previous exercise. You are able to key in the part number directly or select the part from a selection list. Next, you want to work with the parts selection list.

On the order entry panel:

1. Press command key **F4** to see a list of available parts.

   The Select Part panel opens (see Figure 23).

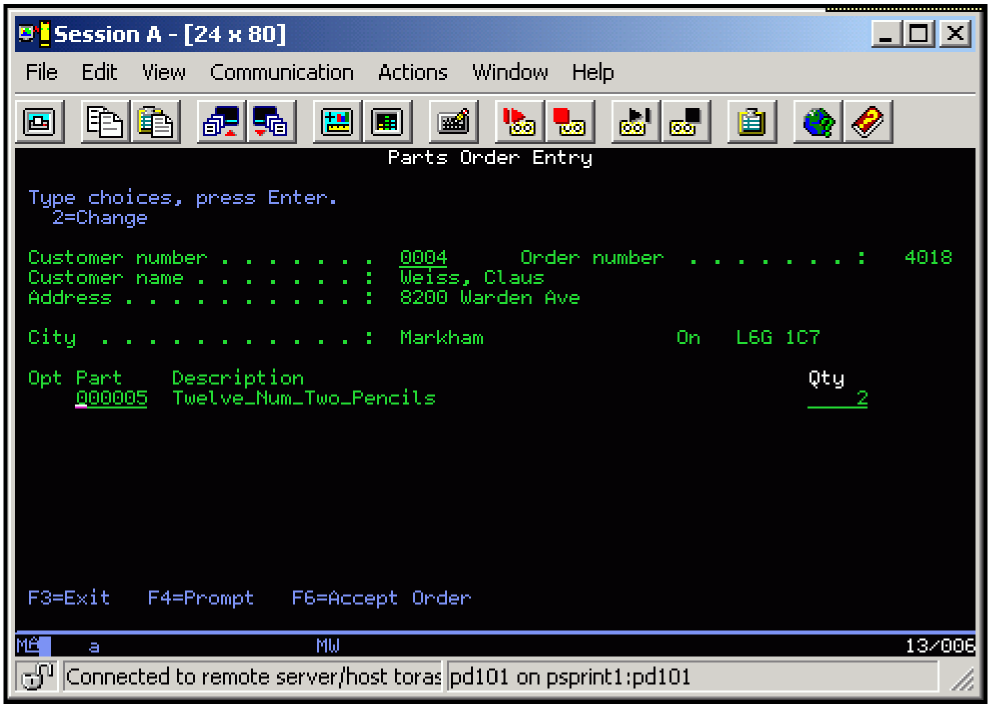


```
                               Parts Order Entry

 Type choices, press Enter.
   2=Change                   ┌──────────────────────────────────────────────────┐
                              │                 Select Part                      │
 Customer number . . . . . .  │                                                  │
 Customer name . . . . . . .  │ Type choices, press Enter.                       │
 Address . . . . . . . . . .  │   1=Select                                       │
                              │                                                  │
 City  . . . . . . . . . . .  │ Opt Part    Description                    Qty   │
                              │  _  000001 WEBSPHERE REDBOOK                78   │
 Opt Part      Description    │  _  000002 Radio_Controlled_Plane        1,398   │
                              │  _  000003 Change_Machine                   14   │
     ______                   │  _  000004 Baseball_Tickets                397   │
                              │  _  000005 Twelve_Num_Two_Pencils          765   │
                              │  _  000006 Over_Under_Shotgun               65   │
                              │  _  000007 Feel_Good_Vitamins              345   │
                              │  _  000008 Cross_Country_Ski_Set            31   │
                              │  _  000009 Rubber_Baby_Buggy_Wheel         345   │
                              │  _  000010 ITSO REDBOOK SG24-2152          294   │
                              │                                        More...   │
                              │                                                  │
 F3=Exit   F4=Prompt   F6=    │                                                  │
                              └──────────────────────────────────────────────────┘
MA     a                  MW                                              11/031
Connected to remote server/host toras pd101 on psprint1:pd101
```

Figure 23. The Select Part panel

2. In the **Options** field, type 1 next to a part that has a lot of quantity remaining.

   Back on the order screen, the part appears in the order; on the top line of the parts list (see Figure 24).


Session A - [24 x 80]
File Edit View Communication Actions Window Help

```
                              Parts Order Entry

 Type choices, press Enter.
   2=Change

 Customer number . . . . . . .  0004      Order number  . . . . . . . :   4018
 Customer name . . . . . . . :  Weiss, Claus
 Address . . . . . . . . . . :  8200 Warden Ave

 City  . . . . . . . . . . . :  Markham               On   L6G 1C7

 Opt Part    Description                                      Qty
     000005  Twelve_Num_Two_Pencils                                2

 F3=Exit   F4=Prompt   F6=Accept Order
```

MA a MW 13/006
Connected to remote server/host toras pd101 on psprint1:pd101

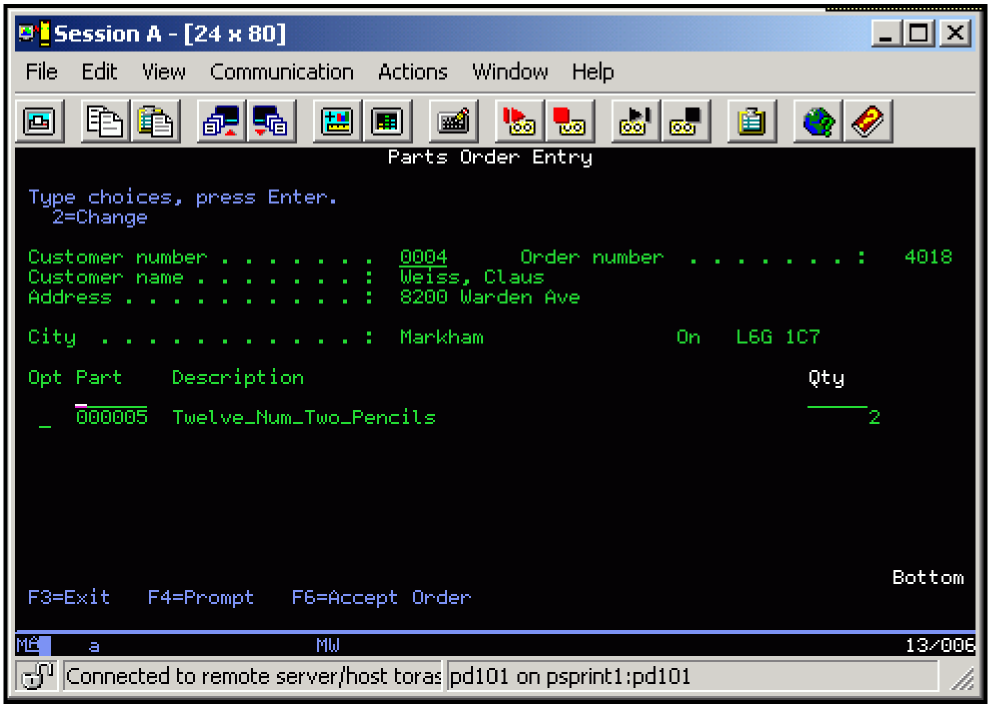
Figure 24. The part appears on the Order screen

   Next, you order a certain quantity of this part.

3. Change the **Qty** field to a value, for example **2**.

4. Press the **Enter** key to add the part and quantity ordered to the order.

   The detail order line for this part, with the quantity specified, is now part of this order (see Figure 25).



```
                               Parts Order Entry

 Type choices, press Enter.
   2=Change

 Customer number . . . . . . .  0004      Order number  . . . . . . . :   4018
 Customer name . . . . . . . :  Weiss, Claus
 Address . . . . . . . . . . :  8200 Warden Ave

 City  . . . . . . . . . . . :  Markham                 On    L6G 1C7

 Opt Part    Description                                         Qty

  _  000005  Twelve_Num_Two_Pencils                              _____2




                                                                      Bottom
 F3=Exit    F4=Prompt    F6=Accept Order
```

```
MA   a                  MW                                             13/006
Connected to remote server/host toras  pd101 on psprint1:pd101
```

Figure 25. Detail order line for chosen part

   Add more parts until your order is complete.

5. Press command key **F4** to see a list of available parts.

6. Select a part from the parts list and press **Enter**.

7. Specify the part quantity and press **Enter**.

Exercise 1.5: Completing the order

Now you are ready to accept the order. You use the Order Entry panel to complete the order:

1. Press command key **F6** to accept the order.

   The order is added to the database, and the next order can be filled. You return to the starting screen of the application.

   You have worked with all panels in this application:

   - The start order entry panel
   - Several help panels
   - The customer list window
   - The full order entry panel
   - The parts list window

   You can add more orders, but if you understand how the user interface of this application works, stop adding orders.

2. Press command key **F3** to exit the application.

You are now ready to put a brand new user interface onto this application.

Recap

You have completed “Reviewing the 5250 order entry application.” You can now understand how to:

- Start the order entry application
- Use the help panels in this application
- Select a customer from the customer selection window
- Select parts from the parts selection window
- Order a certain quantity of parts for the order
- Complete the order
- Exit the application

Before you can use the IBM WebFacing Tool to reface your application, you need to create a WebFacing project in the WebSphere Development Studio Client workbench. You identify the project information, like the DDS source member names to be converted using the IBM WebFacing Tool, and they are stored in a workspace on your workstation.

This next chapter guides you through the process of identifying the DDS and UIM Panel Group source members to be converted by the IBM WebFacing Tool. It also goes through specifying some of the runtime information you need later when running the WebFacing application. Continue to “Creating the WebFacing project.”

Creating the WebFacing project

In this chapter, you learn how to create a WebFacing project, something that you need before you can create a Web user interface for the Order Entry Application. You then learn how to use this WebFacing project to facilitate the conversion of your DDS source and to test the generated output.

A WebFacing project is a complete Web project with the directory structure and files needed to conform to the Java 2 platform, Enterprise Edition (J2EE) standard of a Web application. It also contains additional information unique to WebFacing, like the source to be converted.

To accomplish these learning objectives, several steps are involved, including:

1. Exercise 2.1: Starting WebSphere Development Studio Client workbench
2. Exercise 2.2: Opening the WebFacing perspective
3. Exercise 2.3: Starting the WebFacing Web Project wizard
4. Exercise 2.4: Selecting display file members to convert
5. Exercise 2.5: Specifying the CL command to launch the application
6. Exercise 2.6: Selecting a Web style
7. Exercise 2.7: Completing the WebFacing project information

The exercises in this chapter must be completed in order. Start with “Exercise 2.1: Starting WebSphere Development Studio Client workbench” when you are ready to begin.

Length of time

This chapter takes approximately 25 minutes to complete.

Exercise 2.1: Starting WebSphere Development Studio Client workbench

Start WebSphere Development Studio Client. To start WebSphere Development Studio Client:

1. Click **IBM Rational > IBM WebSphere Development Studio Client Advanced Edition for iSeries V6.0 > WebSphere Development Studio Client Advanced Edition for iSeries** (see Figure 26).

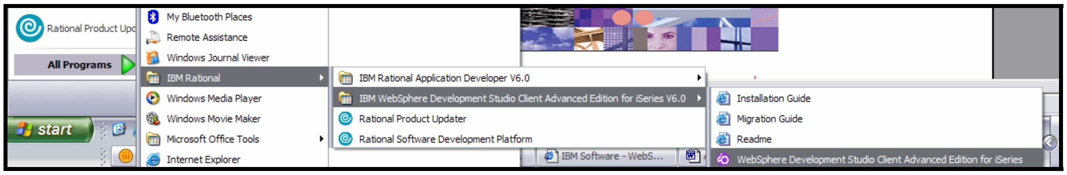
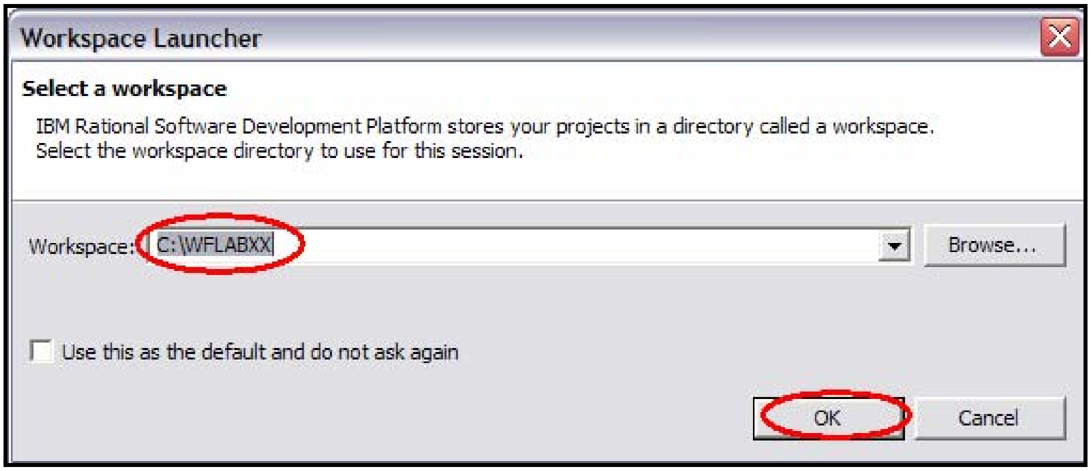


Figure 26. Starting WebSphere Development Studio Client workbench

A dialog appears asking you for the workspace location (unless you used the WebSphere Development Studio Client before and selected not to show this dialog again).

The workspace contains all the information about your WebSphere Development Studio projects. You can accept the default or store the work related to this tutorial in a separate workspace. You can always start with a new workspace later, if you do not want to mix these exercise WebFacing projects with your real work (see Figure 27).


Workspace Launcher

Select a workspace

IBM Rational Software Development Platform stores your projects in a directory called a workspace.
Select the workspace directory to use for this session.

Workspace: C:\WFLABXX

□ Use this as the default and do not ask again

Figure 27. Select a workspace

You might run into problems if your workspace path is too long. If you have a problem with WebFacing projects not running in the WebSphere test environment, and you see an error in the console similar to:

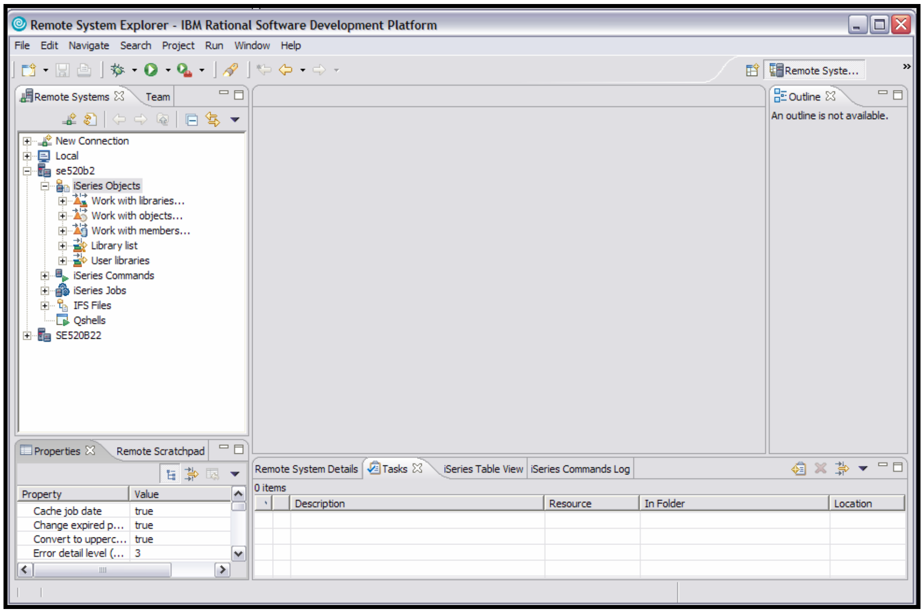
```
6cb33777 SystemOut O Exception in generateFieldAttributes :
java.io.FileNotFoundException:
...
...
...
(The system cannot find the path specified.)
```

You have run into a limitation of the server to handle long directory names. If you use a workspace in a directory with a long path or choose long names for your projects, you might get this error when starting a server or when testing files on a server. If you receive this error, you can take any of the following actions:

1. Move the workspace to a location with a shorter path, such as C:/workspace.
2. Rename the Enterprise Application project or other projects to a shorter name.

1. Replace the default directory structure with your hard drive root and change the name of the workspace directory to the name **WFLABXX**.
2. Click **OK**.

After a few moments of loading, the workbench opens, and the initial window of WebSphere Development Studio Client opens (see Figure 28).



Property	Value
Cache job date	true
Change expired p...	true
Convert to upperc...	true
Error detail level (...	3

Figure 28. The workbench opens

Note: If you did not reset the workspace, or you are working with the standard version of WebSphere Development Studio Client, your environment differs slightly from the screen captures in this tutorial.

Exercise 2.2: Opening the WebFacing perspective

The WebSphere Development Studio Client shows the Remote System Explorer (RSE) perspective by default. This is the perspective that you use to work with i5/OS objects. It allows specifying connections to System i servers and provides the programmer with a similar interface to System i objects as the Program Development Manager (PDM) does in a green-screen environment.

The IBM WebFacing Tool provides its own perspective because it needs to give its users access to unique views and tools targeted towards the WebFacing task. To create a WebFacing project, you first need to open the WebFacing perspective.

To open a WebFacing perspective:

1. Click **Window > Open Perspective > WebFacing** from the workbench menu (see Figure 29).

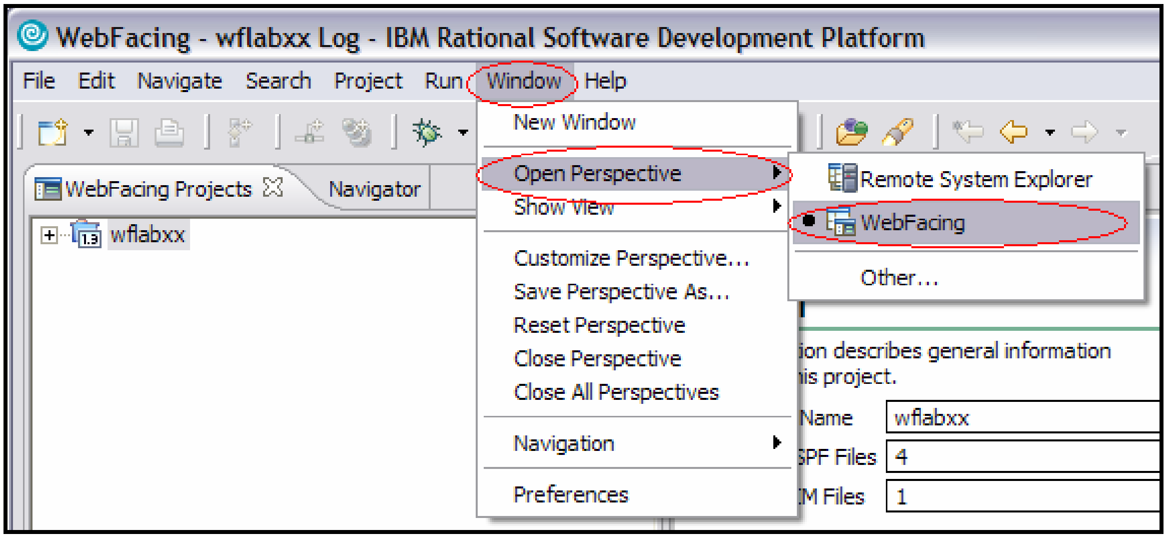
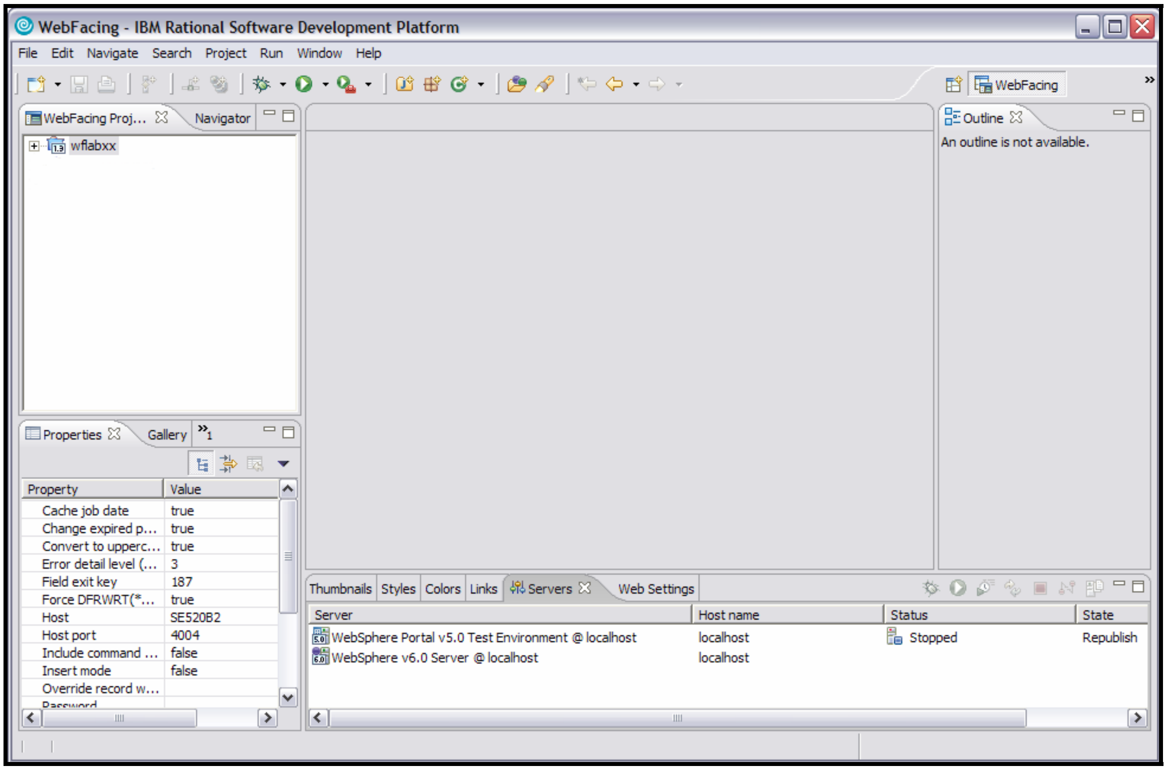


Figure 29. Opening a WebFacing perspective

The workbench now shows the WebFacing perspective with the WebFacing Projects view open in the left pane of the workbench (see Figure 30).



An outline is not available.

Property	Value
Cache job date	true
Change expired p...	true
Convert to upperc...	true
Error detail level (...	3
Field exit key	187
Force DFRWRT(*...	true
Host	SE520B2
Host port	4004
Include command ...	false
Insert mode	false
Override record w...	
Password	

Server	Host name	Status	State
WebSphere Portal v5.0 Test Environment @ localhost	localhost	Stopped	Republish
WebSphere v6.0 Server @ localhost	localhost		

Figure 30. The WebFacing Projects view

You are ready to start the WebFacing Web Project wizard by creating a WebFacing project.

Exercise 2.3: Starting the WebFacing Web Project wizard

Start the WebFacing Web Project wizard:

1. In the WebFacing perspective, click **File > New > WebFacing Web Project** from the workbench menu (see Figure 31).

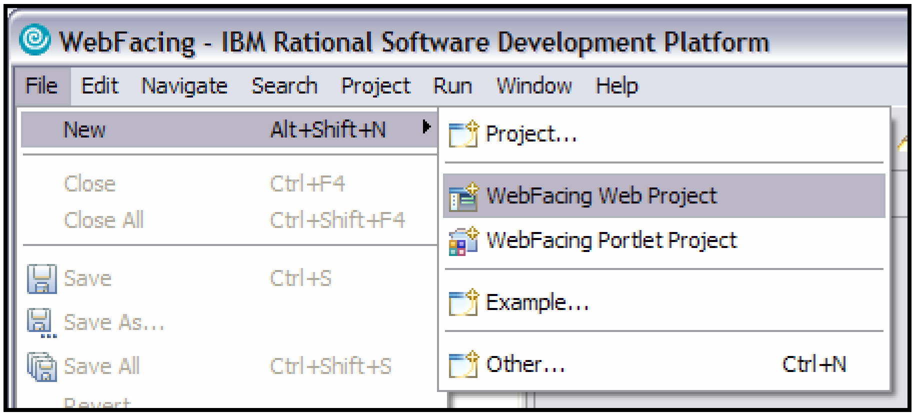
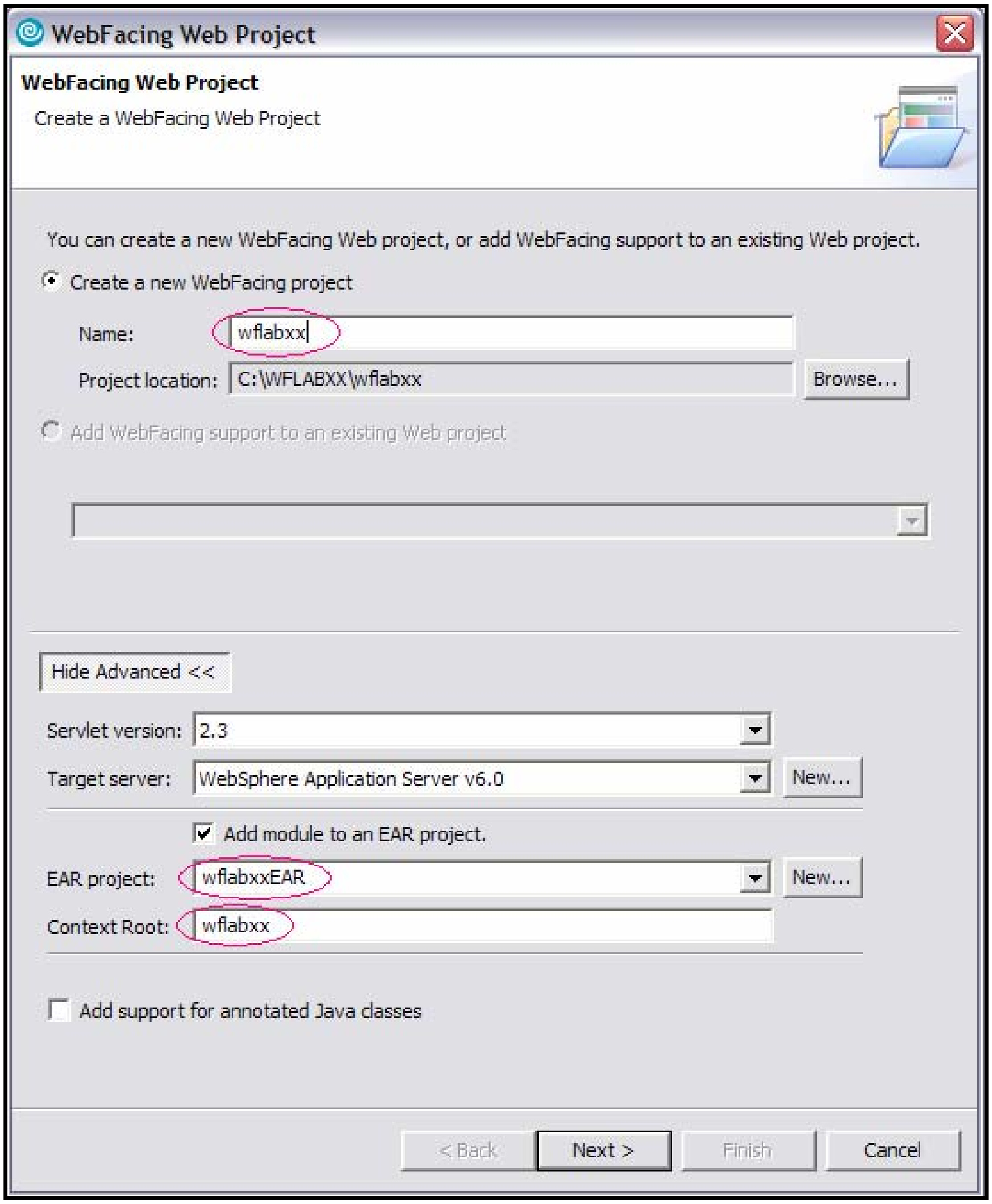


*Figure 31. Starting the **WebFacing Web Project** wizard*

This starts the WebFacing Web Project wizard.

2. In the WebFacing Web Project wizard, you are presented with the WebFacing Web Project page. Click the **Show Advanced** button to see the additional fields shown below (see Figure 32).



WebFacing Web Project

WebFacing Web Project

Create a WebFacing Web Project

You can create a new WebFacing Web project, or add WebFacing support to an existing Web project.

● Create a new WebFacing project

Name: wflabxx

Project location: C:\WFLABXX\wflabxx

○ Add WebFacing support to an existing Web project

Servlet version: 2.3

Target server: WebSphere Application Server v6.0

☑ Add module to an EAR project.

EAR project: wflabxxEAR

Context Root: wflabxx

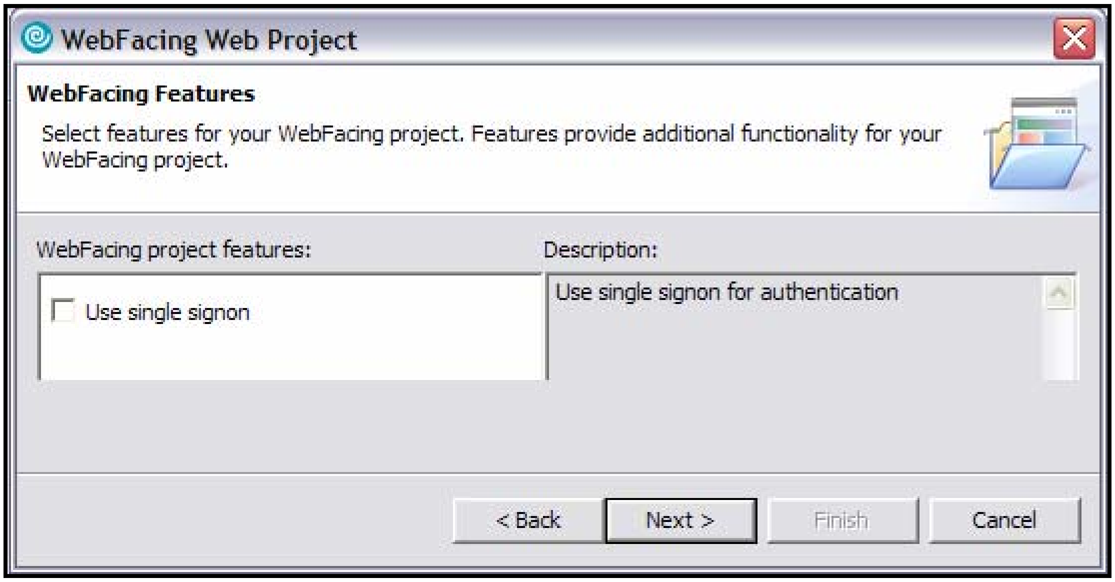
□ Add support for annotated Java classes

Figure 32. The WebFacing Web Project page

3. In the **Project name** field, type the project name `wflabxx`.

   The Enterprise Archive (**EAR project**) File and **Context Root** fields update automatically.

   This creates a unique Application file for this WebFacing project.

4. Click **Next**.
5. The WebFacing Features page is displayed. Click **Next** (see Figure 33)**.**
6. 

WebFacing Web Project

WebFacing Features

Select features for your WebFacing project. Features provide additional functionality for your WebFacing project.

WebFacing project features:

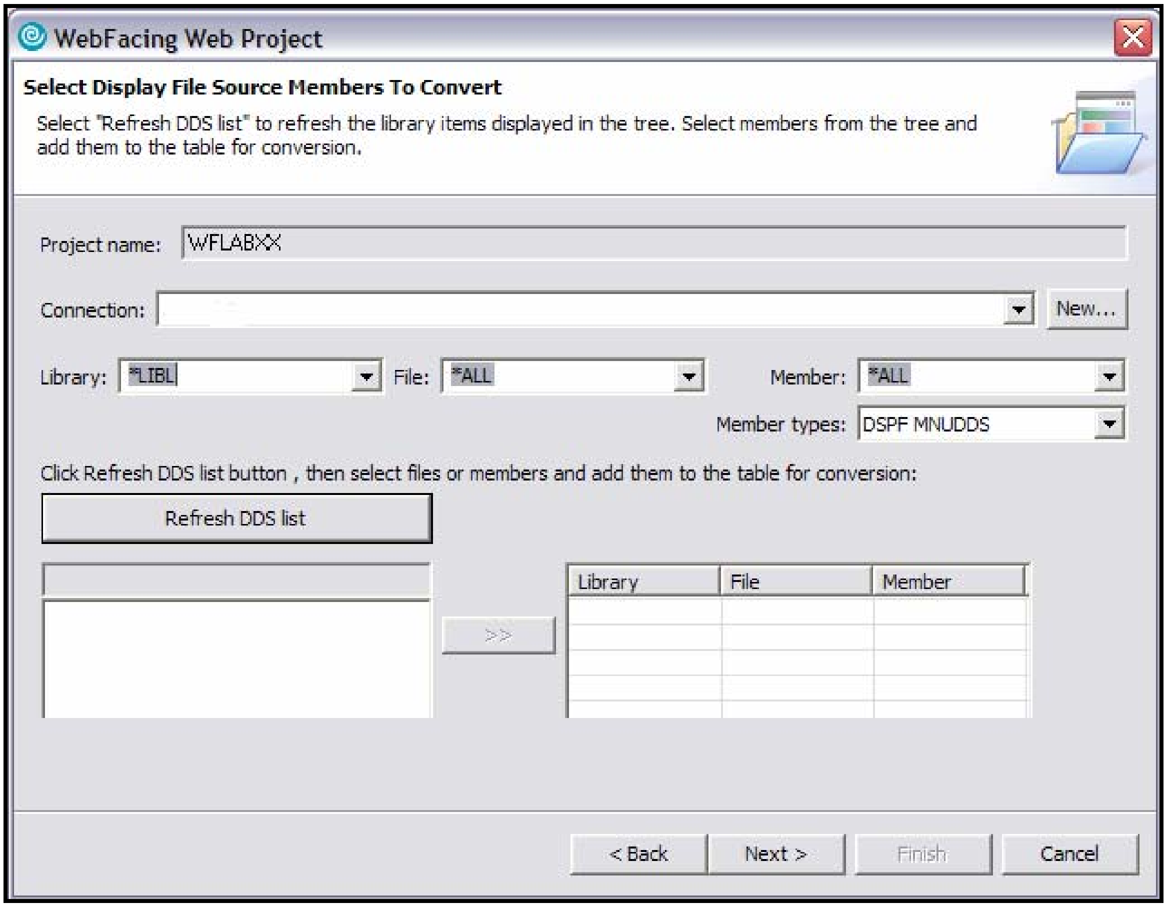
□ Use single signon

Description:

Use single signon for authentication

Figure 33. The WebFacing Features page

7. The Select Display File Source Members to Convert page opens (see Figure 34).



WebFacing Web Project

Select Display File Source Members To Convert

Select "Refresh DDS list" to refresh the library items displayed in the tree. Select members from the tree and add them to the table for conversion.

Project name: WFLABXX

Connection:

Library: *LIBL

File: *ALL

Member: *ALL

Member types: DSPF MNUDDS

Click Refresh DDS list button , then select files or members and add them to the table for conversion:

Library	File	Member

Figure 34. The Select Display File Source Members to Convert page

Exercise 2.4: Selecting display file members to convert

On this second page of the WebFacing project wizard, you need to specify the name of the System i model that contains your display file DDS source, as well as the members of the source file you want to convert. Specifically, the IBM WebFacing Tool needs to know these names:

- Server name
- Library name
- Source file name
- Member name

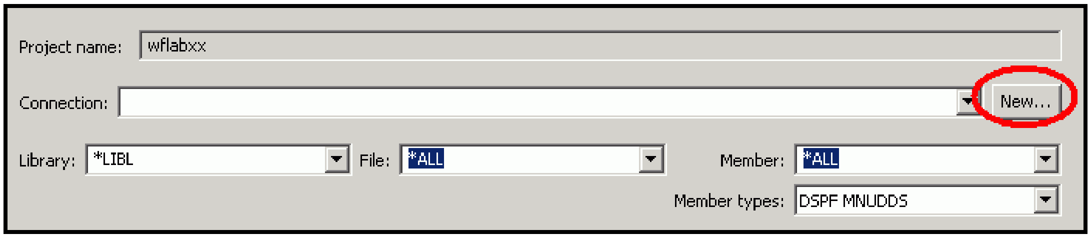
Note: You need to identify members for all record formats used in your application. If you miss one, and if at run time, the record format is requested, the user gets a **Page not found** error.

Because you are starting with a new workspace, you need to specify the System i name to specify where the DDS is located.

Connecting to a System i model

To connect to a System i model:

1. Click **New** beside the **Connection** list (see Figure 35).



Project name: wflabxx

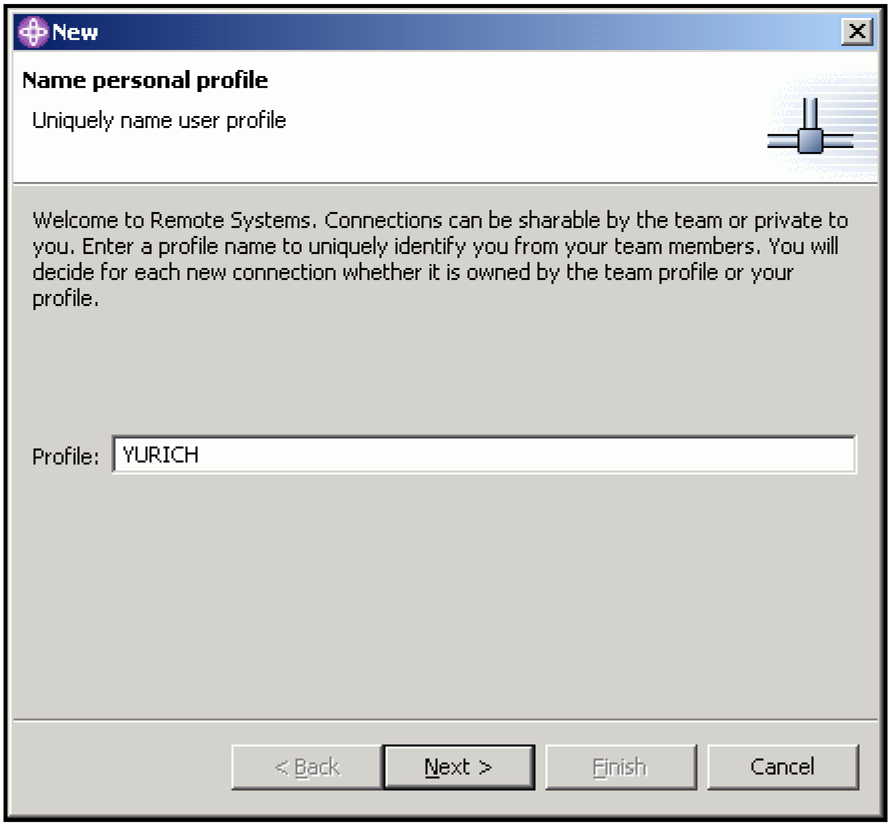
Connection:

Library: *LIBL File: *ALL Member: *ALL

Member types: DSPF MNUDDS

*Figure 35. Click **New** beside the **Connection** list.*

The Name personal profile page opens (see Figure 36).



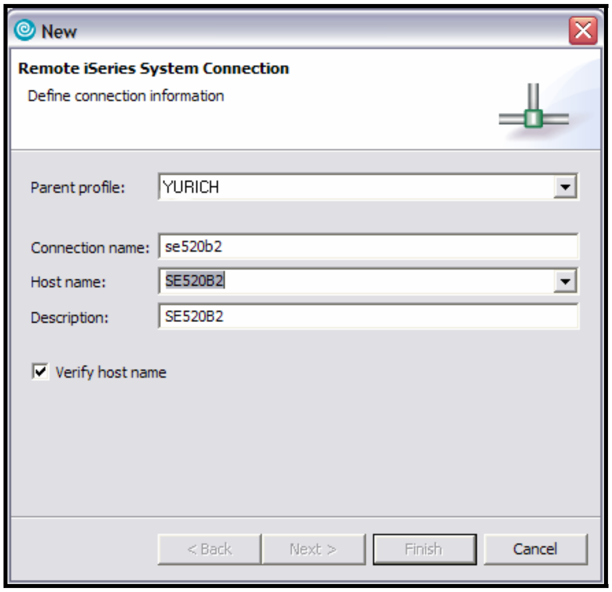
Welcome to Remote Systems. Connections can be sharable by the team or private to you. Enter a profile name to uniquely identify you from your team members. You will decide for each new connection whether it is owned by the team profile or your profile.

Profile: YURICH

Figure 36. The Name personal profile page

2. Accept the default profile name as this is your own private connection.
3. Click **Next**.

The Remote iSeries System Connection page opens (see Figure 37).


New
Remote iSeries System Connection
Define connection information

Parent profile: YURICH

Connection name: se520b2

Host name: SE520B2

Description: SE520B2

☑ Verify host name

< Back
Next >
Finish
Cancel

Figure 37. The Remote iSeries System Connection page

4. In the **Host name** field, type the name of the host system, for example **s400a**. This is the name of your System i model where you have the WebFacing run time running (STRTCPSVR *WebFacing).

   Note: This is your System i model name; do not use the one shown unless your system is named **s400a**.

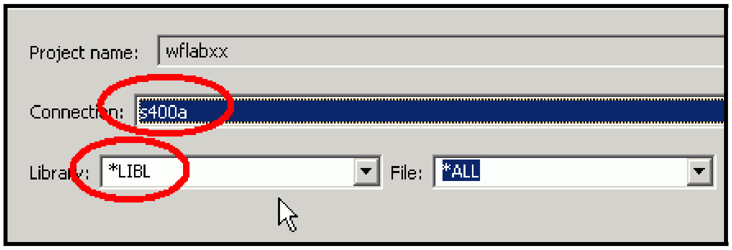
   The connection name is filled automatically with the same value.

5. Leave the **Verify host name** check box selected to verify that the host name or IP address exists.
6. Click **Finish**.

   The connection is created and can be re-used by all tools in the workbench.

   The connection wizard also checks whether the System i host can be reached and if this test is successfully completed, then you return to the Select Display File Source Members to Convert page.

7. Check that your System i model is selected in the **Connection** list (see Figure 38).



Project name: wflabxx

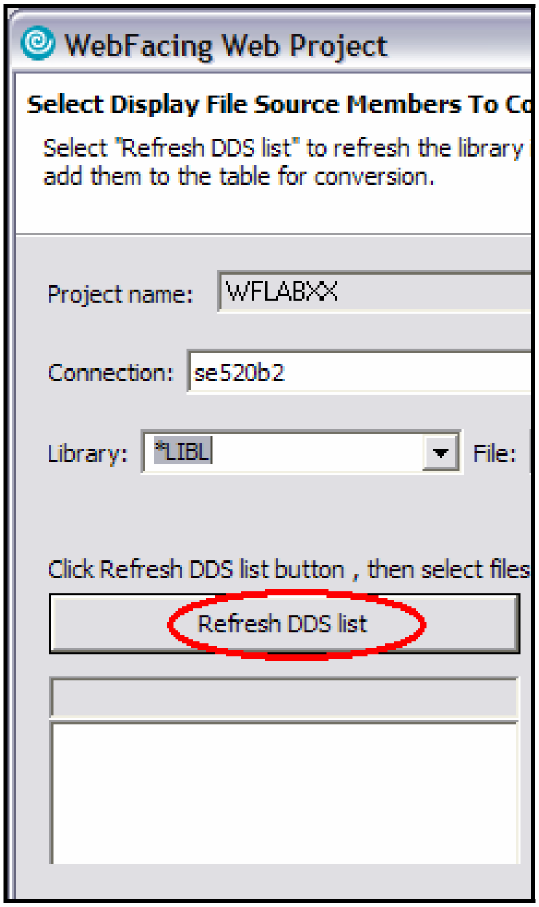
Connection: s400a

Library: *LIBL

File: *ALL

Figure 38. Check that your System i model is selected in the ***Connection*** *list.*

8. Make sure that ***LIBL** is selected in the **Library** list.
9. Click **Refresh DDS list** (see Figure 39**)**.



WebFacing Web Project

Select Display File Source Members To Co

Select "Refresh DDS list" to refresh the library
add them to the table for conversion.

Project name: WFLABXX

Connection: se520b2

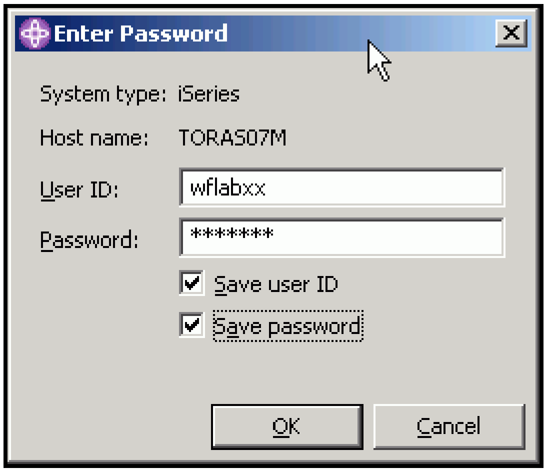
Library: *LIBL

File:

Click Refresh DDS list button , then select files

Figure 39. Click ***Refresh DDS list****.*

A User ID and password dialog opens (see Figure 40).



Enter Password

System type: iSeries

Host name: TORAS07M

User ID: wflabxx

Password: *******

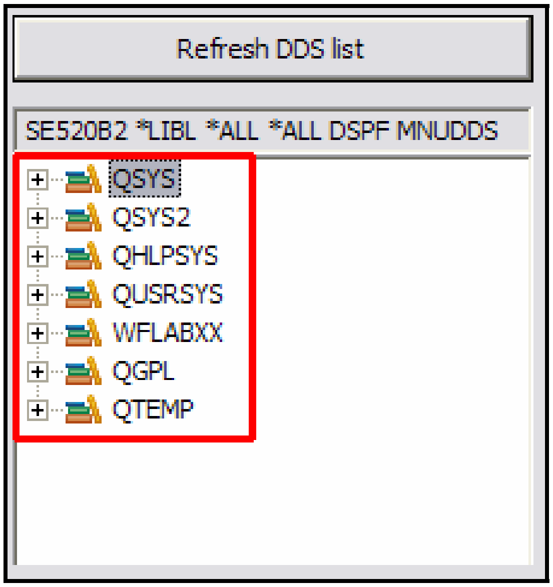
☑ Save user ID

☑ Save password

Figure 40. User ID and password dialog

10. In the **User ID** field, type your User ID. Use the same User ID that you used when running the Order Entry application in “Reviewing the 5250 order entry application.”
11. In the **Password** field, type your password.
12. Select the **Save User ID** check box.
13. Select the **Save password** check box.
14. Click **OK**.

A connection to the System i model is established. The library list of your System i job is displayed under the **Refresh DDS list** button on the Select Display File Source Members to Convert page (see Figure 41).



SE520B2 *LIBL *ALL *ALL DSPF MNUDDS

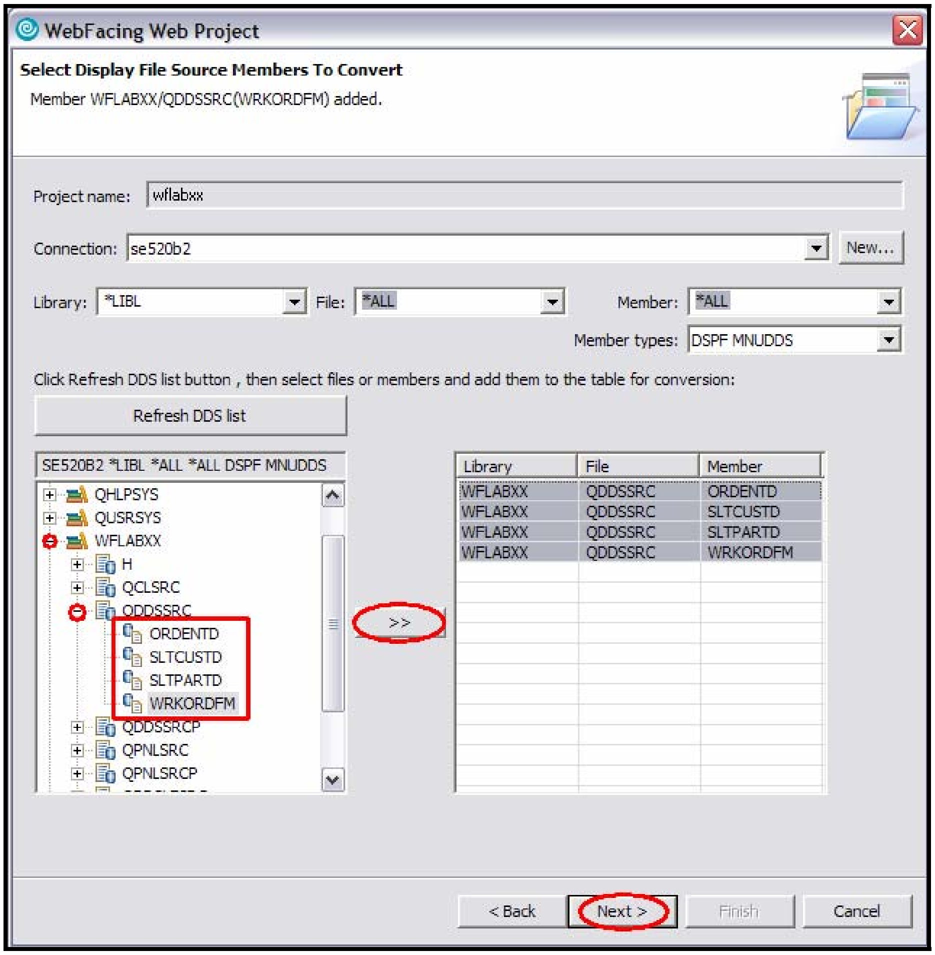
- QSYS
- QSYS2
- QHLPSYS
- QUSRSYS
- WFLABXX
- QGPL
- QTEMP

Figure 41. Library list of your i5/OS job

Selecting DDS members

To select DDS members to convert:

1. Select the WFLABXX library from the list.
2. Click the plus sign (**+**) beside the WFLABXX library to expand it (see Figure 42).



WebFacing Web Project

Select Display File Source Members To Convert

Member WFLABXX/QDDSSRC(WRKORDFM) added.

Project name: wflabxx

Connection: se520b2

Library: *LIBL File: *ALL Member: *ALL

Member types: DSPF MNUDDS

Click Refresh DDS list button , then select files or members and add them to the table for conversion:

Library	File	Member
WFLABXX	QDDSSRC	ORDENTD
WFLABXX	QDDSSRC	SLTCUSTD
WFLABXX	QDDSSRC	SLTPARTD
WFLABXX	QDDSSRC	WRKORDFM

Figure 42. Selecting DDS members

3. Select the **QDDSSRC** source file from the expanded list.
4. Expand **QDDSSRC** and select all members from the list.
5. Click the >> push button in the middle of the page to copy the selected members over to the list of members to be converted at the right.

Tip: To move all members in a source file you can select the source file icon itself and click the >> push button. This adds all members in a source file to the list of members to be converted. The IBM WebFacing Tool now knows which DDS members to convert for this project. The conversion has not been done but the information is stored for future use.

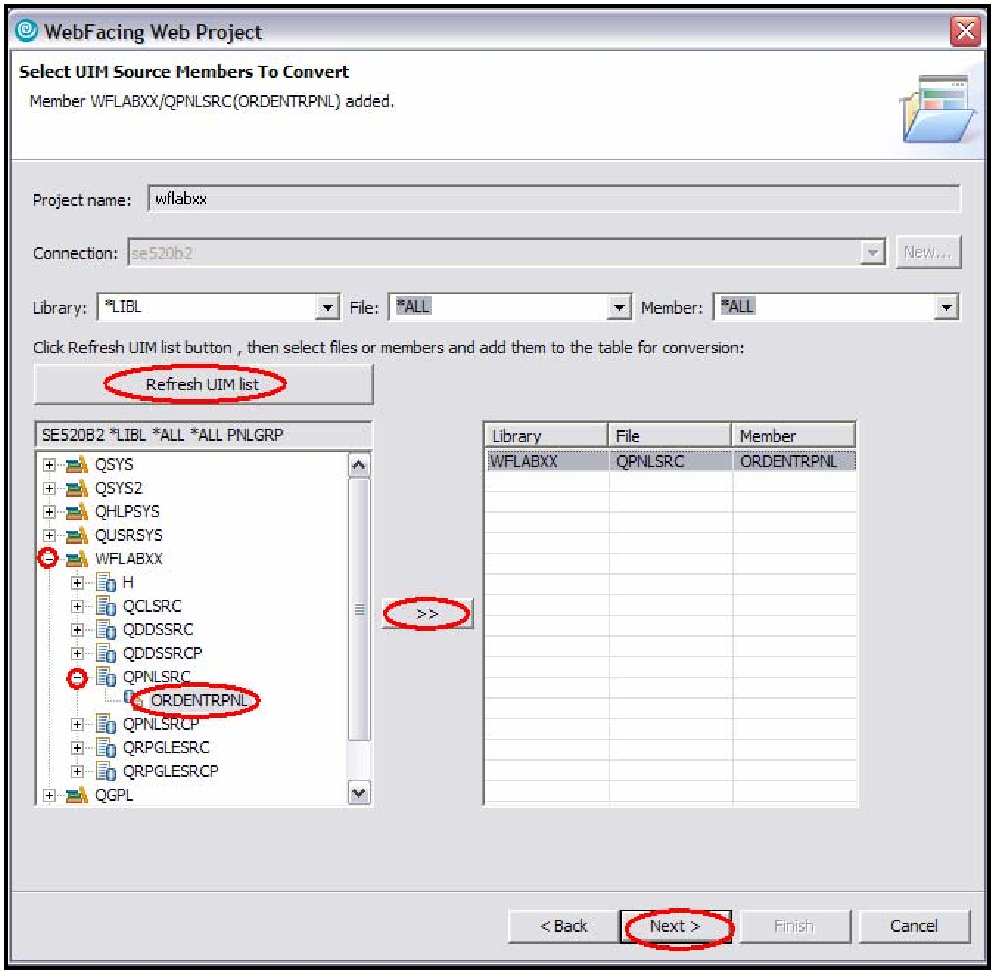
Next, you convert UIM source members.

Selecting UIM source members

To select UIM source members to convert:

1. Click **Next** to proceed to the next page of the wizard.

   The Select UIM Source Members to Convert page opens (see Figure 43).



WebFacing Web Project

Select UIM Source Members To Convert

Member WFLABXX/QPNLSRC(ORDENTRPNL) added.

Project name: wflabxx

Connection: se520b2

Library: *LIBL File: *ALL Member: *ALL

Click Refresh UIM list button , then select files or members and add them to the table for conversion:

Library	File	Member
WFLABXX	QPNLSRC	ORDENTRPNL

Figure 43. The Select UIM Source Members to Convert page

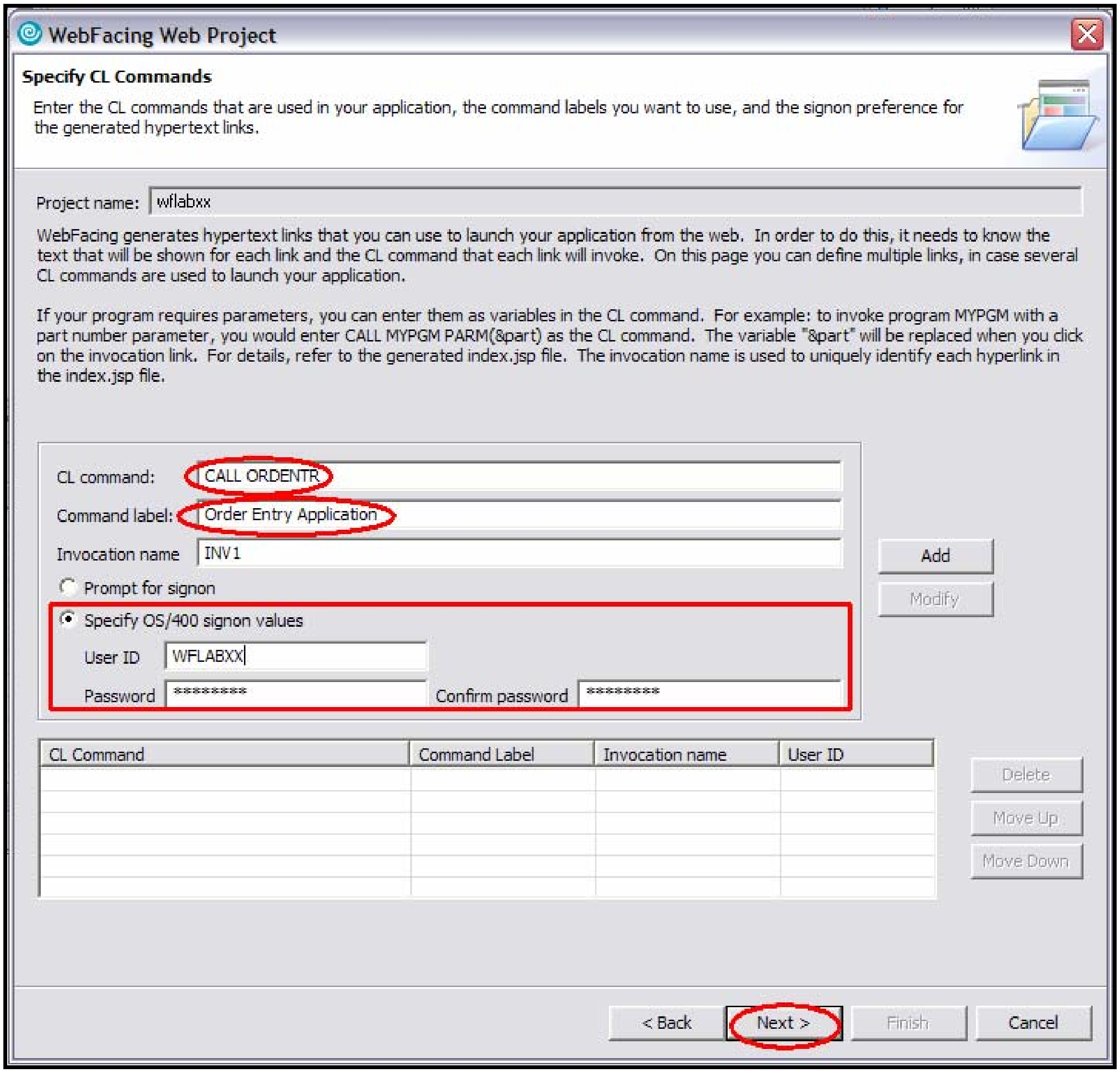
Now you have to identify the panel group source containing the help information for the Order Entry application.

2. Click **Refresh UIM list** push button.
3. Expand library WFLABXX.
4. Select the **QPNLSRC** source file from the expanded list of library.
5. Click the >> push button in the middle of the page to copy all members over to the list of members to be converted at the right. There is actually only one panel group member **ORDENTRPNL** in this source file.
6. Click **Next** to proceed to the next page of the wizard.

Exercise 2.5: Specifying the CL command to launch the application

You now provide the information that allows the WebFacing Web Project wizard to create the initial index.jsp page to start the Order Entry application. Type the same invocation command text that you typed on the command line to start the Order Entry application in the 5250 emulation session during the first exercise in “Reviewing the 5250 order entry application” (which is, CALL ORDENTR).

The reason that this command text is important to remember is that the WebFacing run time needs to know the invocation command for your application to send this command to the System i model to start your application from the browser. The page in the WebFacing Web Project wizard allow you to specify the necessary CL command information and Sign-on information (see Figure 44).



WebFacing Web Project

Specify CL Commands

Enter the CL commands that are used in your application, the command labels you want to use, and the signon preference for the generated hypertext links.

Project name: wflabxx

WebFacing generates hypertext links that you can use to launch your application from the web. In order to do this, it needs to know the text that will be shown for each link and the CL command that each link will invoke. On this page you can define multiple links, in case several CL commands are used to launch your application.

If your program requires parameters, you can enter them as variables in the CL command. For example: to invoke program MYPGM with a part number parameter, you would enter CALL MYPGM PARM(&part) as the CL command. The variable "&part" will be replaced when you click on the invocation link. For details, refer to the generated index.jsp file. The invocation name is used to uniquely identify each hyperlink in the index.jsp file.

CL command: CALL ORDENTR

Command label: Order Entry Application

Invocation name INV1

○ Prompt for signon

● Specify OS/400 signon values

User ID WFLABXX

Password ********* Confirm password *********

CL Command	Command Label	Invocation name	User ID

Figure 44. Specifying CL commands

Follow the following steps to specify the CL commands:

1. In the **CL command** field, type: `CALL ORDENTR`.
2. In the **Command Label** field, type: `Order Entry Application`.

   Note: If you do not type anything into the **Command Label** field, the CL command input is copied into this field.

3. Leave the default value **INV1** in the **Invocation name** field.
4. Click the **Sign-on with specified values** radio button.

   This automatically applies the user ID and password you used when you connected to your System i model in the previous dialog for selecting source members for invocation of this application. You use the setup here for testing purposes to make it easier for you to start the Order Entry application without specifying the authentication information. Do not worry about security in your production environment; this can be easily changed later. You learn how to do this later in the tutorial.

5. Click **Add** on the right side of the page.

   Note: Make sure the text and command you typed into the fields are actually shown in the CL command table at the bottom of the page.

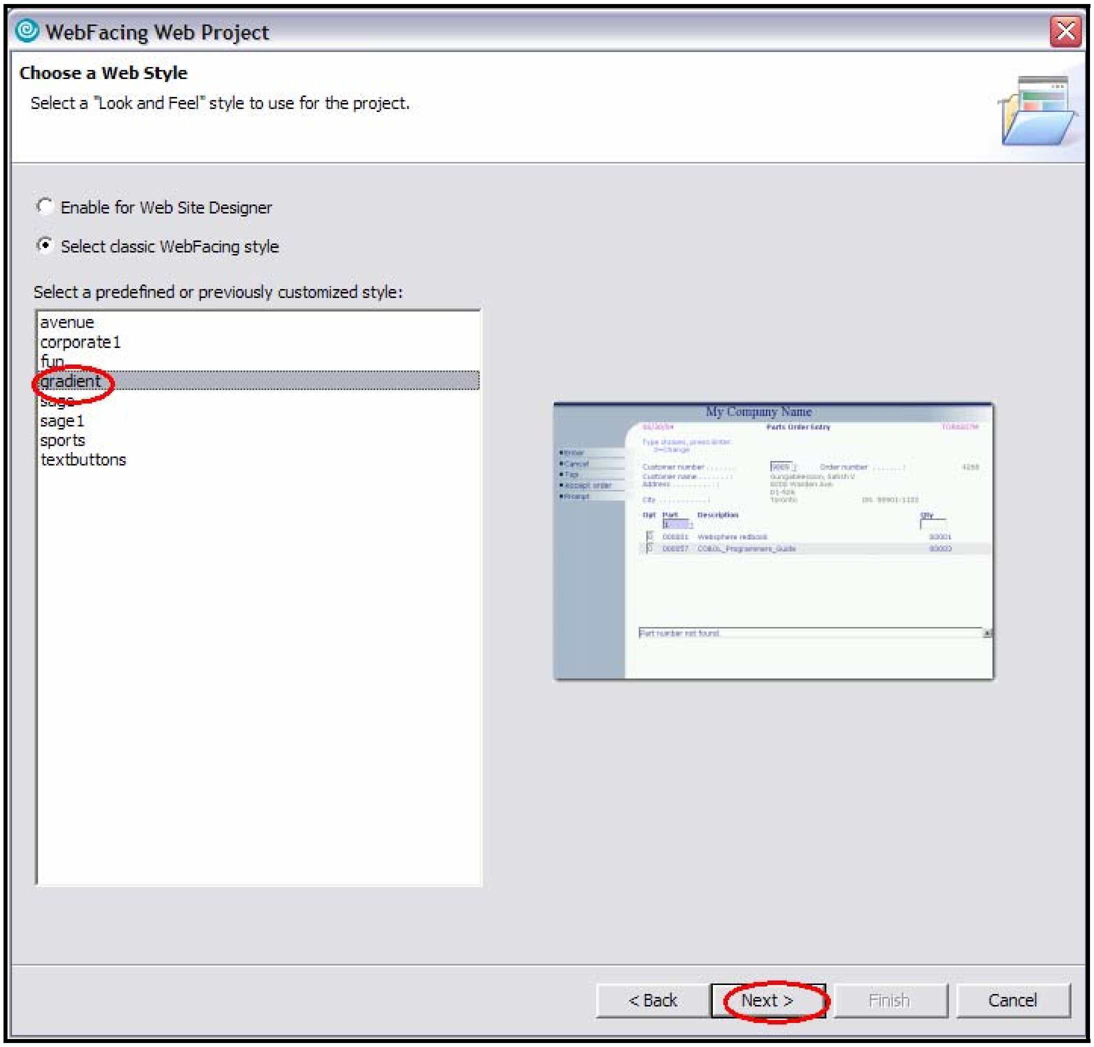
6. Click **Next** at the bottom of the wizard page. The Choose a Web Style page opens.

Exercise 2.6: Selecting a Web style

Next, you select a Web style for your converted screens.

To select a Web style:

1. Select the **gradient style** from the list of available styles.
2. Click **Next** (see Figure 45).



WebFacing Web Project

Choose a Web Style

Select a "Look and Feel" style to use for the project.

○ Enable for Web Site Designer

● Select classic WebFacing style

Select a predefined or previously customized style:

- avenue
- corporate1
- fun
- gradient
- sage
- sage1
- sports
- textbuttons

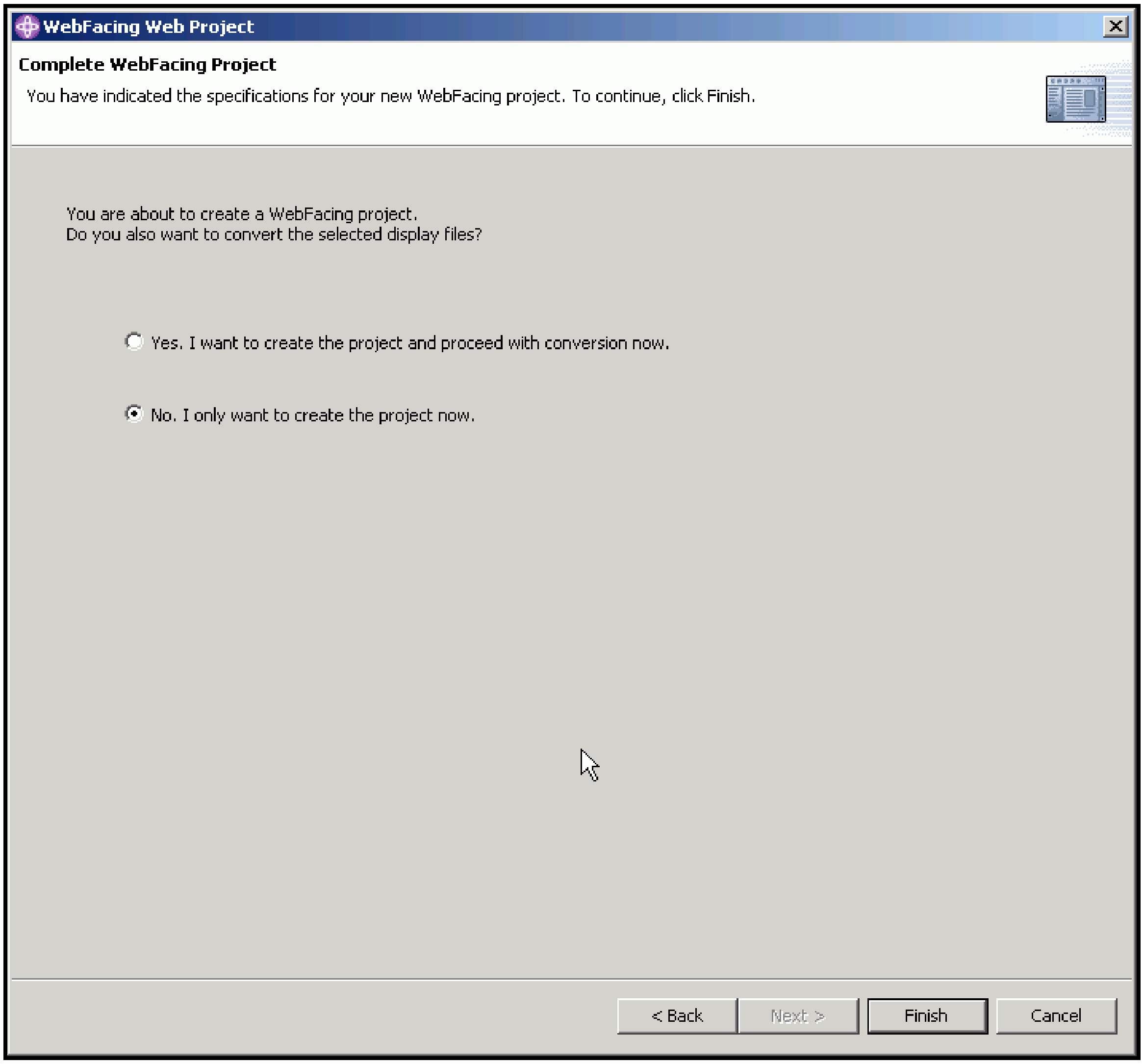
Figure 45. Selecting a Web style

The Completing WebFacing Project page opens.

Exercise 2.7: Completing the WebFacing project information

On this page, you have the choice of creating the project and converting the source in one step or only creating the WebFacing project.

For our example purposes, you just create the WebFacing project. You do not need to convert the source right now. Conversion is part of the next chapter (see Figure 46).



WebFacing Web Project

Complete WebFacing Project

You have indicated the specifications for your new WebFacing project. To continue, click Finish.

You are about to create a WebFacing project.
Do you also want to convert the selected display files?

○ Yes. I want to create the project and proceed with conversion now.

● No. I only want to create the project now.

Figure 46. Completing the WebFacing project information

To complete the WebFacing project information:

1. Select the **No. I only want to create the project now** radio button.
2. Click **Finish**.

The WebFacing project is created. The workbench opens in the WebFacing perspective with your new WebFacing project in the WebFacing Projects view (see Figure 47.)

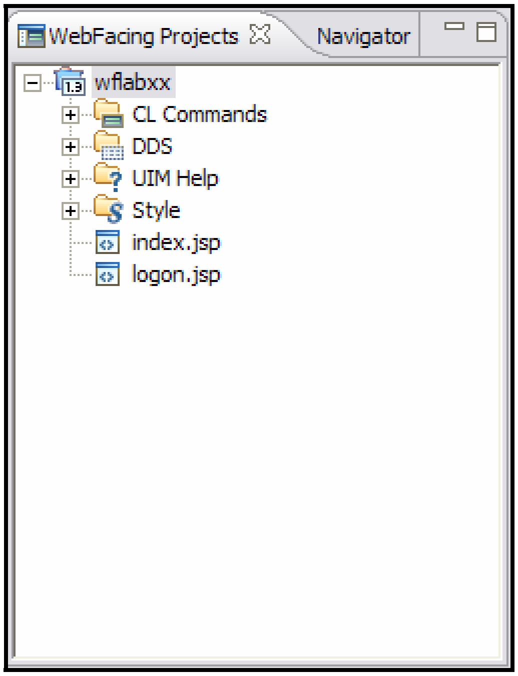


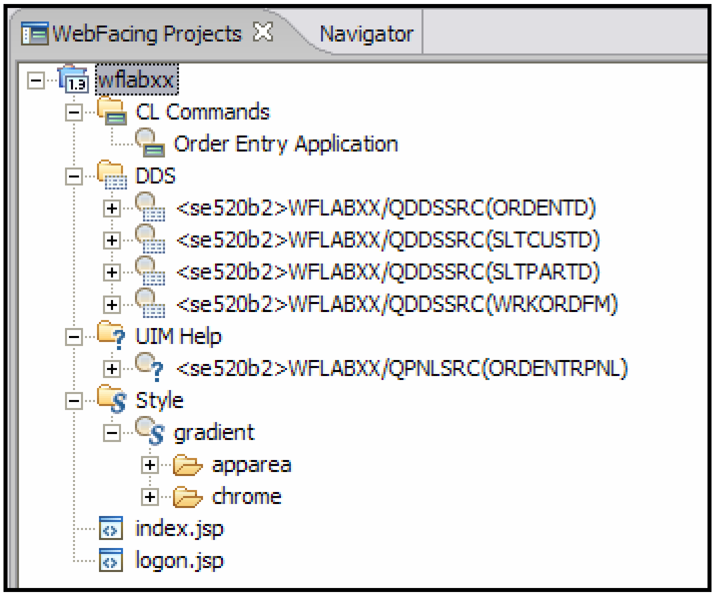
Figure 47. Completed WebFacing project

Notice that the WebFacing project icon indicates which J2EE level is used for the project.

If you expand the **wflabxx** project, notice that several files and folders have been added to this WebFacing project:

1. The information for display file source members to be converted
2. The information for panel group source members to be converted
3. The CL ommand that is used to start the Order Entry application
4. The style to be used for the converted screens

When you expand the folders and files under your WebFacing project, you can see the information that is captured for this project (see Figure 48).



WebFacing Projects | Navigator

- wflabxx
  - CL Commands
    - Order Entry Application
  - DDS
    - <se520b2>WFLABXX/QDDSSRC(ORDENTD)
    - <se520b2>WFLABXX/QDDSSRC(SLTCUSTD)
    - <se520b2>WFLABXX/QDDSSRC(SLTPARTD)
    - <se520b2>WFLABXX/QDDSSRC(WRKORDFM)
  - UIM Help
    - <se520b2>WFLABXX/QPNLSRC(ORDENTRPNL)
  - Style
    - gradient
      - apparea
      - chrome
  - index.jsp
  - logon.jsp

Figure 48. Project information

3. Click each folder or file. Click the plus (**+**) sign beside each file or folder.

   Tip: If you later need to add source members to the WebFacing project, right-click the DDS or UIM help icon and click **Add** on the pop-up menu. This launches the WebFacing Web Project wizard and display the correct dialog (see Figure 49).

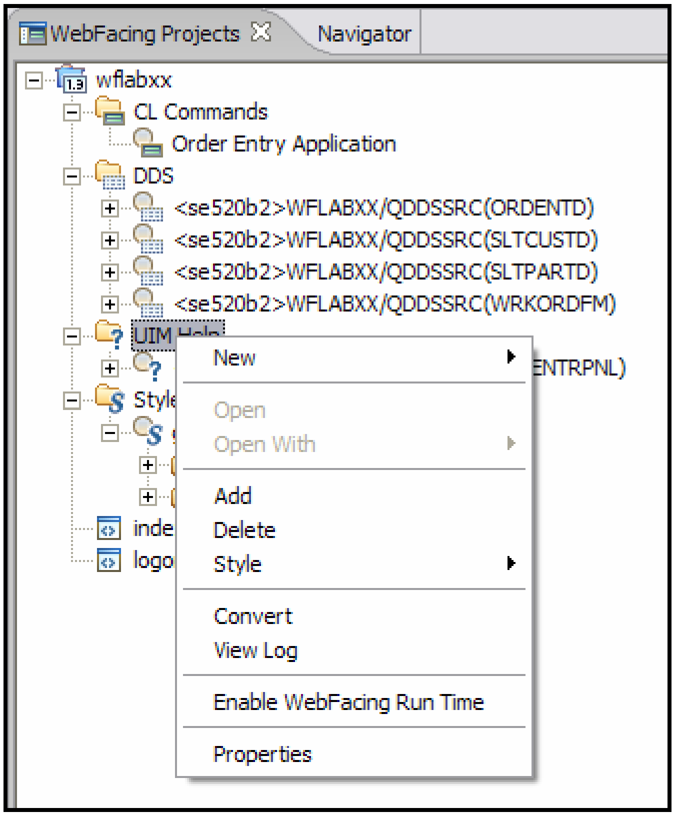

WebFacing Projects
Navigator
wflabxx
CL Commands
Order Entry Application
DDS
<se520b2>WFLABXX/QDDSSRC(ORDENTD)
<se520b2>WFLABXX/QDDSSRC(SLTCUSTD)
<se520b2>WFLABXX/QDDSSRC(SLTPARTD)
<se520b2>WFLABXX/QDDSSRC(WRKORDFM)
UIM Help
ENTRPNL)
Style
inde
logo
New
Open
Open With
Add
Delete
Style
Convert
View Log
Enable WebFacing Run Time
Properties

Figure 49. Launching the WebFacing Web Project wizard

Recap

You have completed “Creating the WebFacing project.” You now understand how to:

- Start WebSphere Development Studio Client
- Launch the WebFacing project wizard
- Complete the pages of this wizard to create a new WebFacing project called wflabxx
- Add to this WebFacing project the following information:
    - Which DDS display file source members to use for the WebFacing conversion
    - Which UIM panel group source members to convert
    - The CL command that is used to create the initial index.jsp page to start the Order Entry application
    - The Web style to be used for the resulting Web pages
- Inspect the WebFacing project. Verify that all information entered in the WebFacing project wizard has been captured.

You have created a WebFacing project in WebSphere Development Studio Client. In this project, you specified the display file source and panel group source to convert and the location of these source members, which is determined by the System i model. The IBM WebFacing Tool uses this information during the conversion process. You have to specify in this next chapter whether you want to convert all source members belonging to this project or selected ones only.

A word regarding security

The reason why you do not have to logon to the System i model during conversion is that your user ID and password have been stored during the WebFacing project setup and are reused here.

Continue to “Converting selected source members.”

Converting selected source members

In the previous chapter, you specified the complete information the IBM WebFacing Tool needs to convert your application. You now do the conversion itself.

In this chapter, you learn how the WebFacing conversion creates the JSP files for the browser User Interface description, XML files for record format layouts, and several other files needed for the Web application. You learn how to select the convert option, start the conversion, and work with the conversion logs to check the results of the WebFacing conversion.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 3.1: Selecting the project and source members
- Exercise 3.2: Starting the WebFacing conversion
- Exercise 3.3: Checking the conversion logs

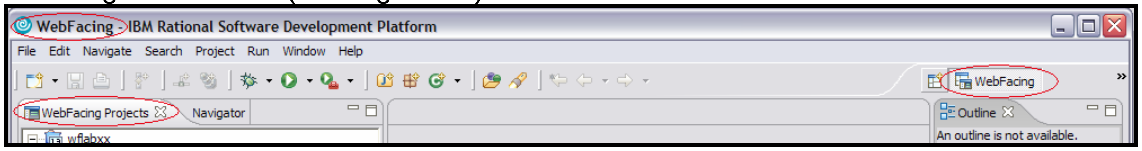
The exercises within each chapter must be completed in order. Start with “Exercise 3.1: Selecting the project and source members” when you are ready to begin.

Length of time
This chapter takes approximately 10 minutes to complete.

Exercise 3.1: Selecting the project and source members

You use WebSphere Development Studio Client to do the WebFacing conversion. If it is not up and running already, start it now. If it is still up and running on your workstation, you are ready to go ahead.

To determine if your active window is displaying the WebFacing perspective, look for the following visual clues (see Figure 50):



An outline is not available.

Figure 50. Selecting the project and source members

If the WebFacing perspective of your project is not visible, you need to open the WebFacing perspective.

Opening the WebFacing perspective

To open the WebFacing perspective:

1. Check the taskbar of the workbench for the WebFacing perspective icon WebFacing and if you find it, click the icon to switch to the WebFacing perspective.

2. If you do not see the WebFacing perspective icon, select the Open a Perspective icon from the task bar (see Figure 51).

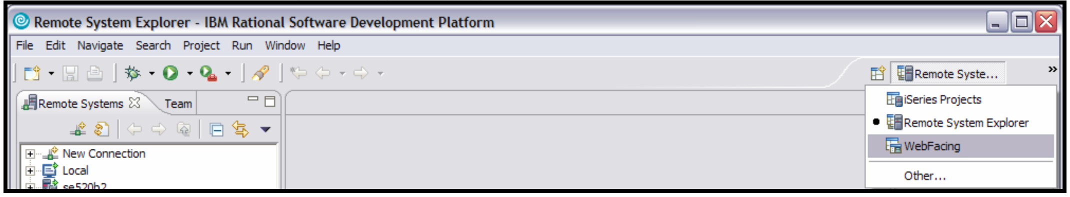

Remote System Explorer - IBM Rational Software Development Platform
File Edit Navigate Search Project Run Window Help
Remote Syste...
Remote Systems
Team
New Connection
Local
se520b2
iSeries Projects
Remote System Explorer
WebFacing
Other...

Figure 51. Opening a WebFacing perspective

3. Select **WebFacing** from the drop-down menu.

4. The WebFacing perspective is the active perspective. Look for the following clues in your active window (see Figure 52):

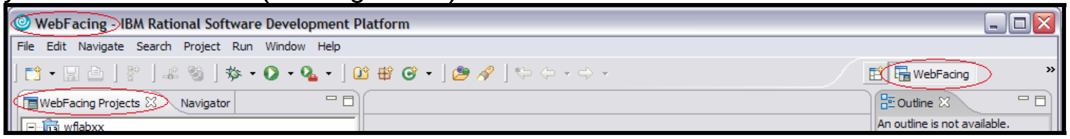

WebFacing - IBM Rational Software Development Platform
File Edit Navigate Search Project Run Window Help
WebFacing
WebFacing Projects
Navigator
wflabxx
Outline
An outline is not available.

Figure 52. The WebFacing perspective is the active perspective.

Next, you start the conversion in the WebFacing perspective.

Selecting members to convert

To select members to convert:

1. Select the WebFacing project (wflabxx) you have been working on in the previous chapter.
2. Expand this project by clicking the plus sign (**+**) beside its icon in the **WebFacing Projects** view. You see the following list displayed (see Figure 53):

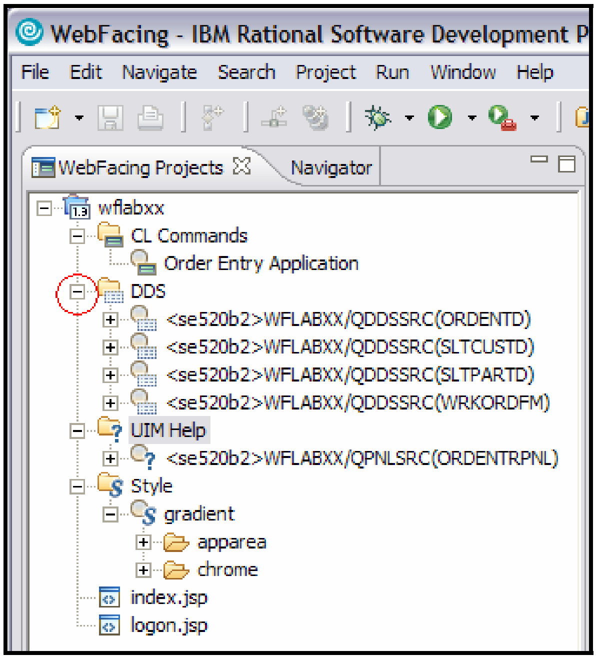

WebFacing - IBM Rational Software Development P
File Edit Navigate Search Project Run Window Help
WebFacing Projects
Navigator
wflabxx
CL Commands
Order Entry Application
DDS
<se520b2>WFLABXX/QDDSSRC(ORDENTD)
<se520b2>WFLABXX/QDDSSRC(SLTCUSTD)
<se520b2>WFLABXX/QDDSSRC(SLTPARTD)
<se520b2>WFLABXX/QDDSSRC(WRKORDFM)
UIM Help
<se520b2>WFLABXX/QPNLSRC(ORDENTRPNL)
Style
gradient
apparea
chrome
index.jsp
logon.jsp

Figure 53. Expand project to see DDS list

You see a DDS folder in the expanded view.

If you do not see a DDS folder, you might be in the **Navigator** view and not in the **WebFacing Projects** view.

3. If the view title bar is not **WebFacing Projects**, locate the **WebFacing Projects** tab and select it (see *Figure 54*).

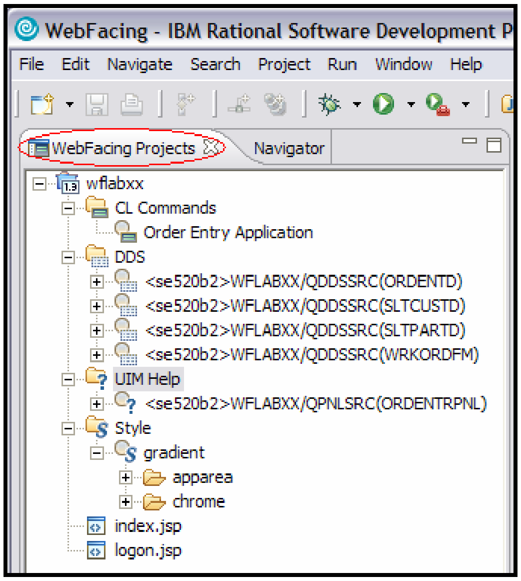

WebFacing - IBM Rational Software Development P
File Edit Navigate Search Project Run Window Help
WebFacing Projects
Navigator
wflabxx
CL Commands
Order Entry Application
DDS
<se520b2>WFLABXX/QDDSSRC(ORDENTD)
<se520b2>WFLABXX/QDDSSRC(SLTCUSTD)
<se520b2>WFLABXX/QDDSSRC(SLTPARTD)
<se520b2>WFLABXX/QDDSSRC(WRKORDFM)
UIM Help
<se520b2>WFLABXX/QPNLSRC(ORDENTRPNL)
Style
gradient
apparea
chrome
index.jsp
logon.jsp

Figure 54. The WebFacing Projects view

The WebFacing Projects view is now active, and you can find the DDS folder (see Figure 55).

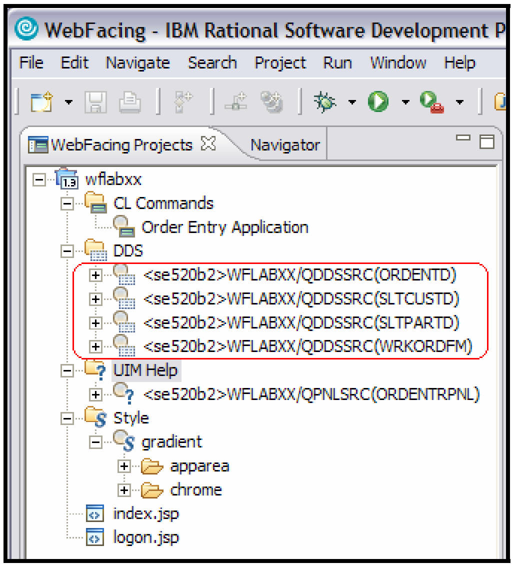

WebFacing - IBM Rational Software Development P
File Edit Navigate Search Project Run Window Help
WebFacing Projects
Navigator
wflabxx
CL Commands
Order Entry Application
DDS
<se520b2>WFLABXX/QDDSSRC(ORDENTD)
<se520b2>WFLABXX/QDDSSRC(SLTCUSTD)
<se520b2>WFLABXX/QDDSSRC(SLTPARTD)
<se520b2>WFLABXX/QDDSSRC(WRKORDFM)
UIM Help
<se520b2>WFLABXX/QPNLSRC(ORDENTRPNL)
Style
gradient
apparea
chrome
index.jsp
logon.jsp

Figure 55. The WebFacing Projects view is active.

4. Expand the DDS folder, so that you can see all three of the members that you selected in the previous chapter.

Next, you convert these members.

Exercise 3.2: Starting the WebFacing conversion

From here, you can select individual members to convert or a collection of members. Later in this tutorial, when you change only one source member, you select a single member to convert. Because you want to convert all members in the project, you can work with the WebFacing project icon to convert DDS and UIM panel group members.

1. Right-click the **wflabxx** WebFacing project (see Figure 56).

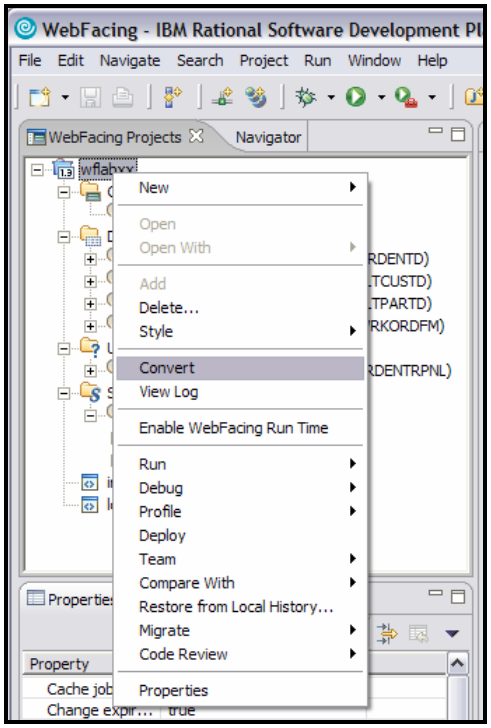
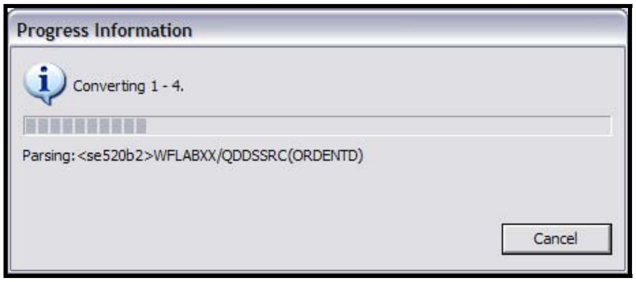


Figure 56. Right-click the wflabxx WebFacing project.

2. Click **Convert** on the pop-up menu. The conversion process starts. You see a progress dialog, indicating what members are being converted (see Figure 57).


Progress Information

Converting 1 - 4.

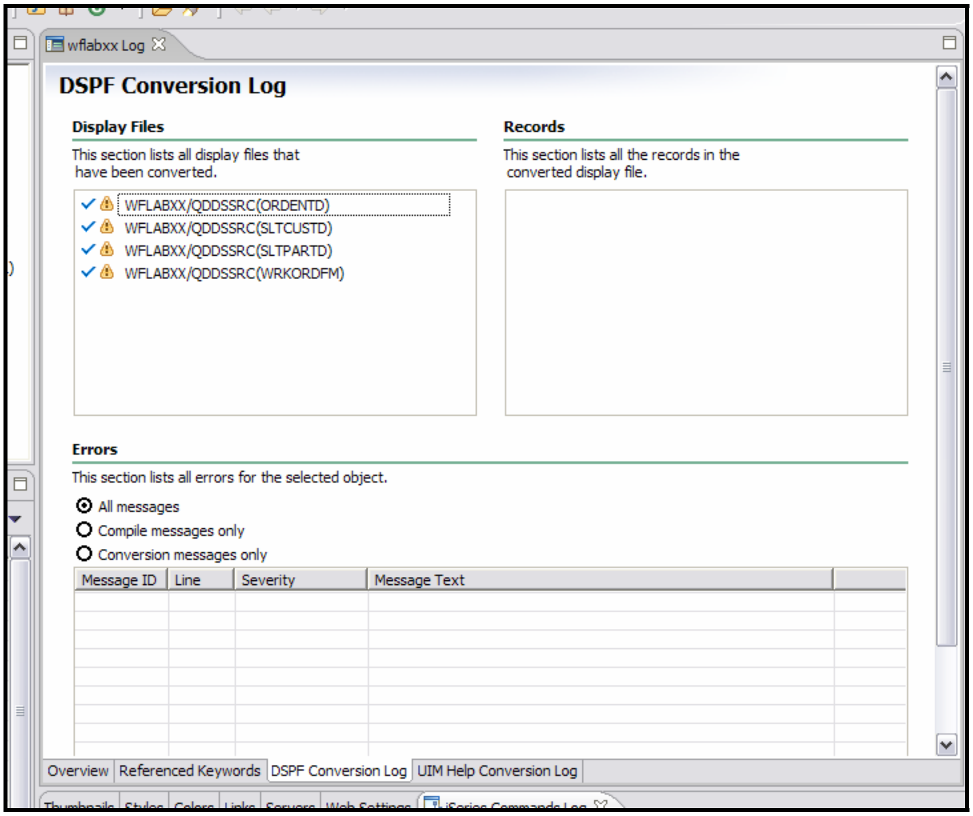
Parsing:<se520b2>WFLABXX/QDDSSRC(ORDENTD)

Figure 57. Progress Information screen

Wait until the progress dialog disappears, indicating the WebFacing conversion has finished.

Exercise 3.3: Checking the conversion logs

Notice the conversion log in the right pane of the workbench (see Figure 58).



wflabxx Log

DSPF Conversion Log

Display Files

This section lists all display files that have been converted.

- ✓ ⚠ WFLABXX/QDDSSRC(ORDENTD)
- ✓ ⚠ WFLABXX/QDDSSRC(SLTCUSTD)
- ✓ ⚠ WFLABXX/QDDSSRC(SLTPARTD)
- ✓ ⚠ WFLABXX/QDDSSRC(WRKORDFM)

Records

This section lists all the records in the converted display file.

Errors

This section lists all errors for the selected object.

● All messages
○ Compile messages only
○ Conversion messages only

Message ID	Line	Severity	Message Text

Figure 58. Checking the conversion logs

To check the conversion log:

1. Click the **Overview** tab.
2. Click the **Reference Keywords** tab.
3. Click the **DSPF Conversion Log** tab.
4. Click the **UIM Help Conversion Log** tab.

As you click each tab, you see more details about each log.

You have now converted the DDS display file and panel group source.

The DDS source and panel group source is now available in a form that a browser understands. The first important step to move your application to the Web is done. Notice that the conversion itself was easy, because you already did the hard part when you created the WebFacing project. Now that the project exists, the IBM WebFacing Tool has all the information it needs readily available. Subsequent DDS conversions are required whenever you make DDS changes. This reprocessing is standard task and is easy to perform. DDS members can be individually converted as they are modified, or you can reconvert the entire DDS collection all at the same time.

Recap

You have completed “Converting selected source members.” You now understand how to:

- Select the WebFacing project to convert
- Select all DDS source members and all panel group source members that are part of the WebFacing project
- Start the WebFacing conversion process
- Analyze the conversion logs to ensure no errors have occurred during conversion

You have used the IBM WebFacing Tool to convert the user interface of the Order Entry application and you now want to see the conversion results. The IBM WebFacing Tool created a complete Web application that can be run in a WebSphere Application Server. The WebSphere Application Server test environment and the Web project in the workbench are completely integrated and are known to the WebSphere Application Server test environment even though none of the files from the Web Project have been published (copied) to the server.

The **index.jsp** file that has been created by the WebFacing conversion routine is part of this Web project. It is used as the default page for this Web application. When the application is up and running you go through the same scenarios as in “Reviewing the 5250 order entry application” when you went through the green-screen panels of the Order Entry application but now you are using the new WebFacing user interface.

Next, you run your application with its new user interface. Continue to “Running the WebFacing application.”

Running the WebFacing application

You normally have to copy all the files of your Web project to a WebSphere Application Server and then run your application at that server location. However, there is a WebSphere Application Server included with WebSphere Development Studio Client, and it is called the WebSphere test environment. Using the WebSphere test environment makes the process of locally testing your WebFacing application easier than testing on a remote WebSphere Application Server.

For your production environment, later in this lab, you copy all files needed to your production application server, and you learn in the section entitled “Exporting to WebSphere Application Server Express for iSeries 6.0.1” how to go through this process for the various versions of WebSphere Application Server. In your real application development environment, you test thoroughly in the test environment and then export the WebFacing project files to a remote WebSphere Application Server for final testing, before moving the Web-enabled application to your production WebSphere Application Server environment. The “Exporting to WebSphere Application Server Express for iSeries 6.0.1” exercise shows you the steps required to deploy to a remote WebSphere Application Server environment. Until you are sure your WebFacing application is ready for public use, you can use the test environment on your workstation, which simplifies the testing of Web applications.

In this chapter, you learn how to test your WebFacing application in the WebSphere test environment. You learn how to specify in the WebFacing perspective that you want to run your project in the WebSphere Development Studio Client test environment. From the initial Web page, you learn how to select the link created by the IBM WebFacing Tool to invoke the WebFacing Order Entry application. The ability to call (programmatically) a WebFaced application also be discussed.

To accomplish these learning objects, several steps are involved, including:

- Exercise 4.1: Showing the index.jsp page
- Exercise 4.2: Testing the WebFacing application
- Exercise 4.3: Testing the help support
- Exercise 4.4: Bypass Log-in screen

The exercises within each chapter must be completed in order. Start with “Exercise 4.1: Showing the index.jsp page” when you are ready to begin.

Length of time

This chapter take approximately 15 minutes to complete.

Exercise 4.1: Showing the index.jsp page

To show the index.jsp page:

1. Right-click the **wflabxx** WebFacing Projects folder (see Figure 59).

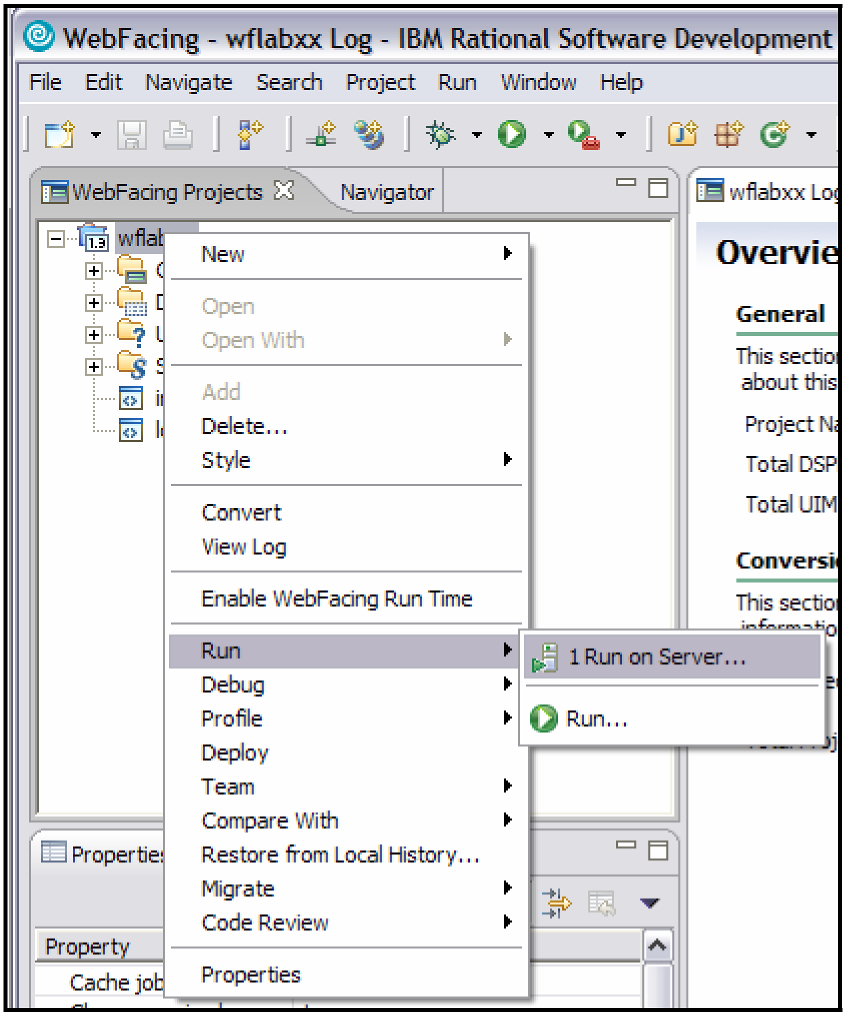
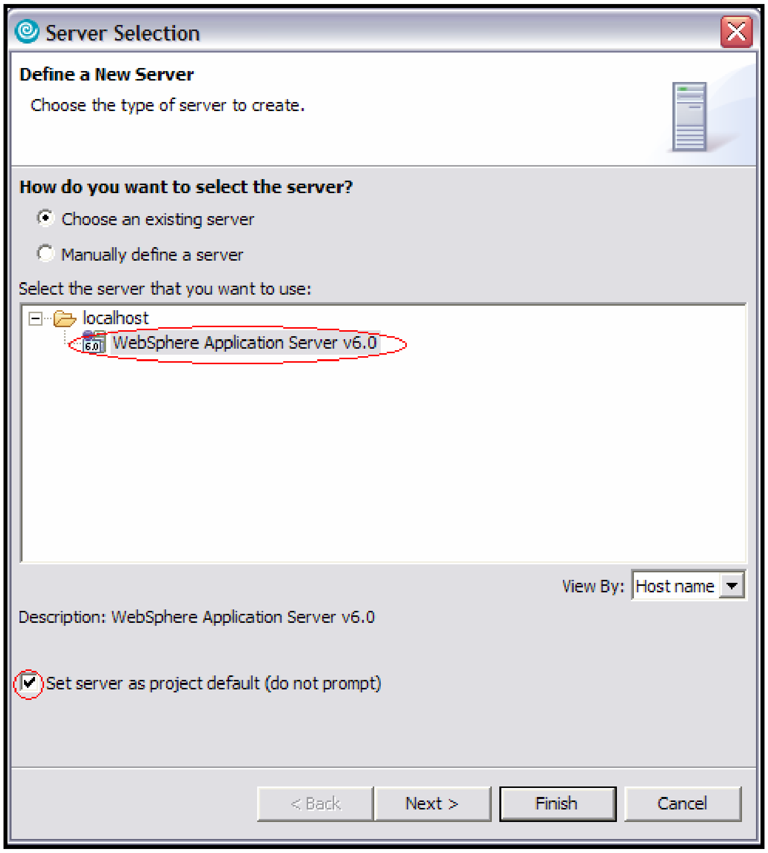


Figure 59. Choose the wflabxx WebFacing Projects folder

2. Click **Run > Run on server** on the pop-up menu.

   A Server Selection dialog opens (see Figure 60).



Server Selection

Define a New Server

Choose the type of server to create.

How do you want to select the server?

● Choose an existing server

○ Manually define a server

Select the server that you want to use:

- localhost
  - WebSphere Application Server v6.0

View By: Host name

Description: WebSphere Application Server v6.0

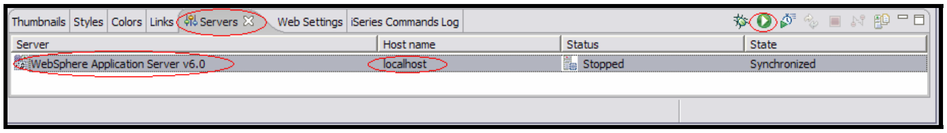
☑ Set server as project default (do not prompt)

Figure 60. Server Selection dialog opens

You can avoid having this dialog open every time you select run on server.

3. Select **WebSphere Application Server v6.0** under **localhost**.
4. Select the **Set server as project default (do not prompt)** check box.
5. Click **Finish**.

   You might receive an error message indicating the project failed to publish because the server is not started. To start the server, select the **Server** tab, select the **WebSphere Application Server v6.0** server with **host name** of **localhost**, and click the start icon ▶ (see Figure 61).



Server	Host name	Status	State
WebSphere Application Server v6.0	localhost	Stopped	Synchronized

Figure 61. Starting the server

The WebSphere Application Server test environment starts. After a few moments, the browser opens in the upper right view area of the workbench. Be patient. It takes time for the application server to start on your workstation.

The minimum memory requirement for running the WebSphere Application Server test environment is 768 megabytes according to the IBM announcement letter for WebSphere Development Studio Client Version 6. IBM recommends 1 gigabyte or more of main memory. It take more time for the application server to start if you are working on a workstation that has less than the required memory. The system speed also have an impact on the application server start up time. Eventually you see the index.jsp page that has been generated by the IBM WebFacing Tool opening in the workbench's built-in browser.

Here you see the browser inside the workbench with the index.jsp page displayed (see Figure 62).

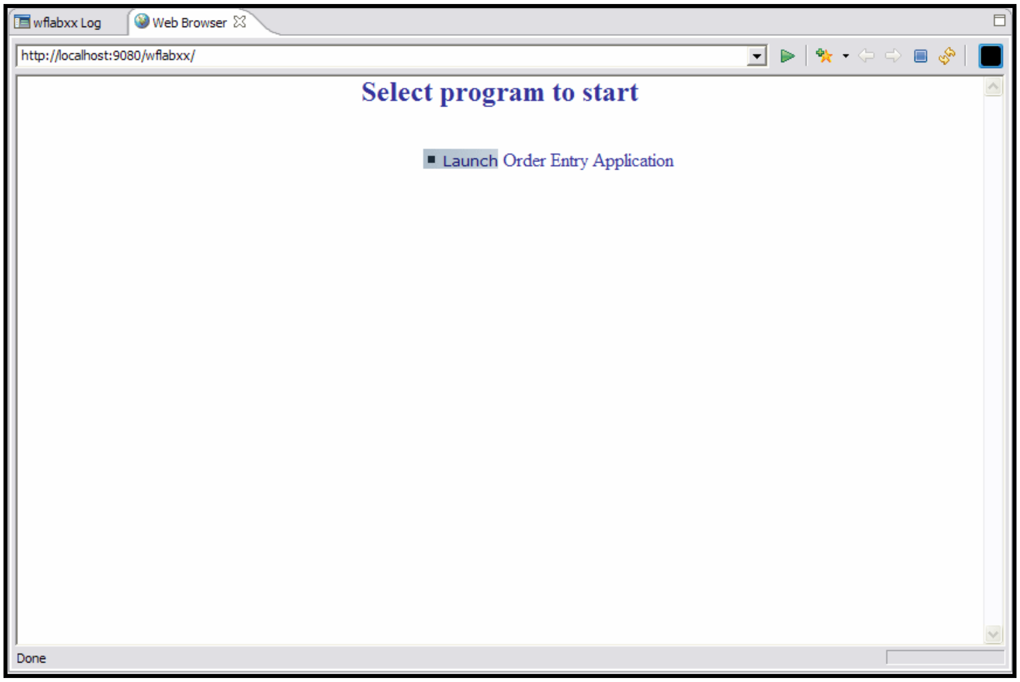

wflabxx Log
Web Browser
http://localhost:9080/wflabxx/
Select program to start
Launch Order Entry Application
Done

Figure 62. index.jsp page displayed

Tip: To enlarge the browser window to the size of the workbench window, double-click the browser title bar. To get it back to its original size in the upper right-hand workbench view area, double-click the enlarged browser window, title bar.

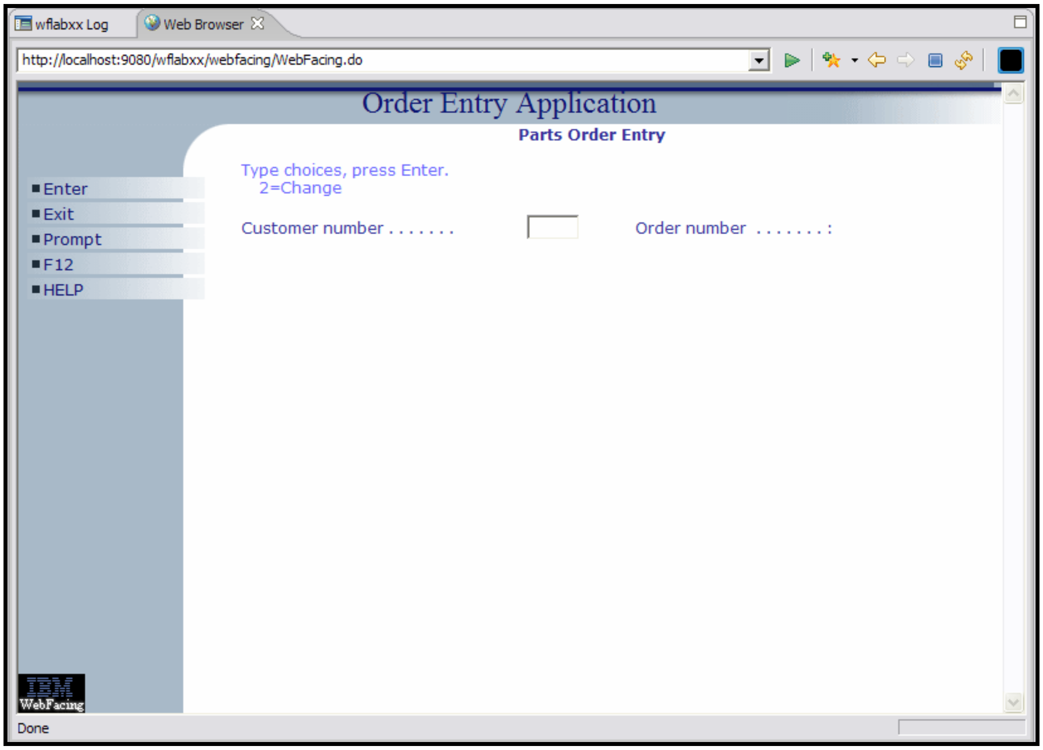
You have a choice of running WebSphere Application Server V5.1 or V6.0.1 in the workbench test environment. In this tutorial, you use WebSphere Application Server V6.0.1 in the test environment and J2EE level 1.3.

Exercise 4.2: Testing the WebFacing application

To run the Web application:

1. In your browser pane, click the **Launch** link.

   After a few moments, the first application page opens in the browser (see Figure 63).



http://localhost:9080/wflabxx/webfacing/WebFacing.do

Order Entry Application

Parts Order Entry

Type choices, press Enter.
2=Change

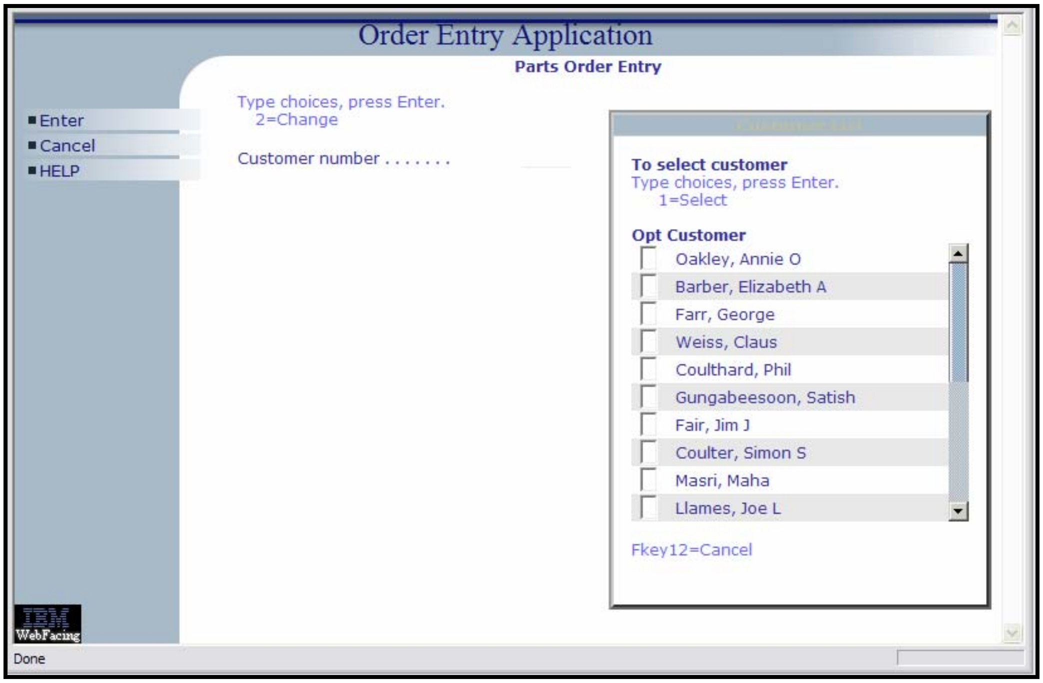
Customer number . . . .   Order number . . . . :

Figure 63. Order Entry Application page opens

If you find the response time slow, keep in mind that the first time the JSP file is requested, it has to be compiled into a servlet. If you run the application a second time, you see improved performance because the servlet already exists. This is normal application server behavior.

Now you want you to go through the same steps as in “Reviewing the 5250 order entry application” to review the results of the WebFacing conversion. This gives you a sense of how the default conversion changes your original application. You then have a chance to improve the conversion results.

2. Click the **Prompt** button, or press command key **F4** to open the selection list window (see Figure 64).



Order Entry Application

Parts Order Entry

Type choices, press Enter.
  2=Change

Customer number . . . .

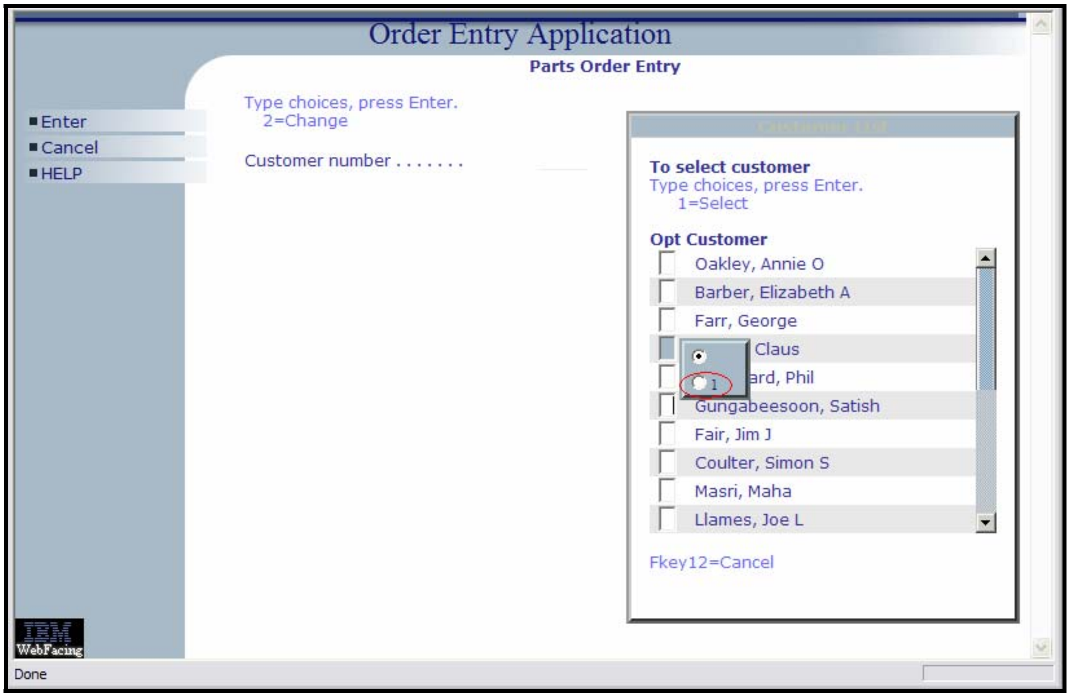
To select customer
Type choices, press Enter.
  1=Select

Opt Customer
□ Oakley, Annie O
□ Barber, Elizabeth A
□ Farr, George
□ Weiss, Claus
□ Coulthard, Phil
□ Gungabeesoon, Satish
□ Fair, Jim J
□ Coulter, Simon S
□ Masri, Maha
□ Llames, Joe L

Fkey12=Cancel

Figure 64. Open the selection list window.

3. Select a customer from the list. (This is your converted subfile.) As you move your cursor over each input field in the customer list, a floating panel with two radio buttons appear. One radio button is unlabeled, and the other is labeled **1**. Position your cursor to the input field left of name you wish to select, then click the radio button labeled **1.** Alternatively, you can select a customer by typing `1` into its **Opt** input field (see Figure 65).



Order Entry Application

Parts Order Entry

- Enter
- Cancel
- HELP

Type choices, press Enter.
  2=Change

Customer number . . . .

Customer List

To select customer
Type choices, press Enter.
  1=Select

Opt Customer

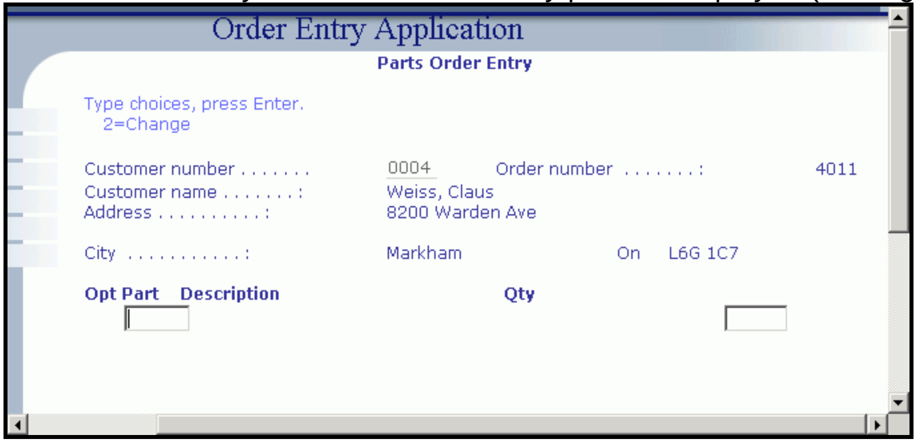
□ Oakley, Annie O
□ Barber, Elizabeth A
□ Farr, George
□ ● Claus
□ ○ 1 ard, Phil
□ Gungabeesoon, Satish
□ Fair, Jim J
□ Coulter, Simon S
□ Masri, Maha
□ Llames, Joe L

Fkey12=Cancel

Done

Figure 65. Select a customer from the list

4. Press the **Enter** key. The Parts Order Entry panel is displayed (see Figure 66).



Order Entry Application

Parts Order Entry

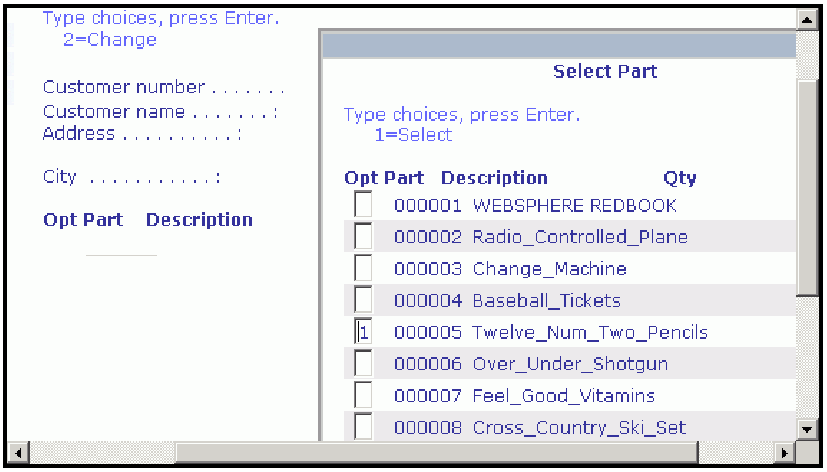
Type choices, press Enter.
  2=Change

Customer number   0004   Order number . . . . :   4011
Customer name . . . . :   Weiss, Claus
Address . . . . :   8200 Warden Ave
City . . . . :   Markham   On   L6G 1C7

Opt Part   Description   Qty

Figure 66. Parts Order Entry panel displayed

5. Click the **Prompt** button, or press command key **F4** to display the parts selection list (see Figure 67).



Type choices, press Enter.
2=Change

Customer number . . . .
Customer name . . . . :
Address . . . . :

City . . . . :

Opt Part Description

Select Part

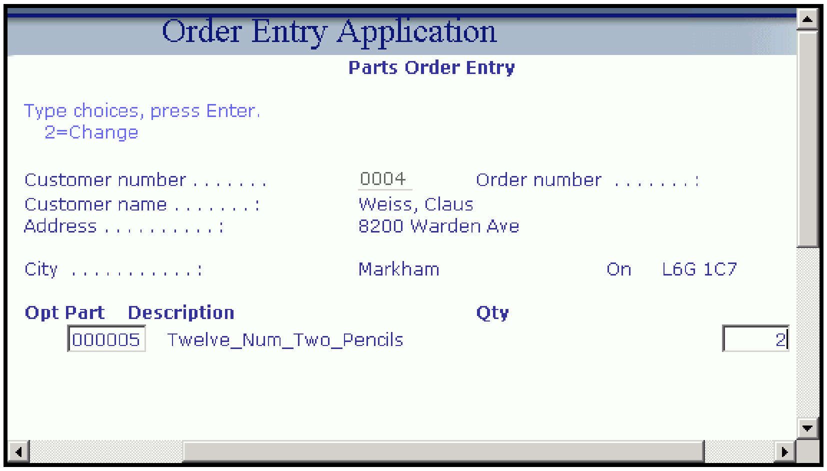
Type choices, press Enter.
1=Select

Opt	Part	Description	Qty
□	000001	WEBSPHERE REDBOOK	
□	000002	Radio_Controlled_Plane	
□	000003	Change_Machine	
□	000004	Baseball_Tickets	
1	000005	Twelve_Num_Two_Pencils	
□	000006	Over_Under_Shotgun	
□	000007	Feel_Good_Vitamins	
□	000008	Cross_Country_Ski_Set	

Figure 67. Display parts selection list

6. Select a part by typing 1 into its **Opt** input field.
7. Press the **Enter** key.

   The Parts Order Entry window opens (see Figure 68).



Order Entry Application

Parts Order Entry

Type choices, press Enter.
2=Change

Customer number . . . . 0004 Order number . . . . :
Customer name . . . . : Weiss, Claus
Address . . . . : 8200 Warden Ave

City . . . . : Markham On L6G 1C7

Opt	Part	Description	Qty
	000005	Twelve_Num_Two_Pencils	2

Figure 68. Parts Oder Entry window opens

8. Enter a quantity.
9. Press the **Enter** key.
10. Continue ordering one or two more parts.
11. Press command key **F6** to accept the order.

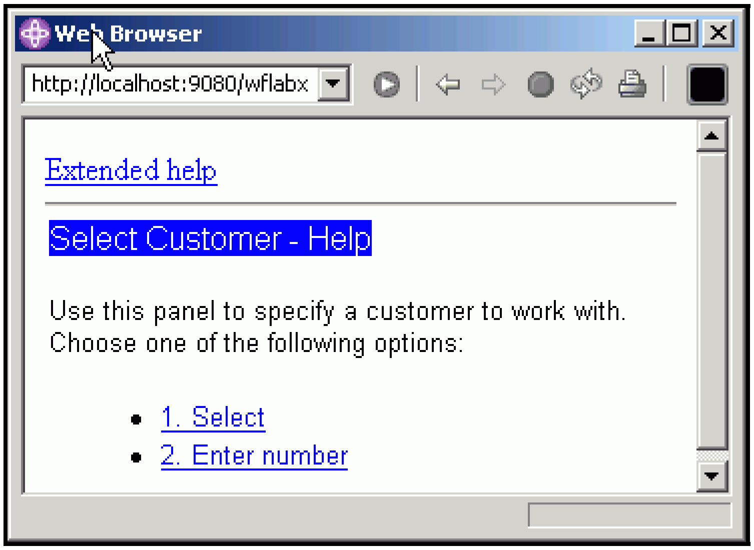
    The application works the same way as before in a 5250 environment, but the application now has a browser user interface.

Exercise 4.3: Testing the help support

To test the help:

1. Press the **F1** key.

   Another browser window opens with the help information (see Figure 69).

   Extended help

   Select Customer - Help

   Use this panel to specify a customer to work with.
   Choose one of the following options:

   - 1. Select
   - 2. Enter number

   Figure 69. Browser window opens with help information

2. Now try the links:

   - 1. Select
   - 2. Enter number
   - Extended help

   Your UIM based help panels are all available in the WebFacing application.

3. Close the help browser window and press **F3** to end the application.

Exercise 4.4: Bypass login screen

In certain circumstances, it might be desirable to bypass the WebFacing login screen. WebFacing applications can be invoked programmatically from other Web applications. This provides a way to integrate WebFacing generated user interfaces with existing Web applications.

WebFacing applications are launched from URLs. Typically, a URL is represented as a link that a user clicks to start the application. During a WebFacing conversion, URLs are written to an index.jsp file and, after the application is deployed, users select these links to start the application.

Other programs, such as controller servlets, can also construct WebFacing URLs dynamically. These URLs can be sent back to users as clickable links on a Web page. However, more importantly, they can be used in the forward() and sendRedirect() methods in other Web applications. Using dynamically determined WebFacing URLs with the forward() and sendRedirect() methods mean that a program like a controller servlet can perform actions such as: determining the host to use, the WebFacing program to launch, what CL command to use for the WebFacing program, or other actions. After these actions are completed, the servlet can provide the appropriate WebFacing application, with its parameters already set, directly to a user.

Here is a simple example for determining the CL command to use to launch a program:

URL constructed by a controller servlet:

webfacing/services/invocation/WFInvocation.do?clcmd=call%20ordentr

In this example, **ordentr** is the name of a program to launch. The value **ordentr** can be determined by a servlet and assigned to a variable such as *iseriesProgram*. Your servlet can construct the URL string using the value determined for *iseriesProgram* and assign it to a variable *newURL* using a line like:

newURL = "webfacing/services/invocation/WFInvocation.do?clcmd=call " + iseriesProgram;

newURL can then be used as the forward or redirect URL for your forward() or sendRedirect() methods.

In this example, the complete URL used by the browser, if sent as a redirect, looks similar to:

http://<hostname>:<port>/<application>/webfacing/services/invocation/WFInvocation.do?clcmd=call%20ordentr

The example shows the full URL beginning with: http://<hostname>:<port>/<application>/.

The value for *newURL* is the string after this. That is, the string: webfacing/services/invocation/WFInvocation.do?clcmd=call%20ordentr.

In an example like this, the first part of the URL,http://<hostname>:<port>/<application>/, represents the host, port, and context root for the application. If your controller servlet is in the same context root, it is not always required for the servlet to determine the entire URL. If necessary, though, you can code the servlet to construct a string for the fully qualified URL.

> **Note:** The characters **%20** in the URL represent a space character as encoded when sent to a browser. Space characters generally cannot be used explicitly in URLs. In the example where the URL string is being constructed and assigned to *newURL*, the space is present in the part of the string just after clcmd=call. The reason for the space in the string is that the example represents a CL command call **ordentr**. In the URL string that is being constructed, it is not necessary to add **%20** directly. The server adds this encoding if necessary.

URL parameters that can be determined dynamically

clcmd: CL command to launch the program.

host: Host name where the original 5250 application is located.

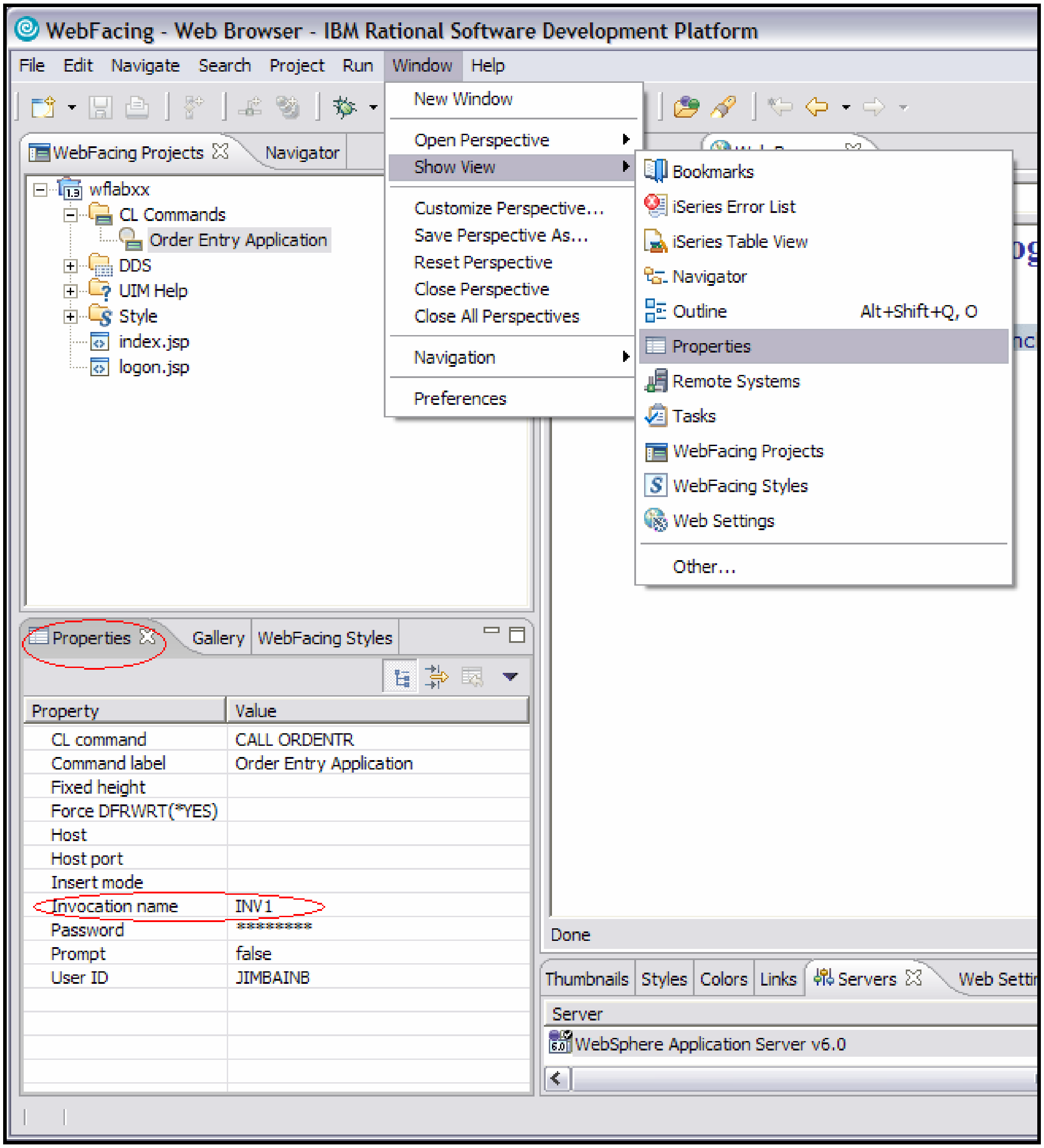
User ID: User ID is used to log onto the application.

Note: If a forward() method is used in your controller servlet, URL parameters are only sent within the application server tier (middle tier). Using sendRedirect() instead exposes URL parameters to the browser. In this way, sendRedirect() is less secure because information such as User IDs and passwords can be revealed in a browser's location field or if a user views the properties for the page they are using.

password: Password used to log onto the application.

Note: If a forward() method is used in your controller servlet, URL parameters are only sent within the application server tier (middle tier). Using sendRedirect() instead exposes URL parameters to the browser. In this way, sendRedirect() is less secure because information such as User IDs and passwords can be revealed in a browser's location field or if a user views the properties for the page they are using.

inv: The invocation name for the WebFacing CL command used to launch the application. If values such as host, User ID and password are defined for a CL command, these values override the general values specified for a project. To view the invocation name for a CL command, open the WebFacing perspective in the workbench, click the **WebFacing Projects** tab, expand your WebFacing project, expand the **CL Commands** folder, and click the label for the command. The value for the invocation name can be viewed in the **Properties** pane. If the Properties pane is not displayed in the WebFacing perspective, to open it, click **Window** > **Show View** > **Properties** (see Figure 70).



Property	Value
CL command	CALL ORDENTR
Command label	Order Entry Application
Fixed height	
Force DFRWRT(*YES)	
Host	
Host port	
Insert mode	
Invocation name	INV1
Password	*********
Prompt	false
User ID	JIMBAINB

Figure 70. Properties pane

Recap

You have completed “Running the WebFacing application.” You now understand how to:

- Set up the WebSphere test environment
- Start the application in the WebSphere test environment
- Select a link on the index.jsp page to start the WebFacing application
- Test the WebFacing application as you did in “Reviewing the 5250 order entry application”
- Understand how to bypass the log-in screen of a WebFacing application by calling it directly

Next, you improve the user interface generated by the IBM WebFacing Tool by improving the customer selection, subfile interface. The interface for selecting a record from a subfile does not operate like a typical Web application. To select a record, one must first key **1** into the options column; this default behavior is a vestige of how the green-screen application works. You need to create a Web user interface that selects a customer simply by clicking on a link.

We also guide you through the Web Setting capabilities of CODE Designer to show you how to make the necessary modifications. You change the customer name column in the customer selection subfile from a simple text output field to a link. The information expected by the 5250 program is sent when you click this new customer name link. The information that the 5250 program is expecting is a **1** in the subfile options field for the selected customer record. The 5250 program still receives the same subfile selection information even through you change the user interface.

The WebFacing conversion adds logic to the user interface, so that the options field is filled with a **1** and the **Enter** key event is invoked when the link is pressed. With WebFacing, you can enhance the user interface without changing the existing 5250 application.

Continue to “Changing the user interface.”

Changing the user interface

To enhance the results of the IBM WebFacing Tool, you use the feature Web Settings in the CODE Designer tool. CODE Designer is the GUI-based screen design tool for 5250 panels. It allows you to change certain aspects of the user interface related to the WebFacing conversion process. CODE Designer stores the Web Setting information as comment lines in the DDS source. When you convert the application using the IBM WebFacing Tool, these DDS source comment lines are picked up and applied to the new user interface.

In this chapter, you learn how to change the user interface of your WebFacing application with CODE Designer. In the WebFacing perspective you learn how to select a DDS source member to work with, start CODE Designer to open this DDS member, work with Web Settings in CODE Designer to change the Web look, and then reconvert the DDS member.

To accomplish these learning objects, several steps are involved, including:

- Exercise 5.1: Opening DDS display file source
- Exercise 5.2: Applying Web Settings
- Exercise 5.3: Testing the changed application

The exercises within each chapter must be completed in order. Start with “Exercise 5.1: Opening DDS display file source” when you are ready to begin.

Length of time

This chapter takes approximately 15 minutes to complete.

Exercise 5.1: Opening DDS display file source

Because you use the WebFacing perspective to invoke CODE Designer, you start this exercise using WebSphere Development Studio Client in the WebFacing perspective.

Starting WebSphere Development Studio Client and opening the Web perspective

Start WebSphere Development Studio Client and open the Web perspective:

1. Start WebSphere Development Studio Client and open the WebFacing perspective, if it is not up and running already.
2. In the WebFacing perspective, make sure that you are in the WebFacing Projects view and not the Navigator view.
3. Expand your project in the WebFacing Projects view by clicking the plus sign (**+**) beside the WebFacing project icon.

Starting CODE Designer

To start CODE Designer, do the following:

1. Expand the DDS folder.
2. Right-click **SLTCUSTD** source member inside the DDS folder.
3. Click **Open With > CODE Designer** on the pop-up menu (see Figure 71).

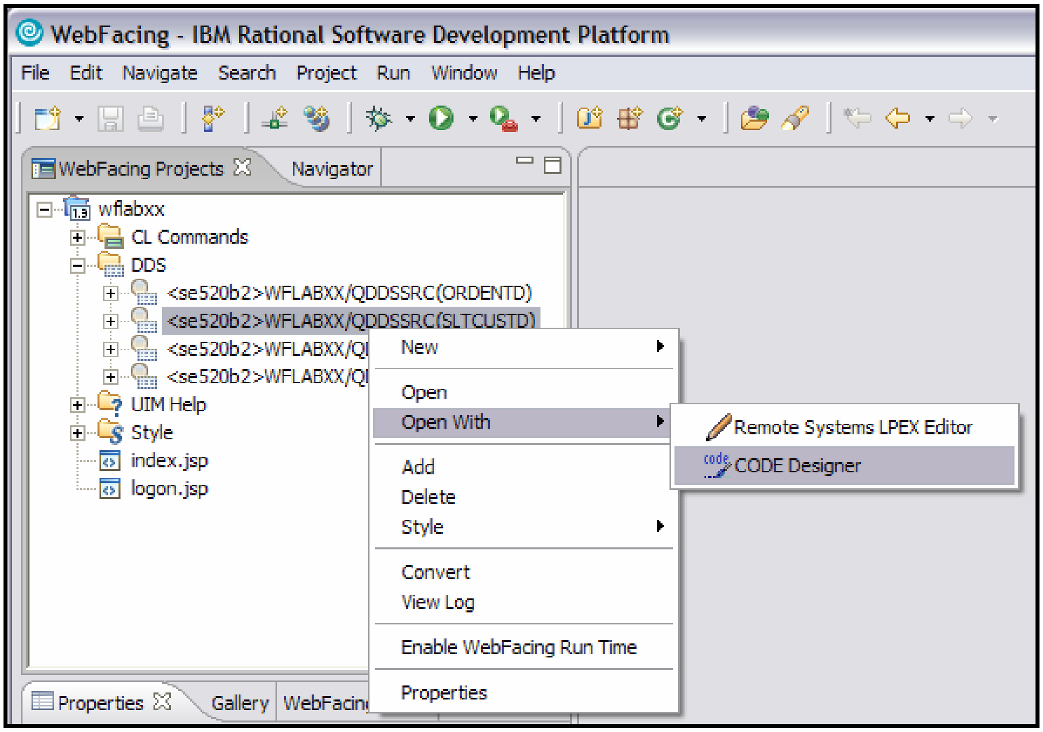
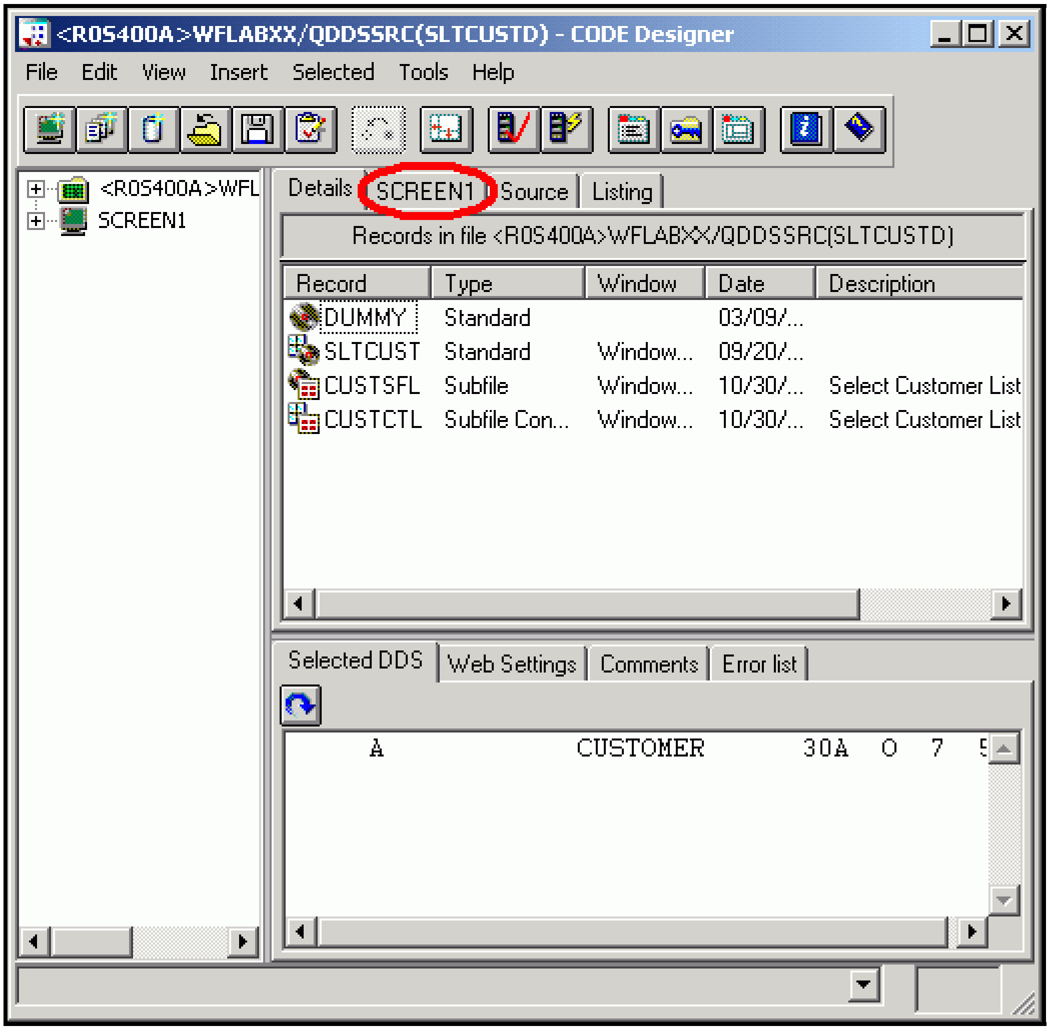

WebFacing - IBM Rational Software Development Platform
File Edit Navigate Search Project Run Window Help
WebFacing Projects
Navigator
wflabxx
CL Commands
DDS
<se520b2>WFLABXX/QDDSSRC(ORDENTD)
<se520b2>WFLABXX/QDDSSRC(SLTCUSTD)
<se520b2>WFLABXX/QI
<se520b2>WFLABXX/QI
UIM Help
Style
index.jsp
logon.jsp
New
Open
Open With
Add
Delete
Style
Convert
View Log
Enable WebFacing Run Time
Properties
Remote Systems LPEX Editor
CODE Designer
Properties
Gallery
WebFacin

*Figure 71. Starting **CODE Designer***

The DDS member opens in CODE Designer. This takes a moment to complete (see Figure 72).



Records in file <ROS400A>WFLABXX/QDDSSRC(SLTCUSTD)

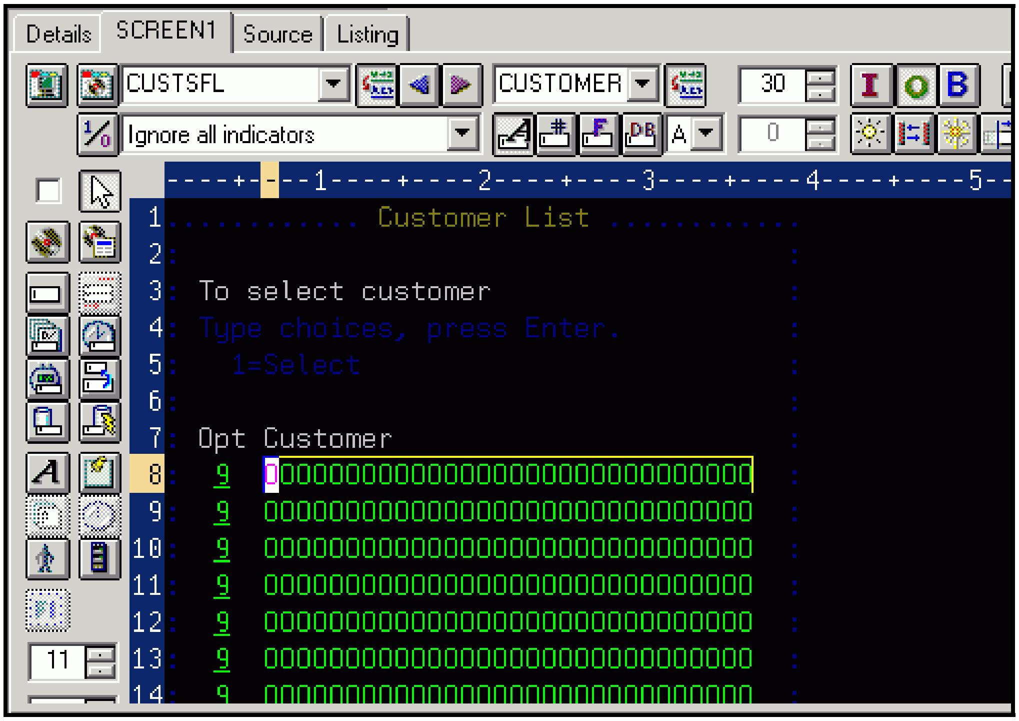
Record	Type	Window	Date	Description
DUMMY	Standard		03/09/...	
SLTCUST	Standard	Window...	09/20/...	
CUSTSFL	Subfile	Window...	10/30/...	Select Customer List
CUSTCTL	Subfile Con...	Window...	10/30/...	Select Customer List

```
     A                    CUSTOMER        30A  O  7  5
```

Figure 72. The DDS member opens in CODE Designer

4. Select the **SCREEN1** tab on the notebook.

   You see a similar dialog to the following in Figure 73):



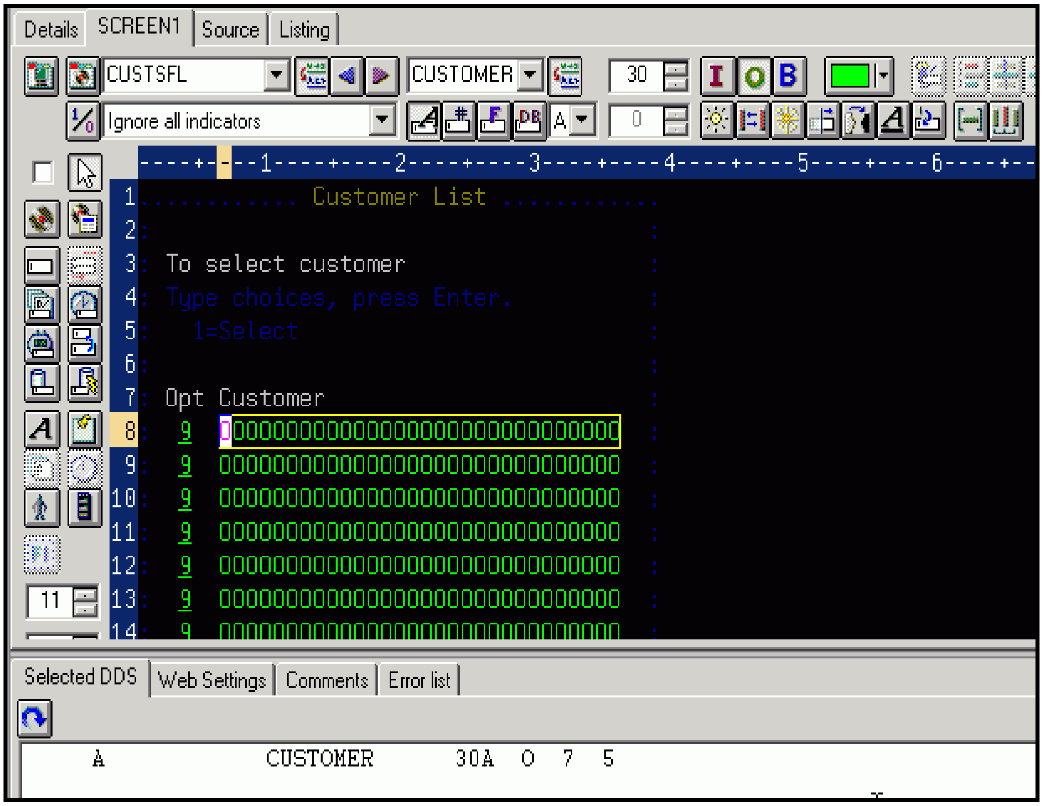
CUSTSFL
CUSTOMER
30
Ignore all indicators
0
11

```
----+----1----+----2----+----3----+----4----+----5--
 1............ Customer List ............
 2:                                      :
 3:  To select customer                  :
 4:  Type choices, press Enter.          :
 5:    1=Select                          :
 6:                                      :
 7:  Opt Customer                        :
 8:   9  000000000000000000000000000000  :
 9:   9  000000000000000000000000000000  :
10:   9  000000000000000000000000000000  :
11:   9  000000000000000000000000000000  :
12:   9  000000000000000000000000000000  :
13:   9  000000000000000000000000000000  :
14:   9  000000000000000000000000000000  :
```

Figure 73. Dialog in green-screen image in CODE Designer

As you see, you are working with the green-screen image in CODE Designer. The screen might be recognizable. It is the customer selection list. This is the screen you change, therefore, it appears more Web-like to the user after the IBM WebFacing Tool runs.

5. Click the second column of the first row in the subfile (the **Customer** field) (see Figure 74.



```
----+--1----+----2----+----3----+----4----+----5----+----6----+--
 1 ........... Customer List ...........
 2:                                        :
 3: To select customer                     :
 4: Type choices, press Enter.             :
 5:   1=Select                             :
 6:                                        :
 7: Opt Customer                           :
 8:  9  000000000000000000000000000000     :
 9:  9  000000000000000000000000000000     :
10:  9  000000000000000000000000000000     :
11:  9  000000000000000000000000000000     :
12:  9  000000000000000000000000000000     :
13:  9  000000000000000000000000000000     :
14:  9  000000000000000000000000000000     :
```

```
     A            CUSTOMER      30A   O   7   5
```

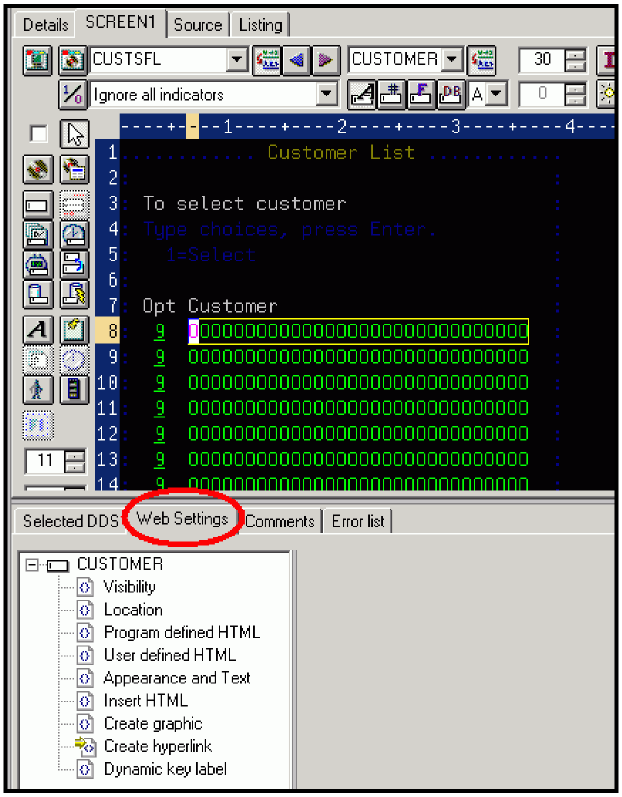
*Figure 74. Click the second column of the first row in the subfile (the **Customer** field).*

Exercise 5.2: Applying Web Settings

Now work with the Notebook underneath the Design page.

To apply Web Settings:

1. Locate the **Web Settings** tab (see Figure 75).



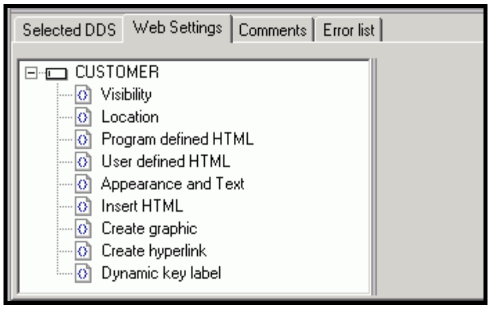
```
----+----1----+----2----+----3----+----4---
 1 ........... Customer List ...........
 2                                        :
 3  To select customer                    :
 4  Type choices, press Enter.            :
 5    1=Select                            :
 6                                        :
 7  Opt Customer                          :
 8   9  000000000000000000000000000000    :
 9   9  000000000000000000000000000000    :
10   9  000000000000000000000000000000    :
11   9  000000000000000000000000000000    :
12   9  000000000000000000000000000000    :
13   9  000000000000000000000000000000    :
14   9  000000000000000000000000000000    :
```

- CUSTOMER
  - Visibility
  - Location
  - Program defined HTML
  - User defined HTML
  - Appearance and Text
  - Insert HTML
  - Create graphic
  - Create hyperlink
  - Dynamic key label

*Figure 75. Locate **Web Settings** tab*

2. Click the **Web Settings** tab.

The Web Settings page in the CODE Designer window opens (see Figure 76).

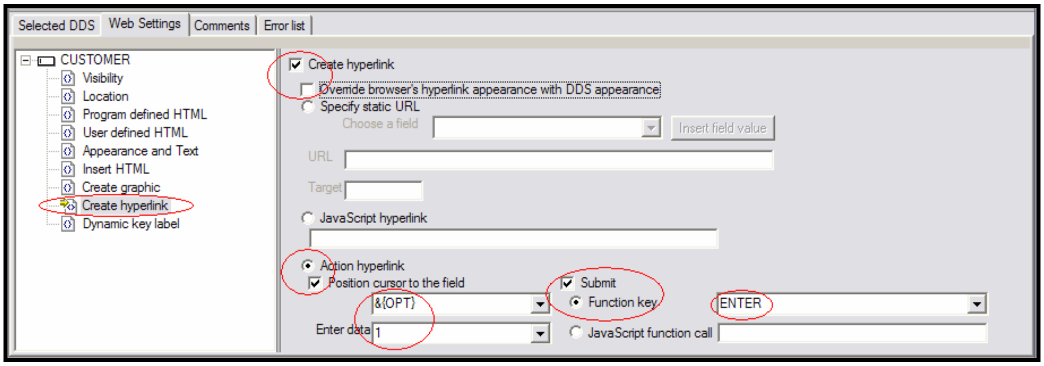


- CUSTOMER
  - Visibility
  - Location
  - Program defined HTML
  - User defined HTML
  - Appearance and Text
  - Insert HTML
  - Create graphic
  - Create hyperlink
  - Dynamic key label

*Figure 76. **Web Settings** page opens*

Steps 3 through 14 apply to Figure 77.

3. On the Web Settings page, locate the list that shows the different Web Settings available for the **Customer** field.
4. If not all Web Settings show, scroll down to the bottom of the list.
5. Click **Create hyperlink** in the **Web Settings** list.



☑ Create hyperlink
□ Override browser's hyperlink appearance with DDS appearance
○ Specify static URL
○ JavaScript hyperlink
● Action hyperlink
☑ Position cursor to the field: &{OPT}
Enter data: 1
☑ Submit
● Function key: ENTER
○ JavaScript function call

*Figure 77. Click **Create hyperlink** in the Web Settings list*

6. Select the **Create hyperlink** check box to the right of the list.
7. Uncheck the **Override browser's hyperlink appearance with DDS appearance** check box.
8. Click the **Action hyperlink** radio button.
9. Leave the **Position cursor to field** check box selected.
10. From the list under the **Position cursor to field**, select the **&{OPT}** value.
11. In the **Enter data** list, type **1**.

    This indicates that you want a **1** to be entered in the **Options** field when the hyperlink is clicked.

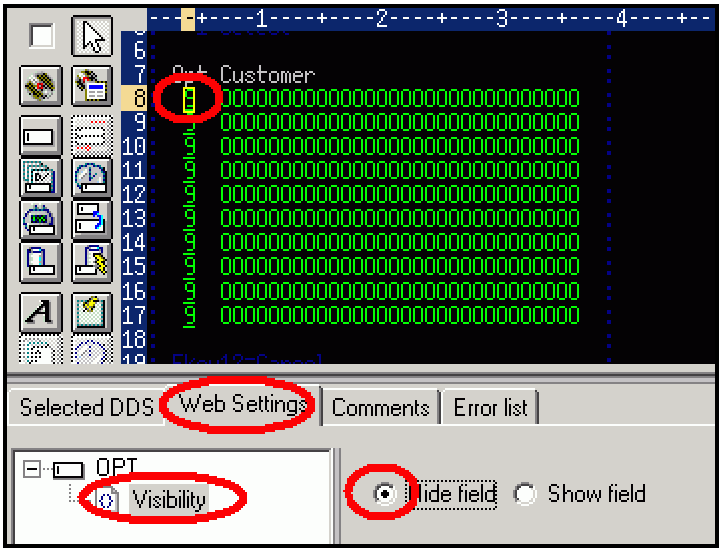
12. Select the **Submit** check box.
13. Leave the **Function key** radio button selected.
14. Select **ENTER** from the **Function key** list.

    Just to recap, you specified the following. When the application is converted with the IBM WebFacing Tool, the customer name cells in the subfile are generated so they appear as links in the browser window. At run time, when the link is clicked, a **1** is placed in the Option field. Also, a submit request is initiated to generate the equivalent of pressing the Enter key and the cursor is positioned in the **Option** field for the selected row.

    You have accomplished the basic task, but the user interface now has an Option column that does not belong there because it is useless with the new link. Also, the instructions on the page on how to select a customer are incorrect. You must fix this with Web Settings prior to testing this new feature.

Hiding the option field

On the Design page, select the option (**Opt**) field to indicate that you want to work with it (see Figure 78).



● Hide field ○ Show field

*Figure 78. Select the **Opt** field to work with it.*

To hide the option heading:

1. Click the **Opt** column on the first record in the subfile to select it.
2. On the Web Settings page, select **Visibility** from the list.
3. Select the **Hide field** radio button.

   Now this column in the subfile is hidden.

Hiding the option heading for the option field

You need to hide the heading for the option field as well. The heading is not in the subfile record; it is located in the subfile control record. Therefore, you need to give focus to a different record format in the design page. What this means is that it looks like one record format, but there are three record formats in this view.

For an example, look at some of the CODE Designer features. Normally in the System i green-screen world, screens are created out of multiple record formats that, at run time, are assembled by the RPG or COBOL program to present the real screen the user sees.

To make it easier for you at design time to get a feel of what the record formats (you create) look like at run time, CODE Designer has added the notion of a group. You can assemble several record formats in a group. This feature gives you the capability of mimicking what happens at run time with your record formats. Therefore, a group resembles the runtime grouping of record formats for you at development time. In the sample you are working with, three record formats are already assembled in group SCREEN1. The record formats are:

- CUSTSFL
- CUSTCTL
- SLTCUST

To get a list of record formats belonging to a group, you can expand the list of record formats (see Figure 79).

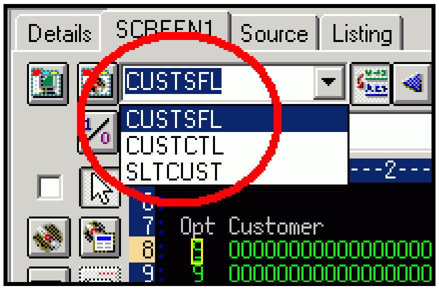


Figure 79. List of record formats

Only one record format in a group is active at a time. The record format name is listed in the top field of the record list. Also, in the design page, all parts in the active record format are shown highlighted but the inactive record format content is half-intensity. You see this effect clearly above. Until now, you have been working with the subfile record format CUSTSFL only, but now you need to clean up the subfile control record format CUSTCTL, so you have to make it the active record format. Another way to describe this would be to give it focus.

Hiding the option heading

To hide the option heading Opt Customer:

1. To shift focus to another record format, at the top of the Design page, click the arrow (see Figure 80).

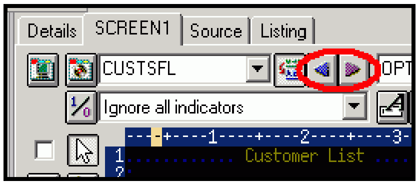


Figure 80. Choosing a different record format

Notice that the top area of the Design page is now highlighted, and the bottom one has only half-intensity. As mentioned before the highlighted area is the one you can work with when performing this step.

Next, you clean up the subfile control record format (see Figure 81).

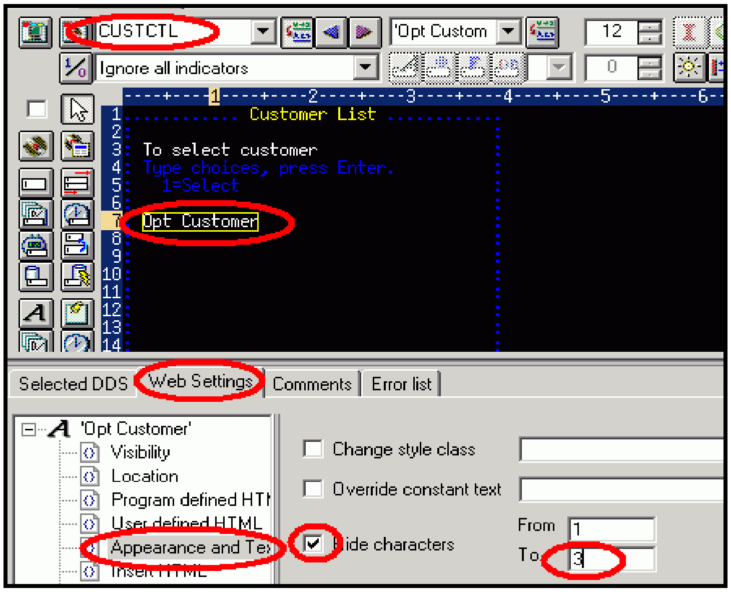


Figure 81. Clean up subfile control record format

2. On the Design page, select the **OPT Customer** constant.
3. On the Web Settings page, select **Appearance and Text** from the list.
4. Select the **Hide characters** check box.
5. Leave **1** in the **From** field.
6. Type **3** in the **To** field.

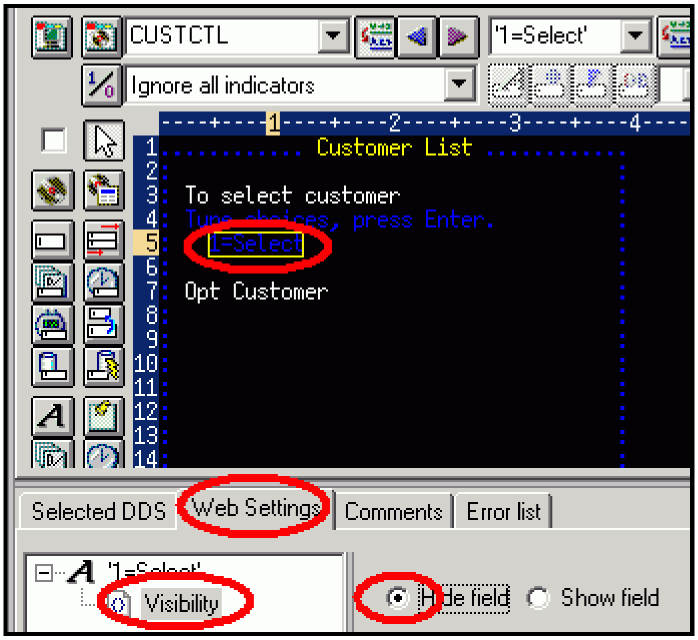
This hides the heading Opt. Now, you need to remove the instructions on the panel that guide the user to put a **1** in the **Option** field to select a specific customer.

Hiding the select instruction

You do not have to give focus to a different record format because this constant is also located on the CUSTCTL subfile control record format.

To hide the select instruction:

1. Select the constant **1=Select** in the Design page (see Figure 82).



```
                Customer List
 To select customer
 Type choices, press Enter.
 1=Select

 Opt Customer
```

'1=Select'
Visibility
● Hide field ○ Show field

Figure 82. Hide the select instruction.

2. On the Web Settings page, select **Visibility** from the list.
3. Click the **Hide field** radio button.

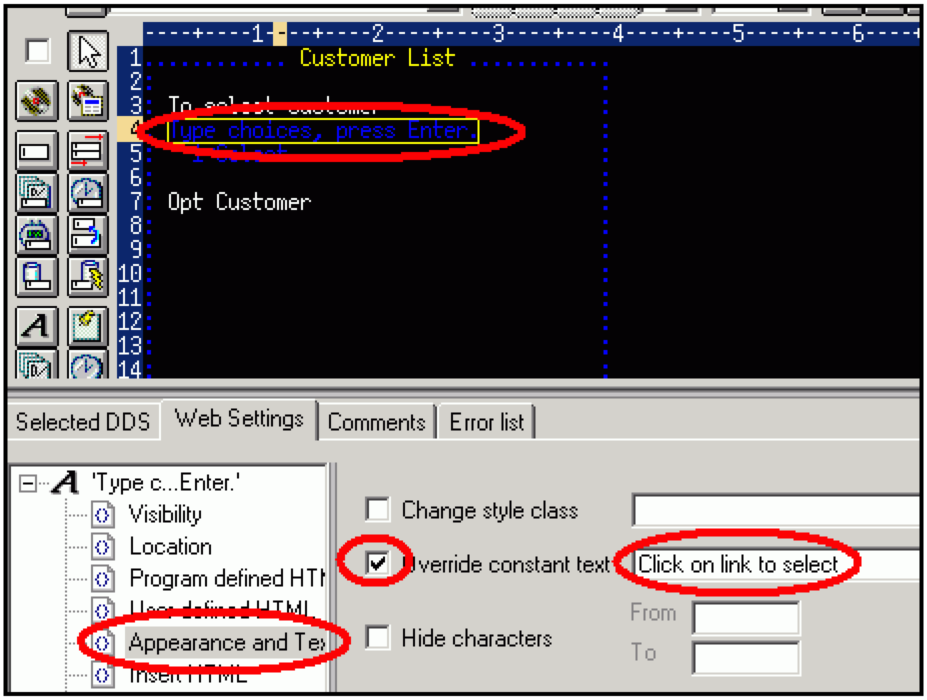
Changing the instruction (Type choices, press Enter)

Now you need to add new instructions for the WebFacing page, so the user knows to click the link to select a customer.

Note: You cannot just change the 5250 constant because you might still use this screen in a 5250 environment. You have to use Web Settings to apply this change to the Web user interface only.

To change the instruction text:

1. Select the instruction constant **(Type choices, press Enter)** on the Design page (see Figure 83).



☑ Override constant text: Click on link to select

*Figure 83. Choose **Type choices, press Enter.***

2. On the Web Settings page, select **Appearance and Text** from the list.
3. Select the **Override constant text** check box, and type the new text `Click on link to select` in the field to the right of the check box.

   Now all Web Settings are in place and you can go ahead and save the DDS source and reconvert this DDS member.

Closing CODE Designer and reconverting DDS

To close CODE Designer and re-convert the DDS member:

1. Click the **X** button at the top right corner of CODE Designer (see Figure 84).

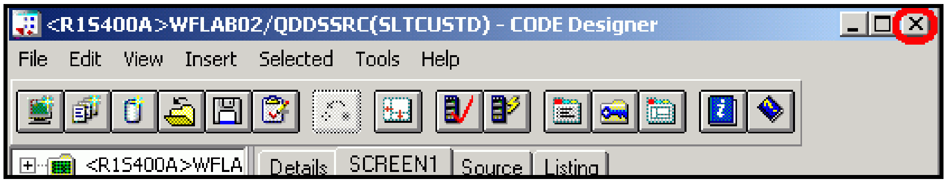

<R15400A>WFLAB02/QDDSSRC(SLTCUSTD) - CODE Designer
File Edit View Insert Selected Tools Help
<R15400A>WFLA
Details
SCREEN1
Source
Listing

*Figure 84. Click the **X** button at the top right corner of CODE Designer.*

2. When asked to do so, click **Yes** to save to return to the WebFacing perspective in the WebSphere Development Studio Client workbench.
3. Select your project **wflabxx** in the WebFacing Projects view.
4. Expand the project by clicking on the plus sign (**+**) in front of its icon, if you cannot see the DDS folder as part of the project.
5. Expand the DDS folder, if it is not expanded already, to show all members in this folder (see Figure 85).

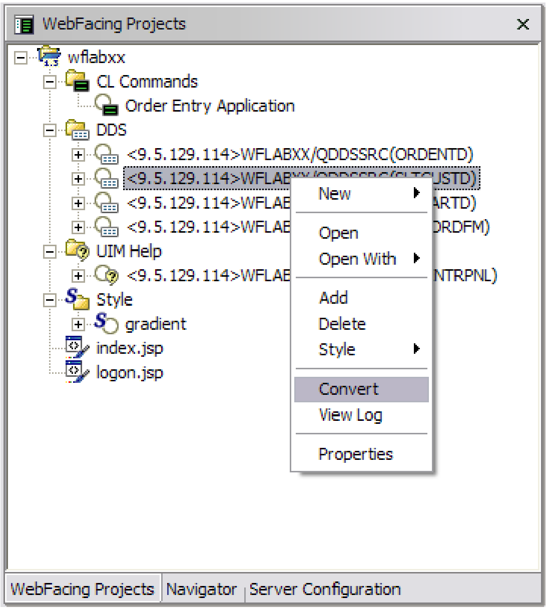


Figure 85. Expand the DDS folder.

6. Select the member **SLTCUSTD** that you just changed.
7. Right-click **SLTCUSTD** member icon.
8. Click **Convert** on the pop-up menu.

Only this member is converted. A conversion report opens after the conversion is finished. Now you test the new user interface by returning to the WebFacing perspective in the WebSphere Development Studio Client workbench.

Exercise 5.3: Testing the changed application

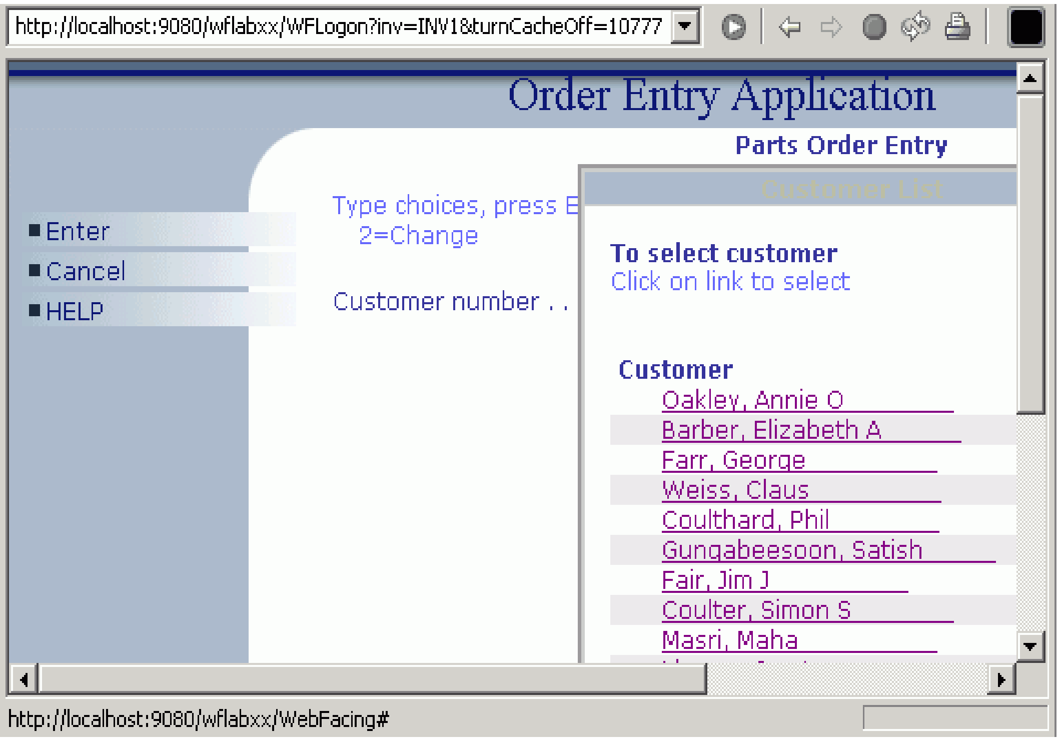
Look at the new user interface in your application.

To test the changed application:

1. Right-click the **wflabxx** project icon in the WebFacing Projects view.
2. Click **Run on Server** on the pop-up menu.
3. Go to the browser pane and click the **Launch** link.

   You see the first screen of your application. Nothing has changed there.

4. Click the **Prompt** push button, or press command key **F4** (see Figure 86).



http://localhost:9080/wflabxx/WFLogon?inv=INV1&turnCacheOff=10777

Order Entry Application

Parts Order Entry

Customer List

Type choices, press E
2=Change

Customer number . .

To select customer
Click on link to select

Customer
Oakley, Annie O
Barber, Elizabeth A
Farr, George
Weiss, Claus
Coulthard, Phil
Gungabeesoon, Satish
Fair, Jim J
Coulter, Simon S
Masri, Maha

http://localhost:9080/wflabxx/WebFacing#

*Figure 86. Click the **Prompt** push button, or press command key **F4**.*

The customer selection window appears and the customer fields are shown as links. Also, notice the instructional text has changed and that the option column is no longer visible.

5. Select a customer from the list by clicking on the link.
6. Go through the same steps as in “Reviewing the 5250 order entry application” to test the changed application.

Recap

You have completed “Changing the user interface.” You now understand how to:

- Access a DDS source member using CODE Designer
- Change the Web user interface using Web Settings in CODE Designer
- Convert the changed DDS
- Test the changed application
- Allow user to change text size in the browser

Note: The green-screen user interface has not changed if the application is invoked as a green-screen application. It still has the original user interface. The Web Settings apply only to the WebFacing user interface.

Suppose that when you tested the changed application, you found that you did not like the way the IBM WebFacing Tool mapped some of the display attributes to colors and text styles available in the browser. You want to change the default WebFacing style rules that determine how certain 5250 parts are displayed in a browser to give the Web user interface a better look. You learn to make these changes in the next exercise.

Continue to “Changing the style of the Web user interface.”

Changing the style of the Web user interface

In this chapter, you learn how to change the user interface of your WebFacing application by using the WebFacing style properties dialog. The style changes are applied on a project level; so all pages in a project are displayed with these changes applied. In this chapter, you change the text color and font for all highlighted fields. You also locate the style file where these changes have been stored.

In this chapter, you learn how to change a style to tailor the Web interface to your needs. You start the Style Properties dialog, create a new style, change style settings, specify the new style to be used for this project, and refresh the project with the new style.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 6.1: Changing the rules for highlighted fields
- Exercise 6.2: Changing styles directly
- Exercise 6.3: Allowing the user to select text size

The exercises within each chapter must be completed in order. Start with Exercise 6.1: Changing the rules for highlighted fields when you are ready to begin.

Length of time: This chapter takes approximately 10 minutes to complete.

Exercise 6.1: Changing the rules for highlighted fields

During this exercise, you use the Style Properties dialog in the IBM WebFacing Tool to tailor your Web-user interface. You change the rules for highlighted fields in the 5250 panels so they appear more visible in the Web-user interface. The style for highlighted fields is altered so at run time, the text font is different and a larger font size is applied.

To change the rules for highlighted fields:

1. Start the workbench of WebSphere Development Studio Client, and open the WebFacing perspective, if it is not up and running already (see Figure 87).

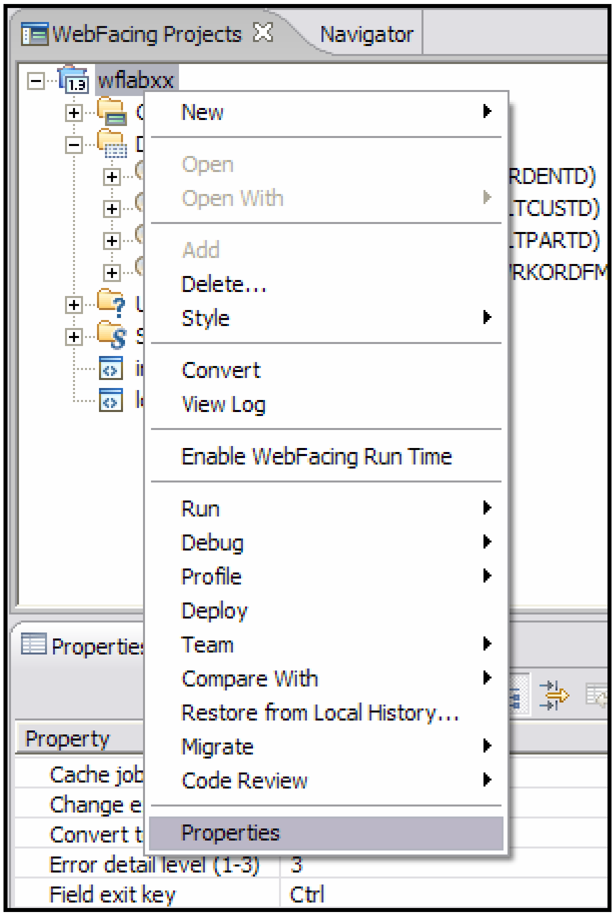
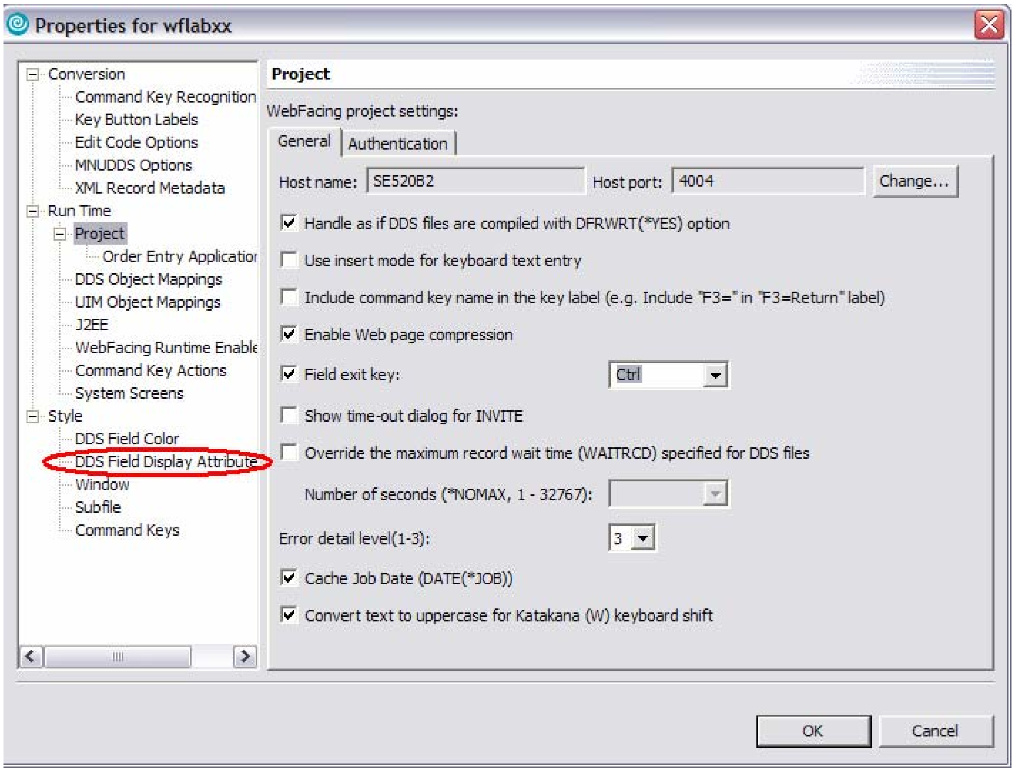


Figure 87. Start the workbench of WebSphere Development Studio Client.

2. Right-click the **wflabxx** project.
3. Click **Properties** on the pop-up menu.

The WebFacing project properties dialog opens (see Figure 88).



Project

WebFacing project settings:

Host name: SE520B2 Host port: 4004

☑ Handle as if DDS files are compiled with DFRWRT(*YES) option

☐ Use insert mode for keyboard text entry

☐ Include command key name in the key label (e.g. Include "F3=" in "F3=Return" label)

☑ Enable Web page compression

☑ Field exit key: Ctrl

☐ Show time-out dialog for INVITE

☐ Override the maximum record wait time (WAITRCD) specified for DDS files

Number of seconds (*NOMAX, 1 - 32767):

Error detail level(1-3): 3

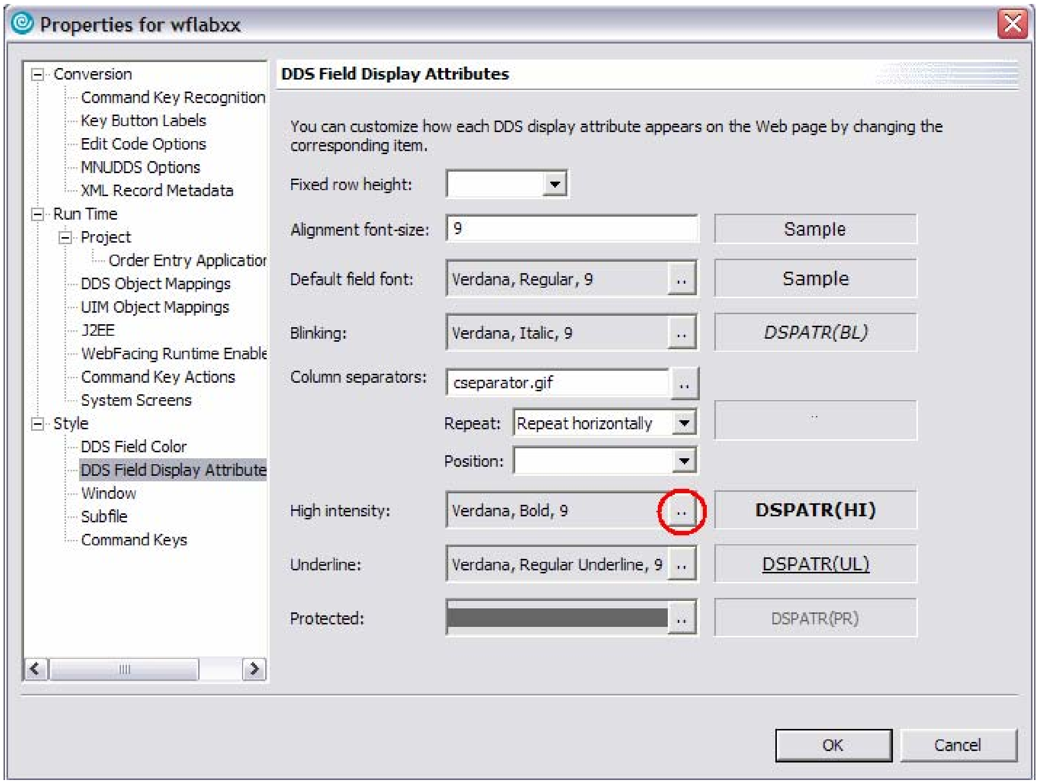
☑ Cache Job Date (DATE(*JOB))

☑ Convert text to uppercase for Katakana (W) keyboard shift

Figure 88. WebFacing project properties dialog

4. Under **Style** in the left pane of the Properties for the wflabxx project, click **DDS Field Display Attributes**.

In the right pane of the Properties for wflabxx, the DDS Field Display Attributes pane opens (see Figure 89).



DDS Field Display Attributes

You can customize how each DDS display attribute appears on the Web page by changing the corresponding item.

Fixed row height:

Alignment font-size: 9 Sample

Default field font: Verdana, Regular, 9 Sample

Blinking: Verdana, Italic, 9 DSPATR(BL)

Column separators: cseparator.gif

Repeat: Repeat horizontally

Position:

High intensity: Verdana, Bold, 9 DSPATR(HI)

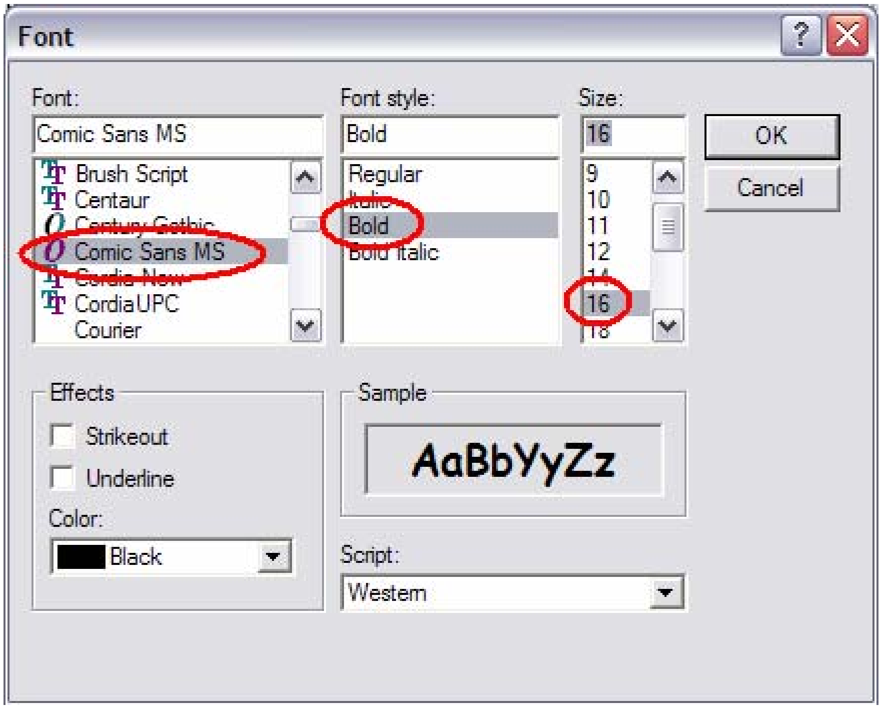
Underline: Verdana, Regular Underline, 9 DSPATR(UL)

Protected: DSPATR(PR)

Figure 89. ***DDS Field Display Attributes*** *pane*

5. Click the push button to the right of the **High intensity** list.

The Font dialog opens (see Figure 90).



Font: Comic Sans MS

Font style: Bold

Size: 16

Color: Black

Script: Western

*Figure 90. **Font** dialog*

6. Under the **Font** list, select **Comic Sans MS**.
7. Under the **Font style** list, select **Bold**.
8. Under **Size**, select **16**.

   Note: Do not try to change the color in this dialog; the color setting does not work. Later in this chapter, you see how to change the color.

   This changes the font size and appearance for parts that have the high intensity attribute active.

9. Click **OK** on the Font dialog.
10. On the Properties page, click **OK**.

Checking the new style

Check the new style in your application by doing the following.

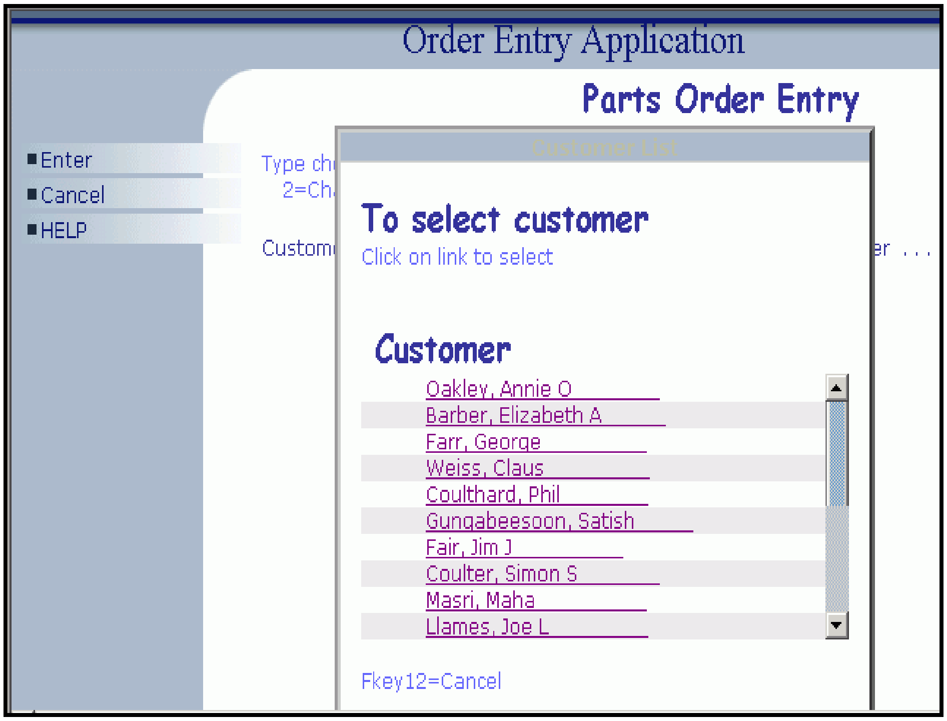
1. Right-click the project icon in the WebFacing Projects list.
2. Click **Run on Server** on the pop-up menu.
3. Go to the browser pane and click the Order Entry link.

   You see the first screen of your application.

   Note: See the different font type and font size of some of the text; these are the highlighted areas of your 5250 screen.

4. Prompt for the customer.

   The customer selection screen has highlighted text and this text shows in the Comic Sans MS font and size 16 (see Figure 91).



To select customer
Click on link to select

Customer

Oakley, Annie O
Barber, Elizabeth A
Farr, George
Weiss, Claus
Coulthard, Phil
Gungabeesoon, Satish
Fair, Jim J
Coulter, Simon S
Masri, Maha
Llames, Joe L

Fkey12=Cancel

Figure 91. Customer selection screen has highlighted text

   Note: If the text is still showing the old fonts, most likely, the browser has cached the page, and you need to close the browser window, and restart the application. This brings up a new instance of the browser without cached content.

5. The Font dialog has a bug, making it impossible to change the font colors directly.

Next, you learn how to change WebFacing style classes.

Exercise 6.2: Changing styles directly

Instead of working with properties in the WebFacing project, you can also change the styles being applied directly in the Cascading Style Sheet (.css) file. In this exercise, you see where this file is located in the WebFacing Project structure and how to find your way around inside this file.

Modify the cascading Style sheet

To locate the cascading Style sheet file:

1. Click the **Navigator** tab to switch to the Navigator view (see Figure 92).

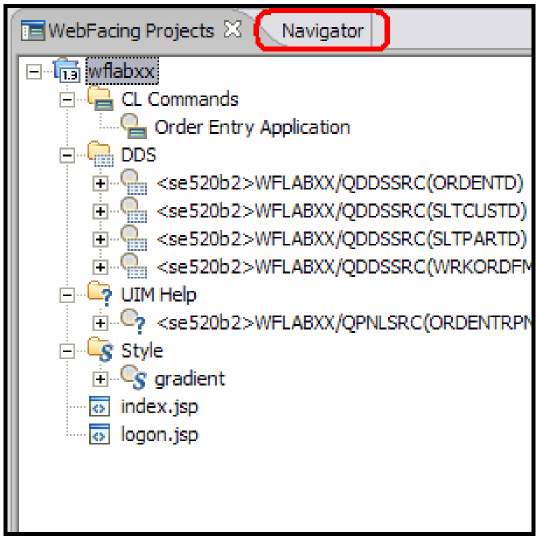

WebFacing Projects
Navigator
wflabxx
CL Commands
Order Entry Application
DDS
<se520b2>WFLABXX/QDDSSRC(ORDENTD)
<se520b2>WFLABXX/QDDSSRC(SLTCUSTD)
<se520b2>WFLABXX/QDDSSRC(SLTPARTD)
<se520b2>WFLABXX/QDDSSRC(WRKORDFM
UIM Help
<se520b2>WFLABXX/QPNLSRC(ORDENTRPN
Style
gradient
index.jsp
logon.jsp

*Figure 92. Switch to **Navigator** view*

2. Locate the apparea.css file in the WebFacing project. It is located in the following directory hierarchy: wflabxx\WebContent\webfacing\styles\apparea\ (see Figure 93).

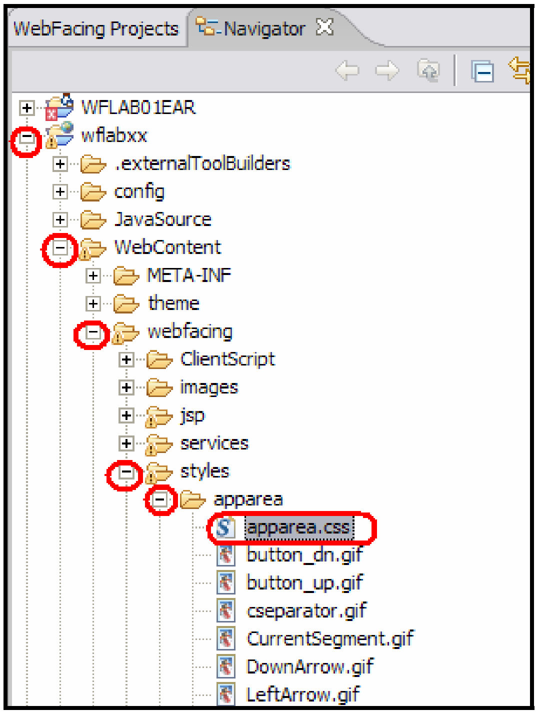


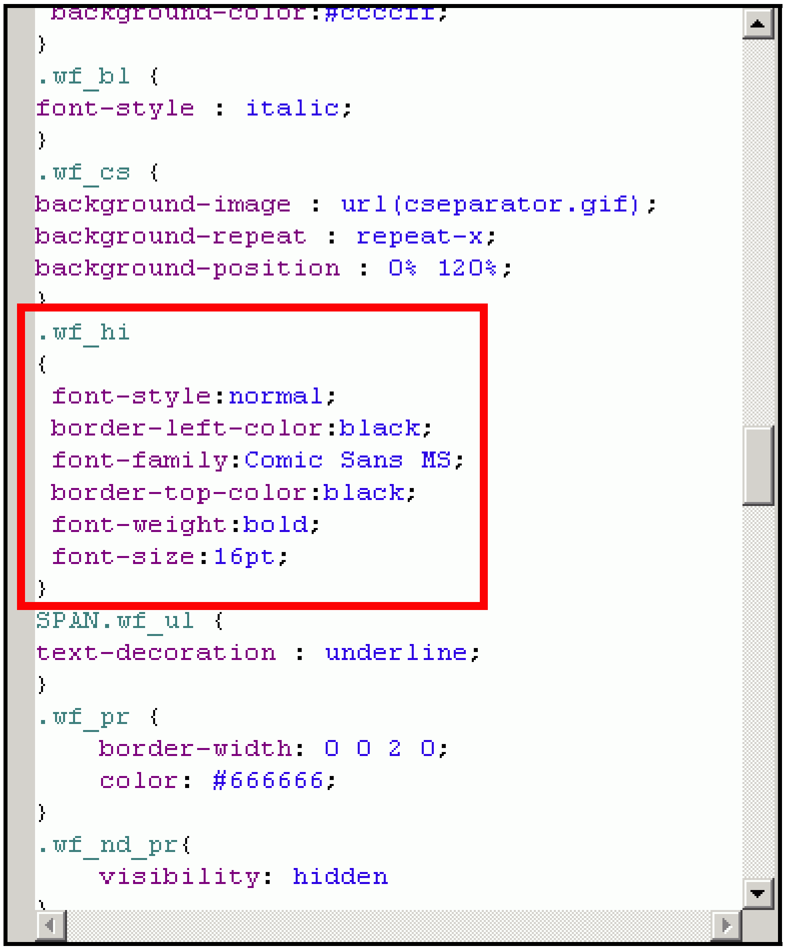
Figure 93. The WebFacing project, ***apparea.css*** *file*

3. Double-click the apparea.css file icon in the Navigator view to open an editor for this file.

Next, you change the WebFacing highlighted style class.

The editor dialog opens and shows the style source.

4. Scroll down the source file until you reach the **.wf-hi** class (see Figure 94).



```
 background-color:#ccccff;
}
.wf_bl {
font-style : italic;
}
.wf_cs {
background-image : url(cseparator.gif);
background-repeat : repeat-x;
background-position : 0% 120%;
}
.wf_hi
{
 font-style:normal;
 border-left-color:black;
 font-family:Comic Sans MS;
 border-top-color:black;
 font-weight:bold;
 font-size:16pt;
}
SPAN.wf_ul {
text-decoration : underline;
}
.wf_pr {
    border-width: 0 0 2 0;
    color: #666666;
}
.wf_nd_pr{
    visibility: hidden
}
```

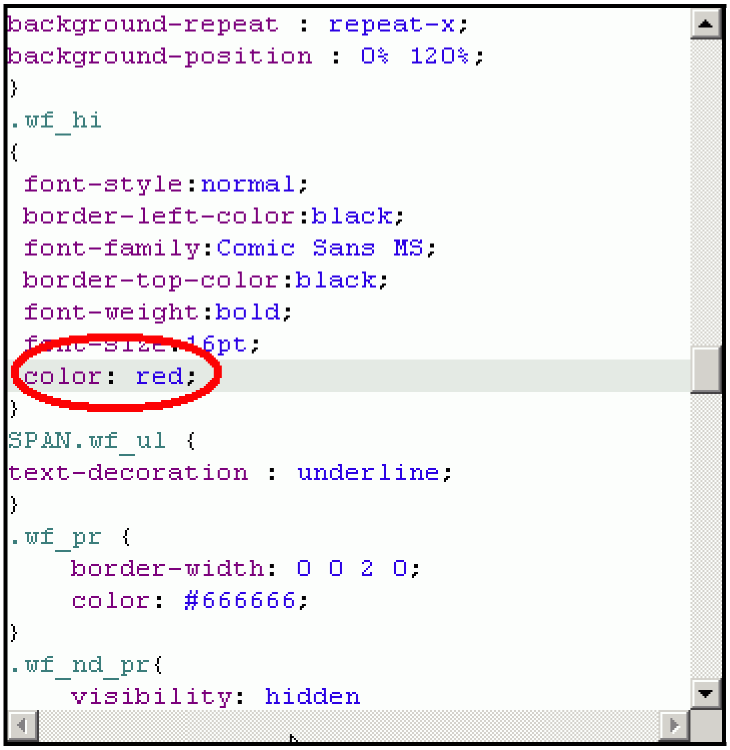
Figure 94. the ***.wf-hi*** *class*

Notice your changes from the previous exercise are applied here. The IBM WebFacing Tool has created style sheet classes for the different 5250 parts and attribute combinations and stored them here. Each of the lines starting with a dot contains a style class name. The statements inside the curly brackets after the style class name describe the attributes of the class itself. For this WebFacing .css file, the name of the classes is self-describing. For example, .wf_hi defines how a part that has the highlighted attribute active is displayed in the browser; .wf_cs describes the column separator, and so forth. You can go in and change the display characteristics for any of the 5250 attributes.

Next, you change the color attribute of the highlighted class.

5. Position the cursor at the end of the last line of the .wf_hi class, before the ending curly (}) bracket.
6. Press the **Enter** key to insert a line.

7. Enter the color attribute you want to use, for example, **color: red;** (see Figure 95).



```
background-repeat : repeat-x;
background-position : 0% 120%;
}
.wf_hi
{
 font-style:normal;
 border-left-color:black;
 font-family:Comic Sans MS;
 border-top-color:black;
 font-weight:bold;
 font-size:16pt;
 color: red;
}
SPAN.wf_ul {
text-decoration : underline;
}
.wf_pr {
    border-width: 0 0 2 0;
    color: #666666;
}
.wf_nd_pr{
    visibility: hidden
```

*Figure 95. Enter color attribute, **red***

Note: Do not forget the semicolon to delimit the line.

8. Save the change by clicking the Save icon in the workbench (see Figure 96).

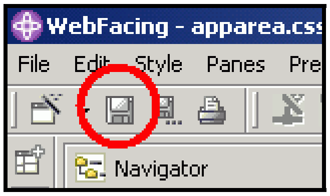
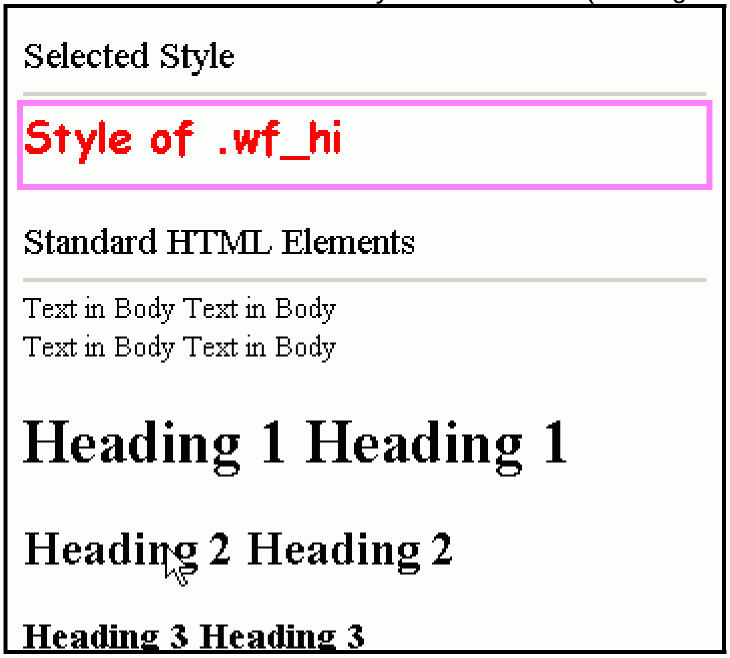


Figure 96. Save the change

For IBM WebFacing Tool V5 and V6 users, there is a feature available in the Style sheet editor that allows you to look at the changes you have made to a class directly in the editor without actually invoking the page and showing in a browser.

9. Look at the left beside the Style-sheet editor (see Figure 97).



Selected Style

Style of .wf_hi

Standard HTML Elements

Text in Body Text in Body
Text in Body Text in Body

Heading 1 Heading 1

Heading 2 Heading 2

Heading 3 Heading 3

Figure 97. Style-sheet editor

To see the changed user interface, run the application again.

Test the changed WebFacing application

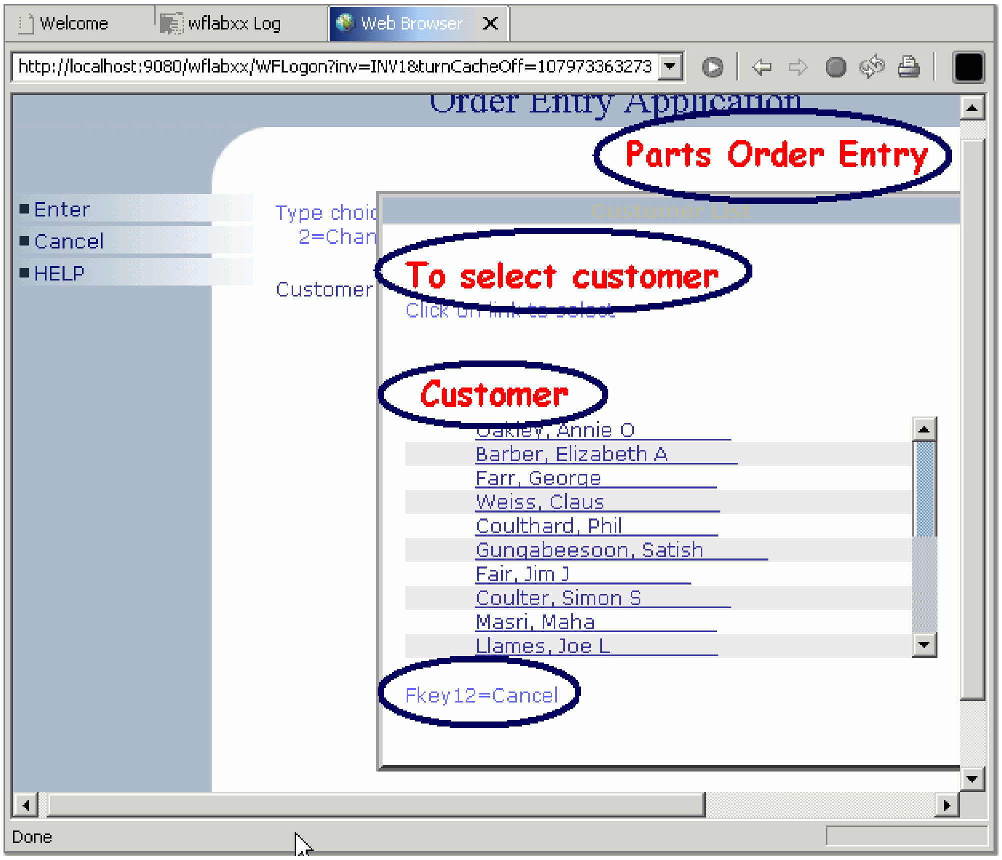
To test the changed WebFacing application, proceed with the following steps:

1. Click the **X** in the Window bar of any editors or views still open at the right upper-hand side of the workbench to close them all.

   Now, check the new style attribute in your application.

2. Right-click the project icon in the WebFacing Projects view or Navigator view in the WebFacing perspective.
3. Select **Run on Server** on the pop-up menu.
4. Go to the browser pane and click the Order Entry link.

You see the first screen of your application. Notice the different color of the text in the highlighted areas of your Web page. Prompt for the customer. The customer selection screen shows the same changed attributes; remember style changes are applied to all pages as shown (see Figure 98).



Oakley, Annie O
Barber, Elizabeth A
Farr, George
Weiss, Claus
Coulthard, Phil
Gungabeesoon, Satish
Fair, Jim J
Coulter, Simon S
Masri, Maha
Llames, Joe L

Figure 98. Style changes are applied to all pages

Note: If the text is still showing the old colors, most likely, the browser has cached the page, and you need to close the browser window, and restart the application. This opens a new instance of the browser without cached content.

You might not have noticed it before, but there is still a command key description at the bottom of the Customer Selection list window.

You fix this in the next chapter.

Adding command key rules and labels

Most of the command key descriptions placed at the bottom of the original panels in the sample Order Entry Application have been removed from the WebFacing user interface. The removal happens automatically if the command key descriptions follow the Common User interface Access (CUA) rules (for example, F3=Exit). If they do not follow these rules, you can supply the IBM WebFacing Tool with the command key description rules you use in your user interface, and the IBM WebFacing Tool applies these rules to recognize your specific command key descriptions. One of the record formats in the sample application contains a command key description that does not follow the CUA rules.

In this chapter, you use the WebFacing conversion properties dialogs to add your own command key description rule. Also, in your user interface you might have enabled command keys without describing them in the user interface because they are only used for specific functions and your end user knows how they work. For example, command key **F3** for ending an application or command key **F12** for canceling a task are well known to be enabled for these tasks. Therefore, your user interface might not list them explicitly at the bottom of the panel. The problem with this kind of user interface design is that the IBM WebFacing Tool does not have a string it can put in the label of the push buttons on the Web page. For example, it just puts CF12 for the **F12** active command key in the push button label. However, there is an alternative. The IBM WebFacing Tool allows you to specify labels for these common command keys so it can label the command key push buttons correctly. You just have to tell the IBM WebFacing Tool in its conversion properties what the label is.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 7.1: Adding a new command key description pattern
- Exercise 7.2: Adding command key button labels

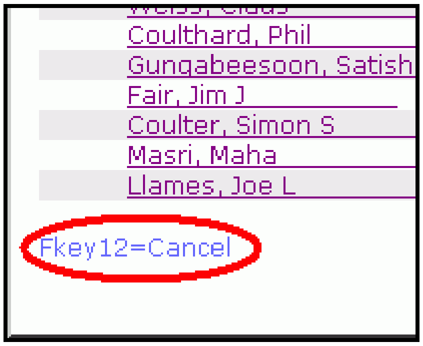
The exercises in this chapter must be completed in order. Start with “Exercise 7.1: Adding a new command key description pattern” when you are ready to begin.

Length of time

This chapter takes approximately 10 minutes to complete.

Exercise 7.1: Adding a new command key description pattern

The prefix for the command key description in the SLTCUST record format is Fkey= (see Figure 99).



Coulthard, Phil
Gungabeesoon, Satish
Fair, Jim J
Coulter, Simon S
Masri, Maha
Llames, Joe L

Fkey12=Cancel

Figure 99. ***Fkey=*** *is the prefix for the command key.*

This does not follow the CUA rules normally used in i5/OS user interfaces, so the WebFacing conversion does not recognize this pattern and does not hide it. You have to add this prefix pattern to the IBM WebFacing Tool conversion rules in the WebFacing Properties dialog.

To add a new command key description pattern to WebFacing conversion rules, follow these steps:

1. In the WebFacing Projects view, right-click your WebFacing project wflabxx.

2. Click **Properties** on the pop-up menu (see Figure 100).

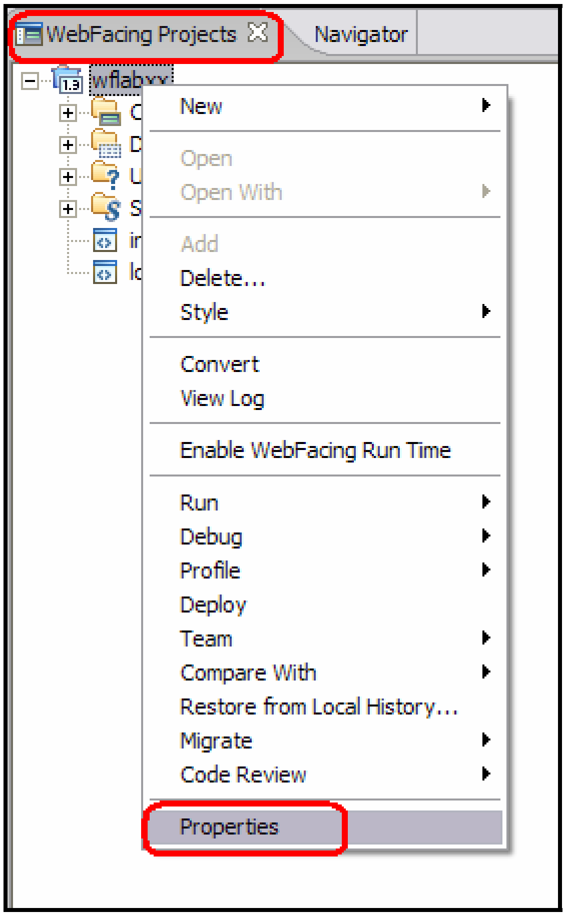


*Figure 100. Click **Properties** on the pop-up menu.*

The Properties for wflabxx dialog opens (see Figure 100).

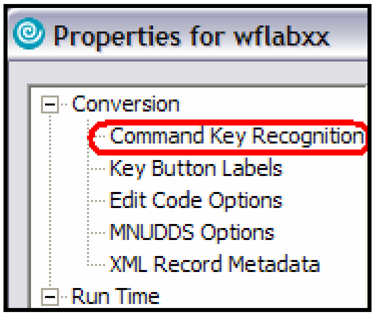
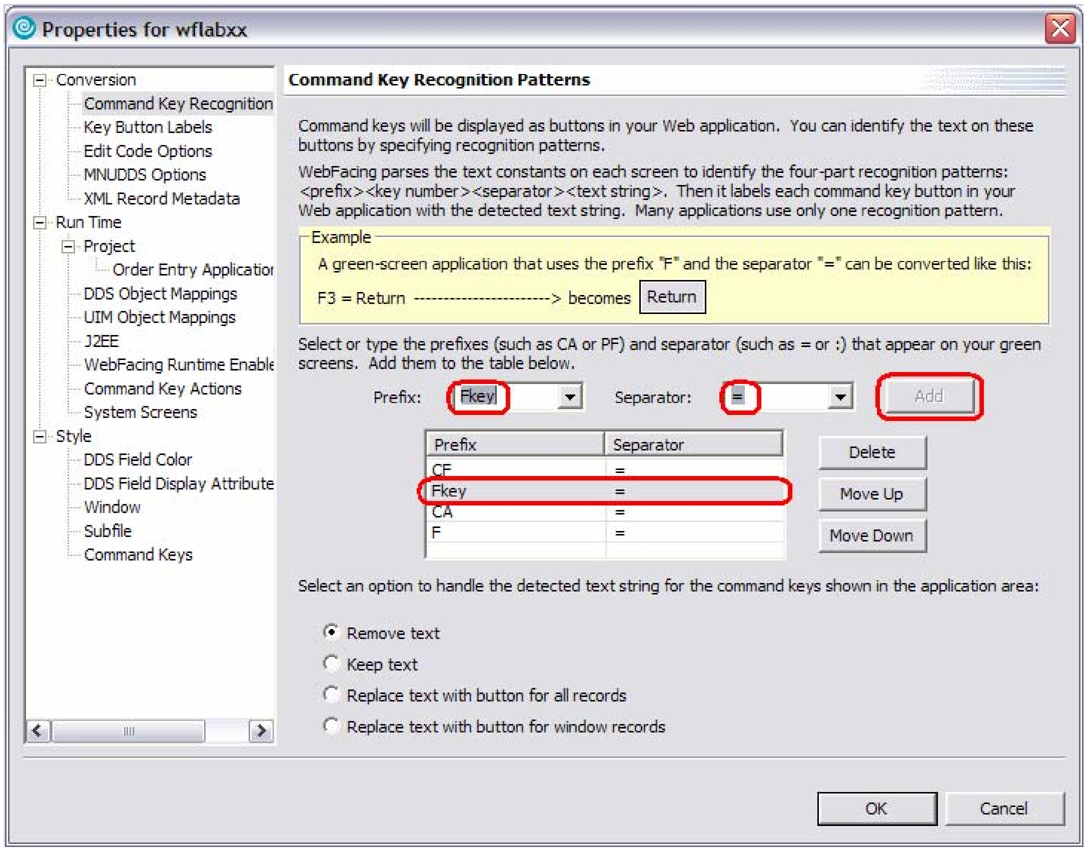


*Figure 101. The **Properties for wflabxx** dialog*

3. In the left pane of the Properties for wflabxx, under **Conversion**, select **Command Key Recognition Patterns**.

   The right pane switches to Command Key Recognition Patterns (see Figure 102).



Command Key Recognition Patterns

Command keys will be displayed as buttons in your Web application. You can identify the text on these buttons by specifying recognition patterns.

WebFacing parses the text constants on each screen to identify the four-part recognition patterns: <prefix><key number><separator><text string>. Then it labels each command key button in your Web application with the detected text string. Many applications use only one recognition pattern.

Example

A green-screen application that uses the prefix "F" and the separator "=" can be converted like this:

F3 = Return -----------------------> becomes Return

Select or type the prefixes (such as CA or PF) and separator (such as = or :) that appear on your green screens. Add them to the table below.

Prefix: Fkey Separator: =

Prefix	Separator
CF	=
Fkey	=
CA	=
F	=

Select an option to handle the detected text string for the command keys shown in the application area:

● Remove text
○ Keep text
○ Replace text with button for all records
○ Replace text with button for window records

Figure 102. Command Key Recognition Patterns

4. In the **Prefix** list, type `Fkey`.
5. In the **Separator** list, leave **=** as is.
6. Click **Add** to the right of these two lists.

   The additional rule is now part of this style.

7. Click **OK**.

Next, you test this feature by converting the DDS member and running the application.

Reconverting DDS members

Because these are conversion properties that you are changing and not style properties, as in the previous chapter, you need to reconvert all members that contain command key descriptions with this pattern. In the sample Order Entry application, this is only SLTCUSTD member.

To reconvert DDS members:

1. In the workbench in the WebFacing Project view, expand the wflabxx project, if not already expanded.
2. Expand the DDS folder, if not already expanded.
3. Right-click the SLTCUSTD member icon.
4. Click **Convert** on the pop-up menu.

   After the conversion, look at the change.

Testing the applied new rule

You now see how the new rule is applied to your WebFacing application.

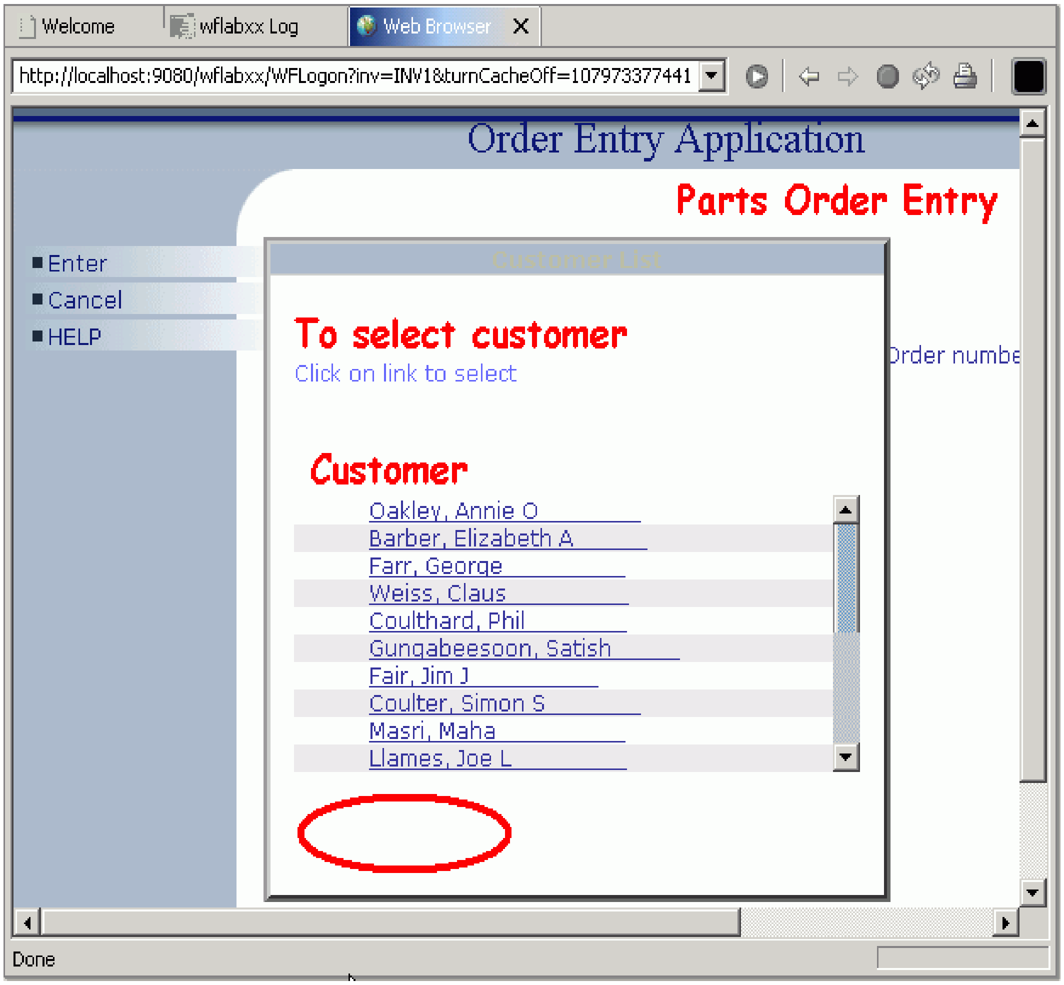
To test the applied new rule:

1. Right-click the project wflabxx in the WebFacing Projects view.
2. Click **Run on Server** on the pop-up menu.
3. Go to the browser pane and click the Order Entry link.

   You see the first screen of the sample application.

4. Click **Prompt** or press **F4**.

   The Customer Selection list window opens (see Figure 103).



To select customer

Click on link to select

Customer

- Oakley, Annie O
- Barber, Elizabeth A
- Farr, George
- Weiss, Claus
- Coulthard, Phil
- Gungabeesoon, Satish
- Fair, Jim J
- Coulter, Simon S
- Masri, Maha
- Llames, Joe L

Figure 103. The Customer Selection list window opens.

Note: Notice that the command key text Fkey12=Cancel is gone from the bottom of the window.

5. Select a customer or cancel by pressing **F12** to return to the Order Entry Application window.

Now that you know how to change the command key recognition patterns in the IBM WebFacing Tool, you can add text to frequently used command keys that do not have a text description in the DDS source.

Back on the Order Entry panel, you can see that the push button for F12 does not have a description. Its label shows F12 (see Figure 104). You learn how to fix the label in the next exercise.

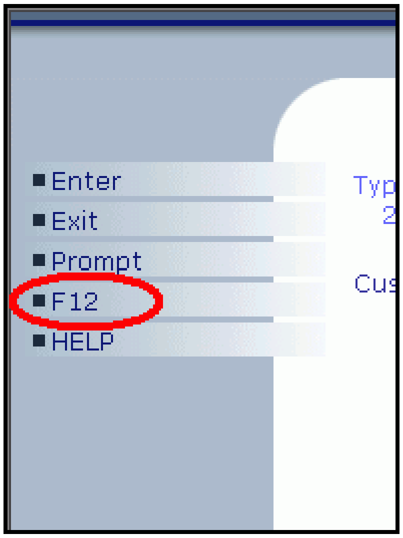


*Figure 104. The **F12** key without a description*

Exercise 7.2: Adding command key button labels

You can specify descriptions for command keys that are commonly used in your application but which are not described in the DDS source. In this way, the push button labels on the Web pages can automatically be changed during conversion to the correct text. You add the description *Cancel* to the WebFacing conversion properties so the push button is created to show a Cancel label instead of an F12 label.

To add command-key button labels:

1. Right-click your WebFacing project wflabxx (see Figure 105).

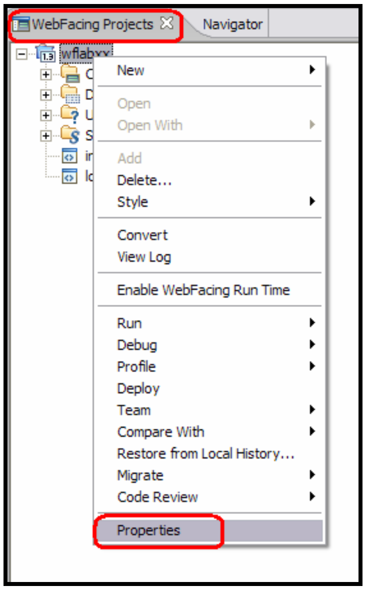


*Figure 105. Right-click your WebFacing project **wflabxx**.*

2. Click **Properties** on the pop-up menu (see Figure 106).

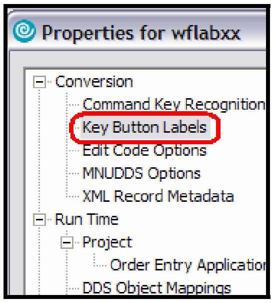
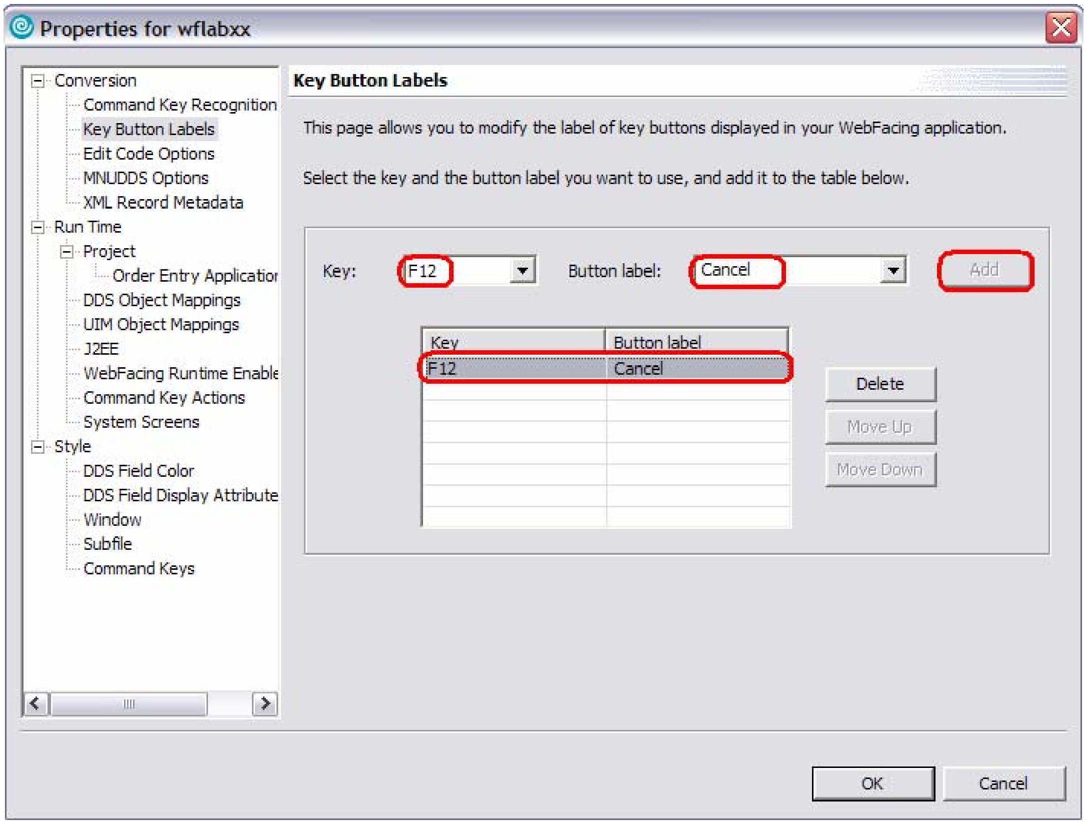


Figure 106. Properties for wflabxx

The Properties for wflabxx dialog opens.

3. In the left pane of this dialog, under **Conversion** select **Key Button Labels**.

   In the right pane of this dialog, the command key **Button label** opens (see Figure 107).



This page allows you to modify the label of key buttons displayed in your WebFacing application.

Select the key and the button label you want to use, and add it to the table below.

Key: F12 Button label: Cancel

Key	Button label
F12	Cancel

*Figure 107. **Button label** key opens*

4. Under the **Command key** list, select **F12**.
5. Under the **Button label** list, select **Cancel**.
6. Click **Add** to the right of these two lists.

   The additional label is now part of the WebFacing conversion properties for this WebFacing project.

7. Click **OK** at the bottom of the properties dialog.

   Next, back in the WebSphere Development Studio Client workbench, convert the changed application.

Reconverting DDS members

To convert the application in the workbench, perform the following steps:

1. Expand the wflabxx project, if not already expanded.
2. Expand the DDS folder, if not already expanded.
3. Right-click the ORDENTD member icon.
4. Click **Convert** on the pop-up menu.

   After the conversion, look at the change.

Testing the new label

Look at the new label for the **F12** push button in your WebFacing application.

To test the new label:

1. Right-click wflabxx project in the WebFacing Projects view.
2. Click **Run on Server** on the pop-up menu.
3. Go to the browser pane and click the Order Entry link.
4. Check that the **Cancel** button on the first page now shows the correct label as shown (see Figure 108).

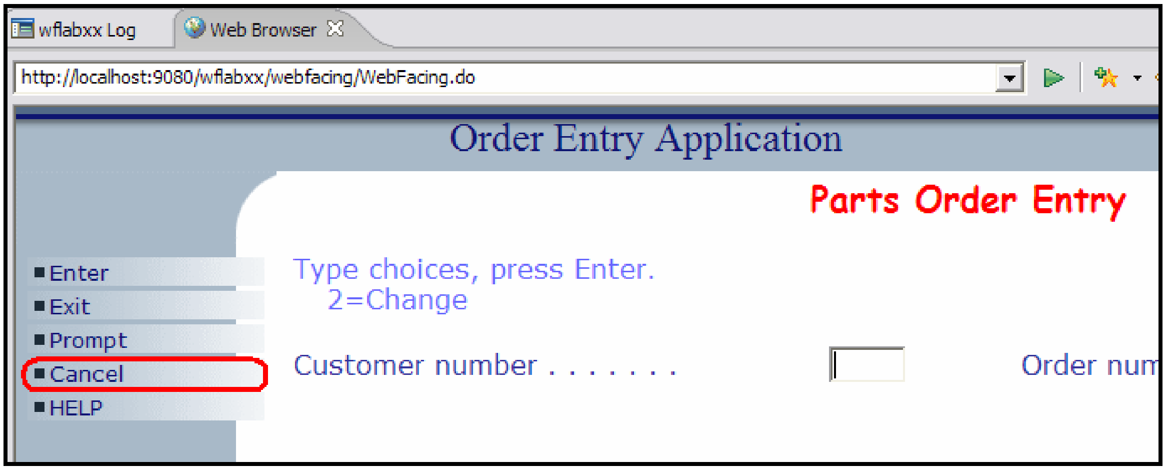

wflabxx Log
Web Browser
http://localhost:9080/wflabxx/webfacing/WebFacing.do
Order Entry Application
Parts Order Entry
Enter
Exit
Prompt
Cancel
HELP
Type choices, press Enter.
2=Change
Customer number . . . .
Order num

*Figure 108. Check that **Cancel** has the correct label*

5. If the **push Cancel** button still has the old text, restart the Application server.
6. Click the **Servers** tab at the bottom view in the workbench.
7. Right-click the server icon.

8. Click **Restart>Start** on the pop-up menu (see Figure 109).

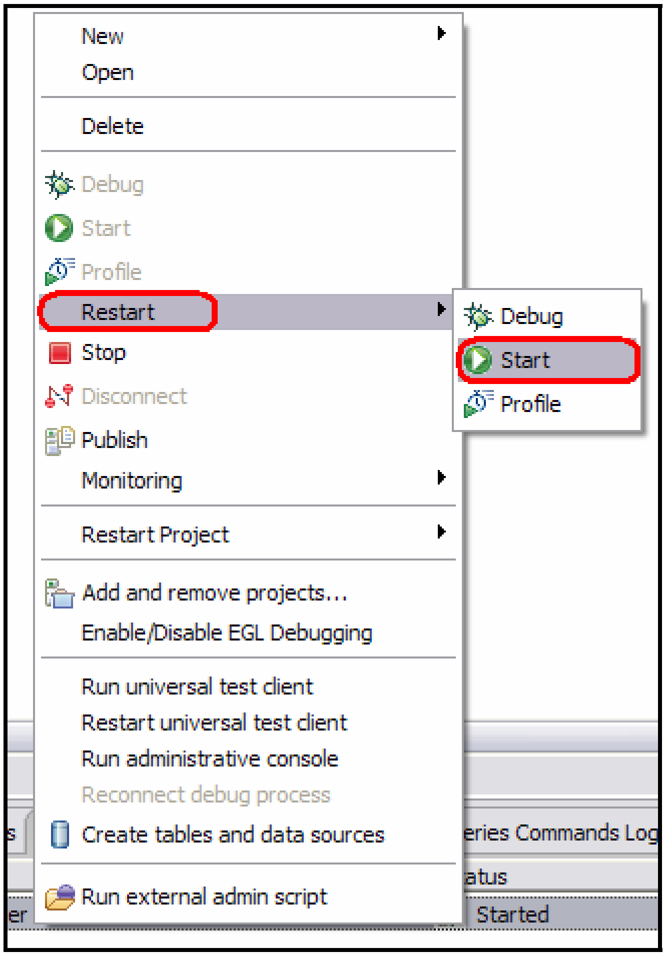


*Figure 109. Click **Restart>Start** on the pop-up menu.*

9. Run the application again.

Exercise 7.3: Remove unwanted buttons

The IBM WebFacing Tool can generate unwanted buttons that you might not want to display in the WebFaced application. This section discusses how to remove unwanted buttons.

Remove unwanted buttons

The IBM WebFacing Tool might generate unwanted buttons that users do not want displayed on their Web pages. The **HELP** button, as shown in the partial screen display, below might be one of those button types (see Figure 110).

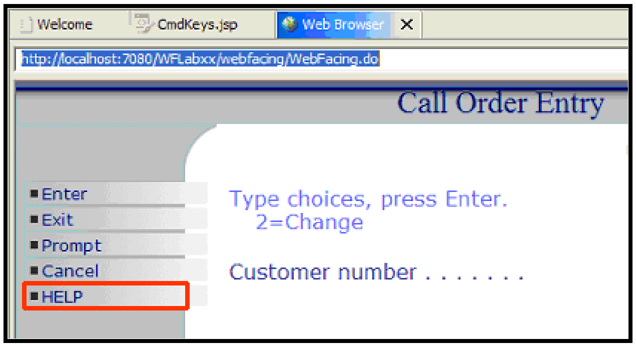

Welcome
CmdKeys.jsp
Web Browser
http://localhost:7080/WFLabxx/webfacing/WebFacing.do
Call Order Entry
Enter
Exit
Prompt
Cancel
HELP
Type choices, press Enter.
2=Change
Customer number . . . .

*Figure 110. **Help** button*

To remove the help button, the CmdKeys.jsp file must be edited. An easy way to understand how CmdKeys.jsp works is to run the application in debug mode. Single stepping through the code shows you how all pages are built.

To remove the HELP button you must perform the following steps:

1. For this exercise, you remove the default help button displayed on every Web page. First, change to the Web Perspective by clicking **Window > Open Perspective > Web**. If Web is not on the list of perspective, click **Other…** and select the Web perspective (see Figure 111).

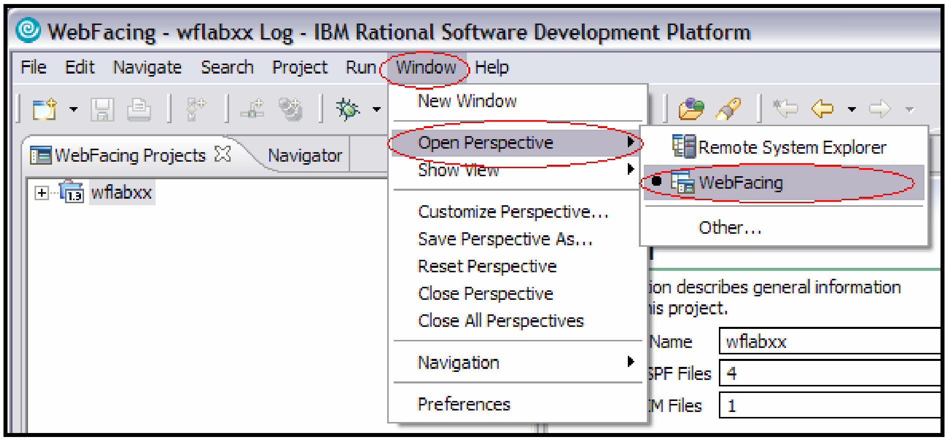


Figure 111. Change the Web Perspective

2. Be sure you are using the Navigator view. Locate the apparea.css file in the Navigator view of the WebFacing Perspective. It is located in the following directory hierarchy: wflabxx\WebContent\webfacing\styles\chrome\html. Open the CmdKeys.jsp file by clicking the file (see Figure 112).

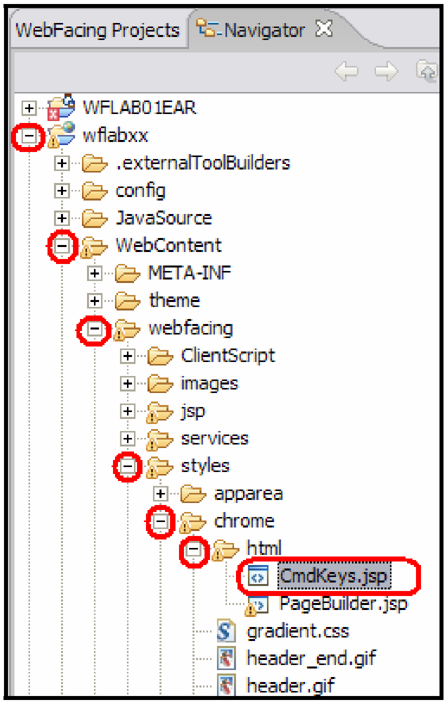


Figure 112. Open ***CmdKeys.jsp***

3. When CmdKeys.jsp is open in the editor, click the **Source** tab located at the bottom of the window (see Figure 113).

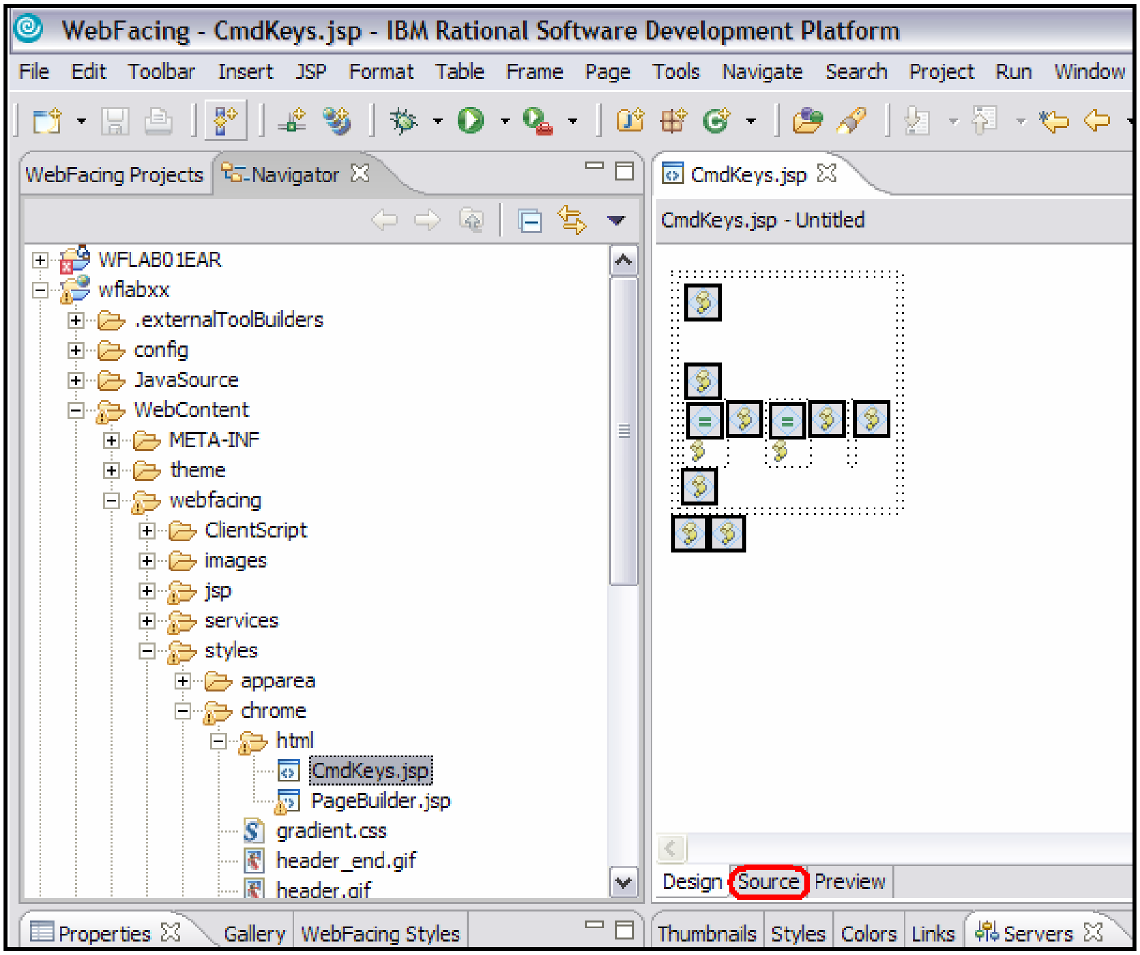
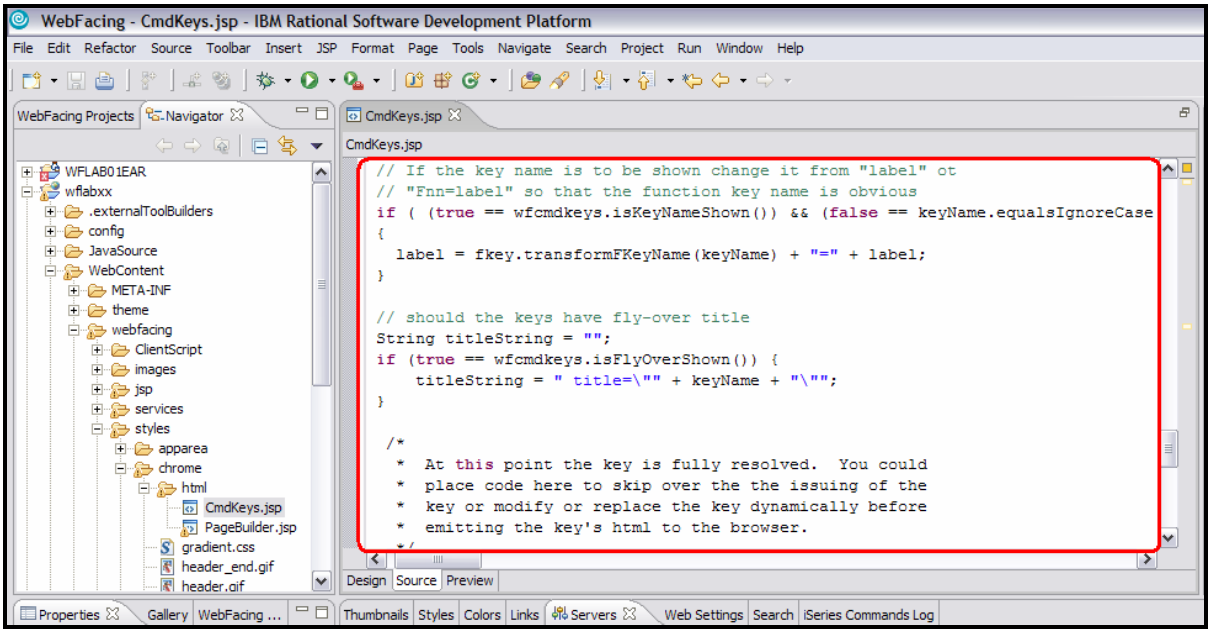


*Figure 113. Click **Source** tab*

4. CmdKeys.jsp opens in an editor window. Look for the code in this file that looks similar to the following in Figure 114):



```
// If the key name is to be shown change it from "label" ot
// "Fnn=label" so that the function key name is obvious
if ( (true == wfcmdkeys.isKeyNameShown()) && (false == keyName.equalsIgnoreCase
{
  label = fkey.transformFKeyName(keyName) + "=" + label;
}

// should the keys have fly-over title
String titleString = "";
if (true == wfcmdkeys.isFlyOverShown()) {
    titleString = " title=\"" + keyName + "\"";
}


 /*
  *  At this point the key is fully resolved.  You could
  *  place code here to skip over the the issuing of the
  *  key or modify or replace the key dynamically before
  *  emitting the key's html to the browser.
  */
```

*Figure 114. **CmdKeys.jsp** opens in an editor window. Look for code*

The complete section of code is:

```
        // If the key name is to be shown change it from "label" ot
        // "Fnn=label" so that the function key name is obvious
       if ( (true == wfcmdkeys.isKeyNameShown()) && (false == keyName.equalsIgnoreCase(label)) &&
       fkey.isCommandKey() )
        {
          label = fkey.transformFKeyName(keyName) + "=" + label;
        }

              // should the keys have fly-over title
              String titleString = "";
              if (true == wfcmdkeys.isFlyOverShown()) {
                     titleString = " title=\"" + keyName + "\"";
              }

         /*
          *  At this point the key is fully resolved. You could
          *  place code here to skip over the the issuing of the
          *  key or modify or replace the key dynamically before
          *  emitting the key's html to the browser.
          */

        if (true == completeOverride)
        {
          %><td id="<%= idn%>" class="buttonup" width="<%=bWidth%>" height="<%=bHeight%>"
<%=titleString%> onmouseover="<%=wfcmdkeys.getJsVersionedPrefix()%>bo_ibm(this);"
onmouseout="<%=wfcmdkeys.getJsVersionedPrefix()%>bu_ibm(this, 'buttonup');"
onClick="<%=wfcmdkeys.getJsVersionedPrefix()%>bd_ibm(this);<%= ovrideClick %>"><%=
fkey.transformFKeyName(label) %><script
language='JavaScript'><%=wfcmdkeys.getJsVersionedPrefix()%>bu_ibm(<%=idn%>);</script></td><%
        }
        else
        {
          %><td id="<%= idn%>" class="buttonup" width="<%=bWidth%>" height="<%=bHeight%>"
<%=titleString%> onmouseover="<%=wfcmdkeys.getJsVersionedPrefix()%>bo_ibm(this);"
onmouseout="<%=wfcmdkeys.getJsVersionedPrefix()%>bu_ibm(this, 'buttonup');"
onClick="<%=wfcmdkeys.getJsVersionedPrefix()%>bd_ibm(this);<%= ovrideClick
%><%=wfcmdkeys.getJsVersionedPrefix()%>validateAndSubmit('<%= keyName
%>','<%=wfcmdkeys.getUniqueId()%>');"><%= fkey.transformFKeyName(label) %><script
language='JavaScript'><%=wfcmdkeys.getJsVersionedPrefix()%>regCmdKey('<%= idn%>', '<%= keyName
%>', '<%= ovrideURI %>','<%= ovrideFrame %>','<%=wfcmdkeys.getUniqueId()%>');</script></td><%
        }
        col++;
      }
    }
  }
}
```

This code is what is responsible for displaying the keys on each page. A very useful way to determine what the code is doing is to set a breakpoint on the “if” statement and run the application in debug mode watching the value of label in the debug window.

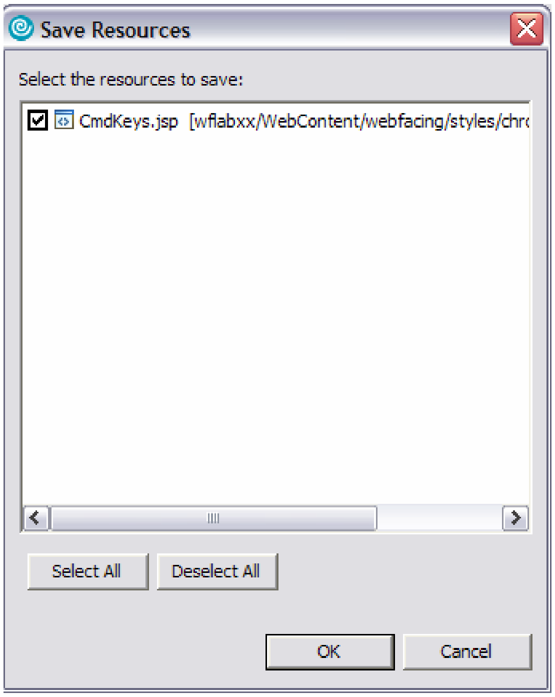
5. To suppress the **HELP** key from being displayed, one way is to check the value of the label, and not write the code or increment the col value if label contains the value **HELP**. The code to do this, looks as follows.

   Note: You only add two lines of code and a comment. The three lines are as follows. The first two need to be added before the comment **//** (if the keys have fly-over title) and the **}** needs to be added at the end of the block of code. The code you need to add is in a larger font in bold. Save your changes.

```
//If HELP button, remove
if(label.equals("HELP") == false){



}
```

```
        // If the key name is to be shown change it from "label" ot
        // "Fnn=label" so that the function key name is obvious
        if ( (true == wfcmdkeys.isKeyNameShown()) && (false == keyName.equalsIgnoreCase(label)) )
        {
          label = fkey.transformFKeyName(keyName) + "=" + label;
        }
               //If HELP button, remove
               if(label.equals("HELP") == false){
               // should the keys have fly-over title
               String titleString = "";
               if (true == wfcmdkeys.isFlyOverShown()) {
                       titleString = " title=\"" + keyName + "\"";
               }

         /*
          *  At this point the key is fully resolved. You could
          *  place code here to skip over the issuing of the
          *  key or modify or replace the key dynamically before
          *  emitting the key's html to the browser.
          */

        if (true == completeOverride)
        {
          %><td id="<%= idn%>" class="buttonup" width="<%=bWidth%>" height="<%=bHeight%>"
<%=titleString%> onmouseover="<%=wfcmdkeys.getJsVersionedPrefix()%>bo_ibm(this);"
onmouseout="<%=wfcmdkeys.getJsVersionedPrefix()%>bu_ibm(this, 'buttonup');"
onClick="<%=wfcmdkeys.getJsVersionedPrefix()%>bd_ibm(this);<%= ovrideClick %>"><%=
fkey.transformFKeyName(label) %><script
language='JavaScript'><%=wfcmdkeys.getJsVersionedPrefix()%>bu_ibm(<%=idn%>);</script></td><%
        }
        else
        {
          %><td id="<%= idn%>" class="buttonup" width="<%=bWidth%>" height="<%=bHeight%>"
<%=titleString%> onmouseover="<%=wfcmdkeys.getJsVersionedPrefix()%>bo_ibm(this);"
onmouseout="<%=wfcmdkeys.getJsVersionedPrefix()%>bu_ibm(this, 'buttonup');"
onClick="<%=wfcmdkeys.getJsVersionedPrefix()%>bd_ibm(this);<%= ovrideClick
%><%=wfcmdkeys.getJsVersionedPrefix()%>validateAndSubmit('<%= keyName
%>','<%=wfcmdkeys.getUniqueId()%>');"><%= fkey.transformFKeyName(label) %><script
language='JavaScript'><%=wfcmdkeys.getJsVersionedPrefix()%>regCmdKey('<%= idn%>', '<%= keyName
%>', '<%= ovrideURI %>','<%= ovrideFrame %>','<%=wfcmdkeys.getUniqueId()%>');</script></td><%
        }
        col++;
      }
    }
  }
   }
```

6. Save your changes to CmdKeys.jsp by selecting **Ctl + S**, selecting the Save icon 💾 from the tool bar, or selecting **File>Save** from the tool bar. If you exit the editor without saving your changes, you are prompted to save your changes in one of the two ways, depending on how the project has been modified. Select **Yes** or **OK** to save your changes and continue (see Figure 115 and Figure 116).

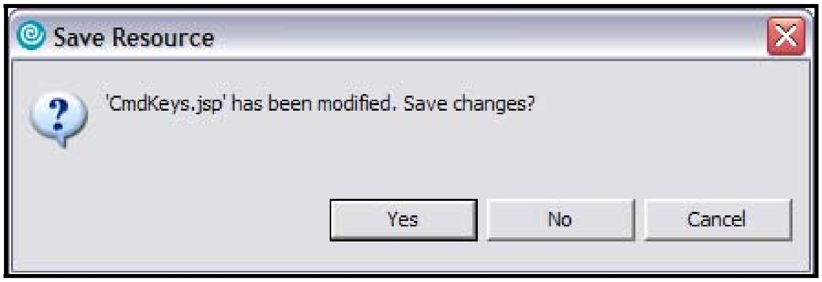


Save Resources

Select the resources to save:

☑ CmdKeys.jsp [wflabxx/WebContent/webfacing/styles/chr

Figure 115. Save resources

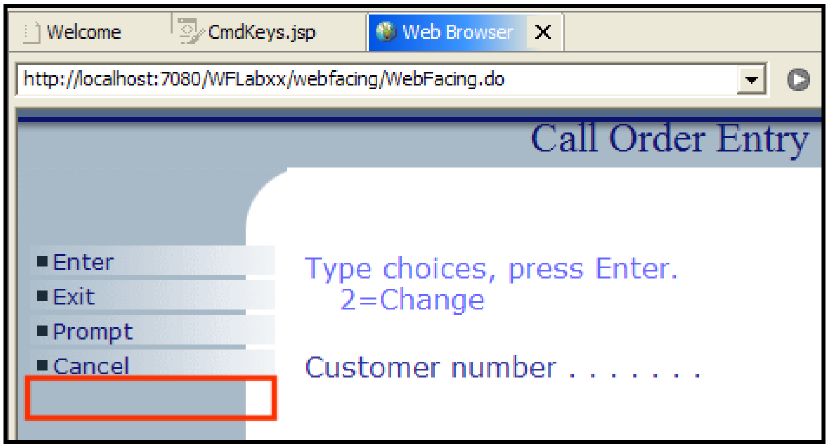


Save Resource

'CmdKeys.jsp' has been modified. Save changes?

*Figure 116. Save changes; click **Yes***

7. Run the application and the **HELP** key no longer display (see Figure 117).



Type choices, press Enter.
2=Change

Customer number . . . .

Figure 117. Run the application

Exercise 7.4: Modify the Enter *key text*

Some developers require the ability to change the **Enter** key text. This is quite easy to do for a whole project by making a simple change in the rtmessage.properties file. It is very important that you edit the correct file because one is the source for the other. The rtmessage.properties file contains a lot of useful values such as password setting, message text, and many other values.

Change the Enter key text

To change the **Enter** key text you must:

1. Open the rtmessage.properties file using the Navigator view in the WebFacing perspective. The file is located under your project: JavaSource>com>ibm>as400ad>webfacing>runtime>rtmessage.properties. Click the file to open it in the editor (see Figure 118).

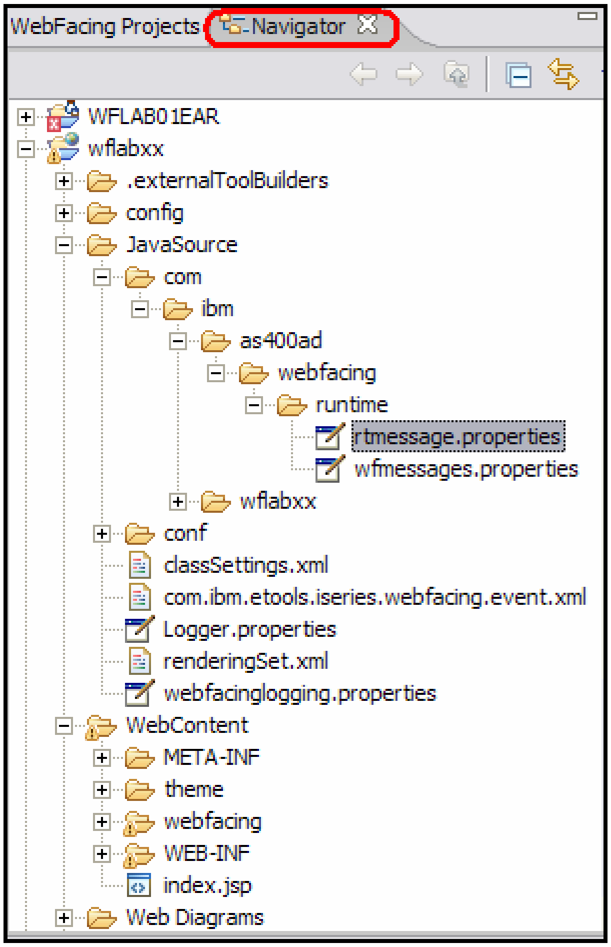
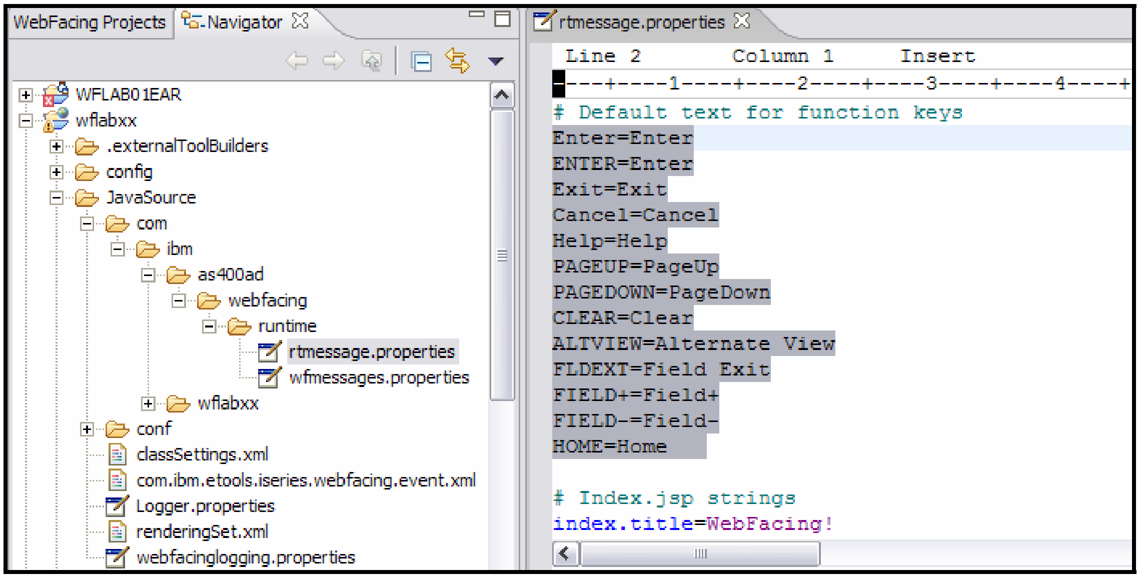


Figure 118. Use the Navigator view

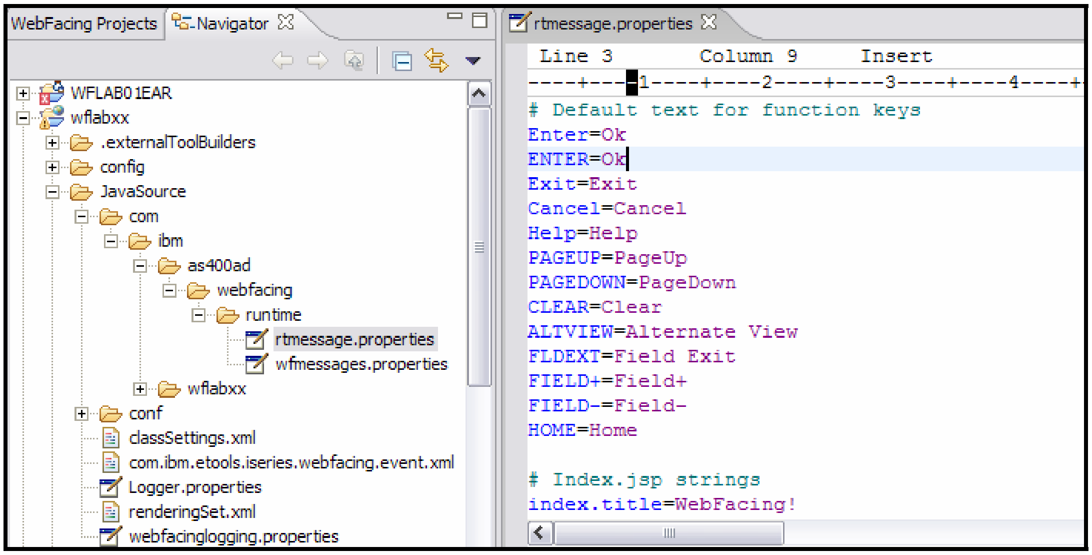
2. Look for the entry highlighted in the image below (see Figure 119).



```
 Line 2         Column 1       Insert
----+----1----+----2----+----3----+----4----+
# Default text for function keys
Enter=Enter
ENTER=Enter
Exit=Exit
Cancel=Cancel
Help=Help
PAGEUP=PageUp
PAGEDOWN=PageDown
CLEAR=Clear
ALTVIEW=Alternate View
FLDEXT=Field Exit
FIELD+=Field+
FIELD-=Field-
HOME=Home

# Index.jsp strings
index.title=WebFacing!
```

Figure 119. Look at highlighted entry text

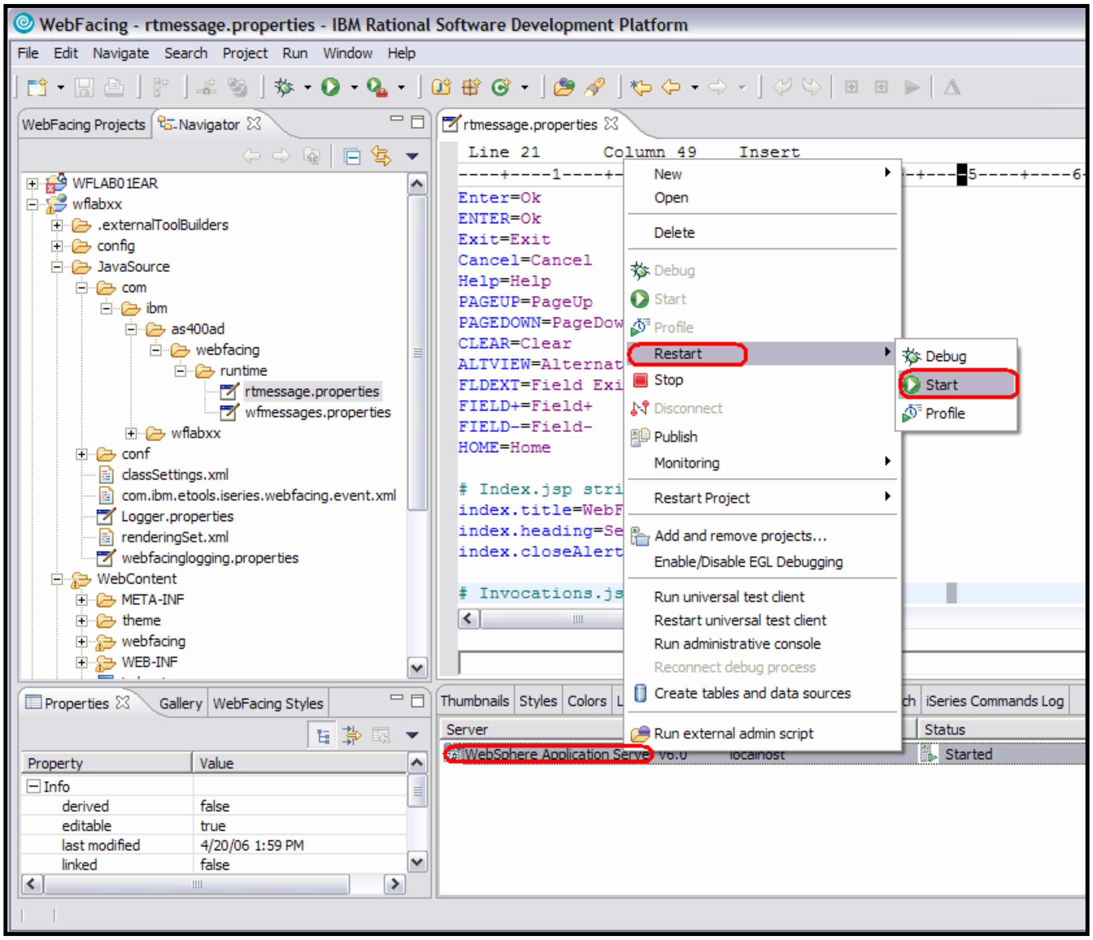
3. Changing the text of the enter key is as simple as typing in a new value on the right-hand side of the **Enter=Enter** statement. For the example, change **Enter=Enter** to **Enter=Ok**. Also, change **ENTER=Enter** to **ENTER=Ok**. Notice the other values in the file such as **Exit** (see Figure 120). Save your changes.



```
 Line 3         Column 9       Insert
----+----1----+----2----+----3----+----4----+
# Default text for function keys
Enter=Ok
ENTER=Ok
Exit=Exit
Cancel=Cancel
Help=Help
PAGEUP=PageUp
PAGEDOWN=PageDown
CLEAR=Clear
ALTVIEW=Alternate View
FLDEXT=Field Exit
FIELD+=Field+
FIELD-=Field-
HOME=Home

# Index.jsp strings
index.title=WebFacing!
```

Figure 120: Change text of enter key by typing in new value on right side of Enter=Enter statement.

4. When you have completed and saved your changes, restart the test environment by clicking on the server tab, selecting your server, right-clicking and selecting **Restart** server (see Figure 121).



```
----+----1----+-                              -+---5----+----6-
Enter=Ok
ENTER=Ok
Exit=Exit
Cancel=Cancel
Help=Help
PAGEUP=PageUp
PAGEDOWN=PageDow
CLEAR=Clear
ALTVIEW=Alternat
FLDEXT=Field Exi
FIELD+=Field+
FIELD-=Field-
HOME=Home

# Index.jsp stri
index.title=WebF
index.heading=Se
index.closeAlert

# Invocations.js
```

Property	Value
Info	
derived	false
editable	true
last modified	4/20/06 1:59 PM
linked	false

Figure 121. Restart the test environment

5. When the test environment has restarted, switch back to the WebFacing Projects WebFacing perspective and test your application. **Enter** must now be replaced with **Ok** (see Figure 122).

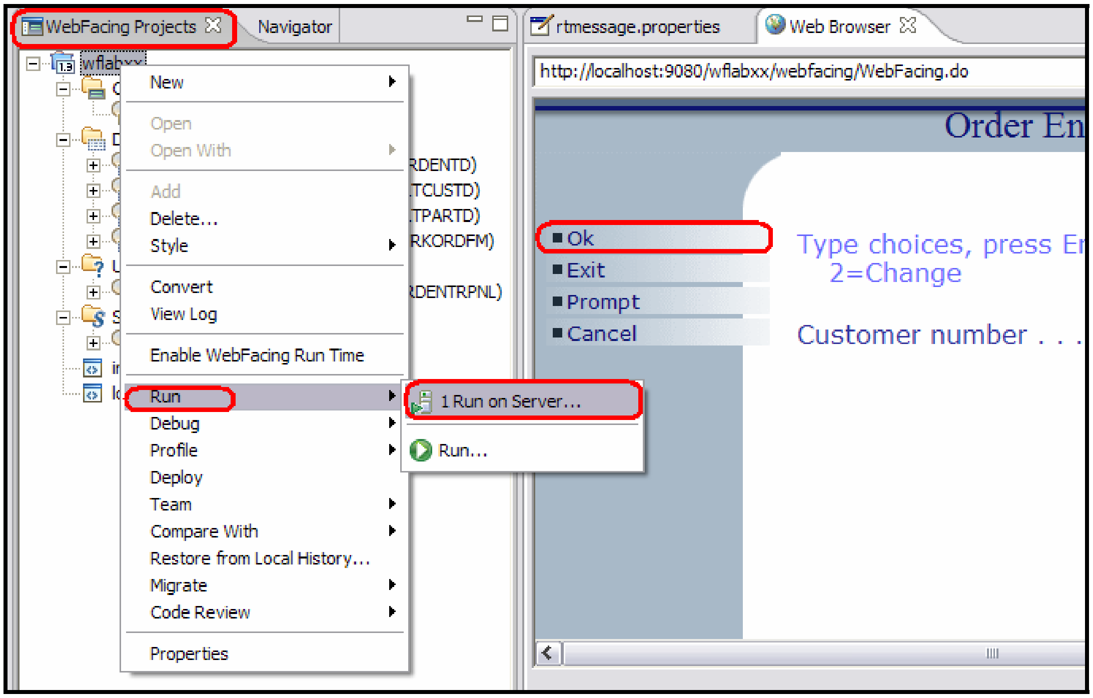


Figure 122. Switch back to WebFacing Projects

Exercise 7.5: How to change the field exit key

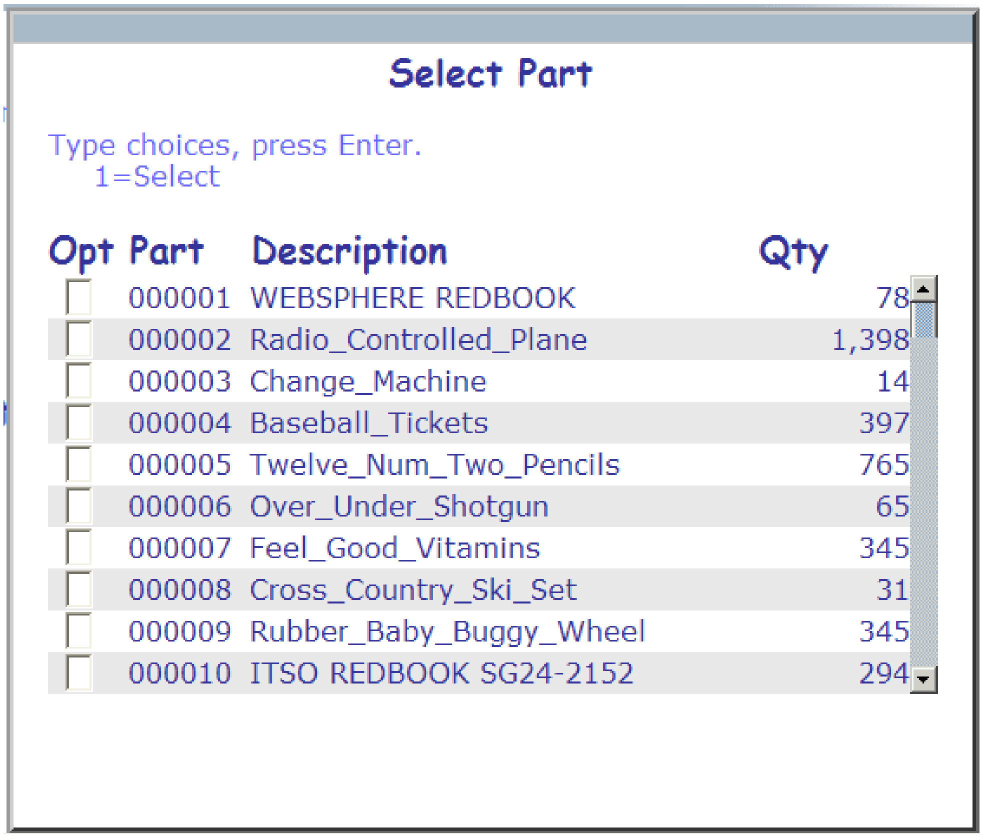
By default, the IBM WebFacing Tool uses CTRL as the field exit key. You might prefer to use other keys as field exit keys. For example, you might use the numeric key pad if have an application in which the **+** key functions as a field exit key.

Note: This example uses the numeric keypad located on most full size keyboards. If you are using a laptop that is equipped with an integrated keypad, you need to enable it, usually done by just turning on NumLock.

Verify Field exit key

To modify the Field exit key, perform these steps:

1. Verify that CTRL is the default exit key by running the Order Entry application. In the **Select Part** subfile, press the CTRL key. Notice that the **Opt** entry field goes to the next field when the CTRL key is pressed (see Figure 123).



Select Part

Type choices, press Enter.
  1=Select

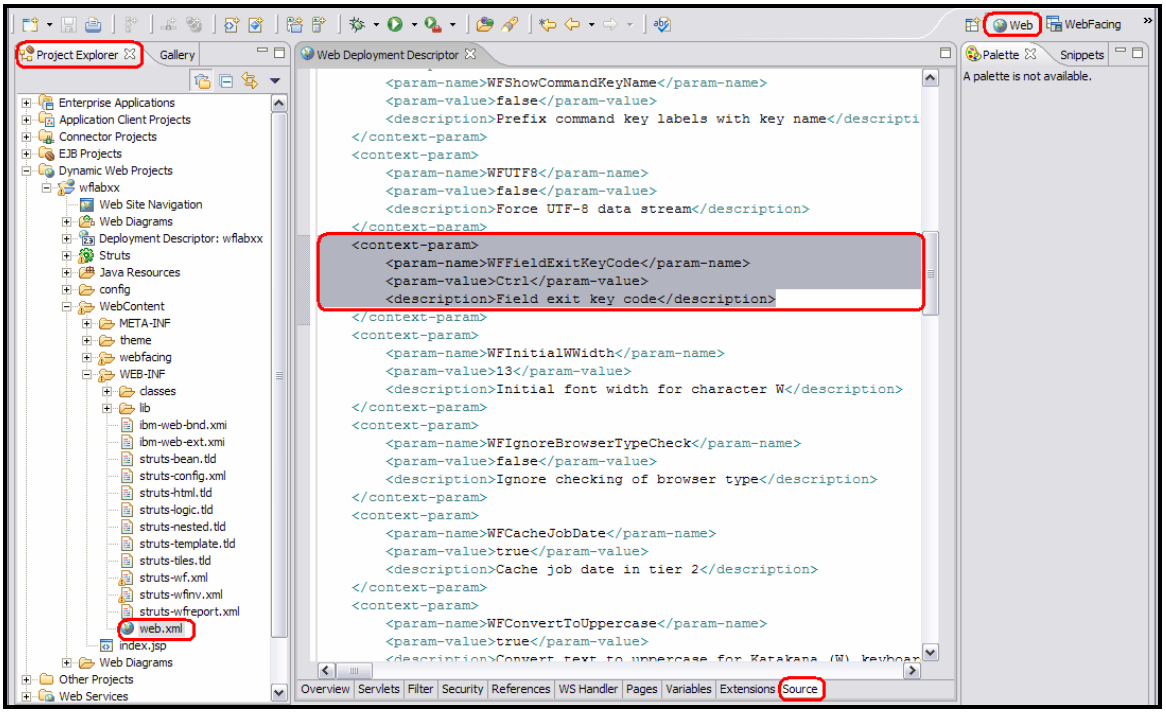
Opt	Part	Description	Qty
□	000001	WEBSPHERE REDBOOK	78
□	000002	Radio_Controlled_Plane	1,398
□	000003	Change_Machine	14
□	000004	Baseball_Tickets	397
□	000005	Twelve_Num_Two_Pencils	765
□	000006	Over_Under_Shotgun	65
□	000007	Feel_Good_Vitamins	345
□	000008	Cross_Country_Ski_Set	31
□	000009	Rubber_Baby_Buggy_Wheel	345
□	000010	ITSO REDBOOK SG24-2152	294

*Figure 123. **Select Part** subfile*

Redefine the Field exit key

To redefine the Field exit keys:

1. **Field exit keys** can be redefined in the projects web.xml file. Switch to the Web Perspective and open the web.xml file by double clicking on it in the Dynamic Web Projects/wflabxx/WebContent/WEB-INF directory. When it is open, switch to the source view by clicking on the source tab at the bottom of the window. Search for the WFFieldExitKeyCode parameter as shown below (see Figure 124).



```
            <param-name>WFShowCommandKeyName</param-name>
            <param-value>false</param-value>
            <description>Prefix command key labels with key name</descripti
        </context-param>
        <context-param>
            <param-name>WFUTF8</param-name>
            <param-value>false</param-value>
            <description>Force UTF-8 data stream</description>
        </context-param>
        <context-param>
            <param-name>WFFieldExitKeyCode</param-name>
            <param-value>Ctrl</param-value>
            <description>Field exit key code</description>
        </context-param>
        <context-param>
            <param-name>WFInitialWWidth</param-name>
            <param-value>13</param-value>
            <description>Initial font width for character W</description>
        </context-param>
        <context-param>
            <param-name>WFIgnoreBrowserTypeCheck</param-name>
            <param-value>false</param-value>
            <description>Ignore checking of browser type</description>
        </context-param>
        <context-param>
            <param-name>WFCacheJobDate</param-name>
            <param-value>true</param-value>
            <description>Cache job date in tier 2</description>
        </context-param>
        <context-param>
            <param-name>WFConvertToUppercase</param-name>
            <param-value>true</param-value>
            <description>Convert text to uppercase for Katakana (W) keyboar
```

*Figure 124. **Field Exit Keys** can be redefined.*

2. To change the exit code, you substitute CTRL with the numeric key code for the + sign. The numeric value for the + sign is 107. Other key code values can be found by searching on Google for “key codes in Internet Explorer.” One of the Web sites is: www.parkenet.com/apl/KeyCodes.htm.

   Modify the web.xml file changing the value from **Ctrl** to **107**.

```
<context-param>
       <param-name>WFFieldExitKeyCode</param-name>
       <param-value>107</param-value>
       <description>Field exit key code</description>
   </context-param>
```

3. Save the changes to web.xml. Restart your test environment and test your WebFacing project. (If you do not remember how to restart your test server you can review it in exercise 7.4, step number 4.) It is necessary to restart the server because changes have been made to the web.xml file, which is read one time at server startup.

4. When the server has restarted, test the changes by running the wflabxx application. Notice that the CTRL key no longer operates as the field exit key, but that now pressing the + key operates as the field exit key.

Recap

You have completed “Adding command key rules and labels.” You now have the information to understand how to:

- Start the WebFacing Project Properties dialog
- Use the command key recognition properties page to change the F12 key command key recognition pattern for the command key label on the Customer Selection list window
- Use the command key button labels properties page to change the undefined F12 command key label
- Reconvert the application and then run the application and check the changes
- Remove unwanted buttons
- Change the text for buttons
- Select which key represents the Field exit key

Now you have a Web user interface where the green-screen look does not show through as much as after the first WebFacing conversion you did in converting selected source members. However, you might still want to add some more refinements to the appearance of your Web pages.

Continue to “Working with more style properties.”

Working with more style properties

In this chapter, you learn how to change the user interface further, using style properties. You learn to work with the IBM WebFacing Tool style properties and check the changes by refreshing the style and running the application.

Because a style is used on a project level, all Web pages in your project contain the changes you apply to a particular style.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 8.1: Starting the WebFacing properties style dialog
- Exercise 8.2: Changing the appearance of the subfile
- Exercise 8.3: Customizing the appearance of command keys
- Exercise 8.4: Checking the new style

The exercises in this chapter must be completed in order. Start with “Exercise 8.1: Starting the WebFacing properties style dialog” when you are ready to begin.

Length of time

This chapter takes approximately 10 minutes to complete.

Exercise 8.1: Starting the WebFacing properties style dialog

First, you use the WebFacing Style properties dialog to tailor your Web user interface. You apply changes to the window layout of your screens, the subfile appearance is changed, and you enhance the command key push button looks.

To start the WebFacing properties dialog:

1. Start WebSphere Development Studio Client and open the WebFacing perspective, if it is not up and running already.
2. In the WebFacing project view, select the wflabxx project.
3. Right-click the wflabxx project, and click **Properties** on the pop-up menu.

   The Properties dialog opens (see Figure 125).

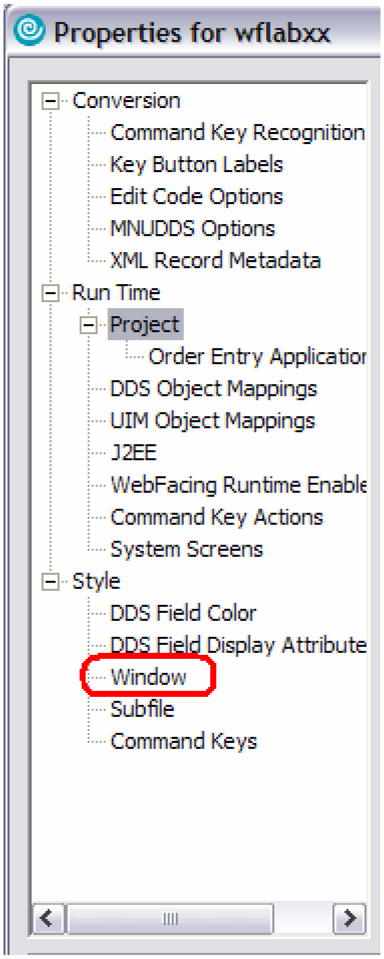
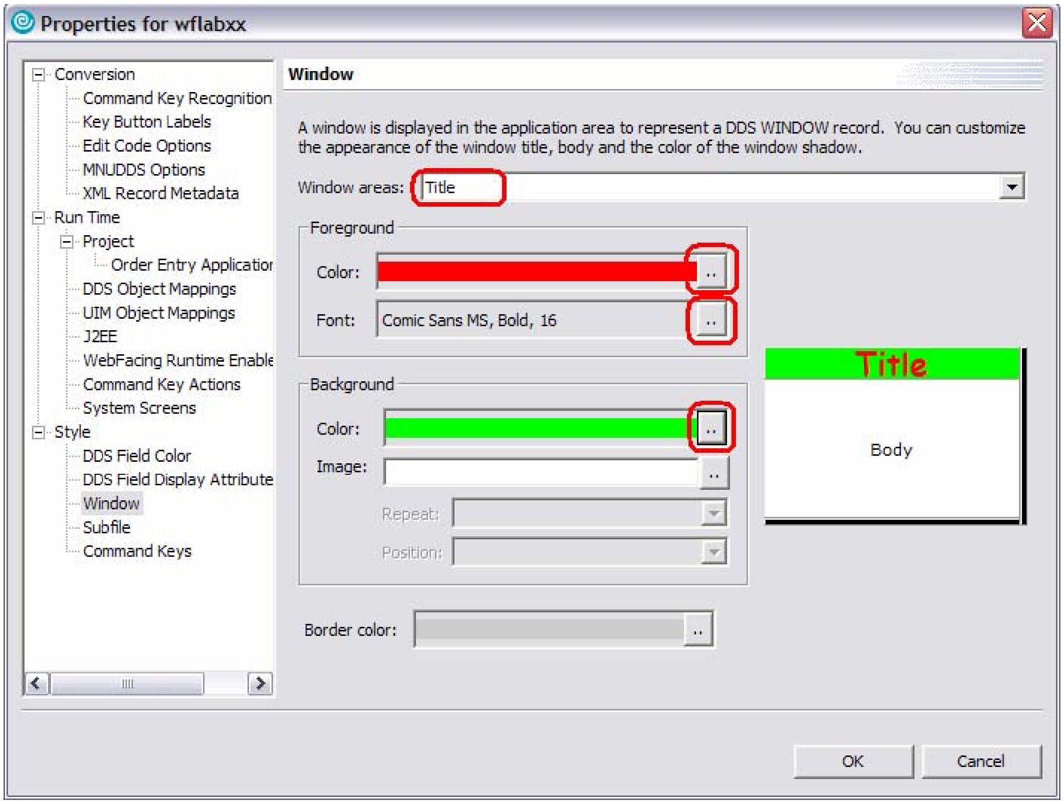


Figure 125. The Properties dialog

4. Under **Style** in the left pane of the Properties dialog, select **Window**.

   The right pane of the Properties dialog shows the Window properties (see Figure 126).



A window is displayed in the application area to represent a DDS WINDOW record. You can customize the appearance of the window title, body and the color of the window shadow.

Window areas: Title

Foreground Color: 🟥

Font: Comic Sans MS, Bold, 16

Background Color: 🟩

Image:

Figure 126. Windows properties

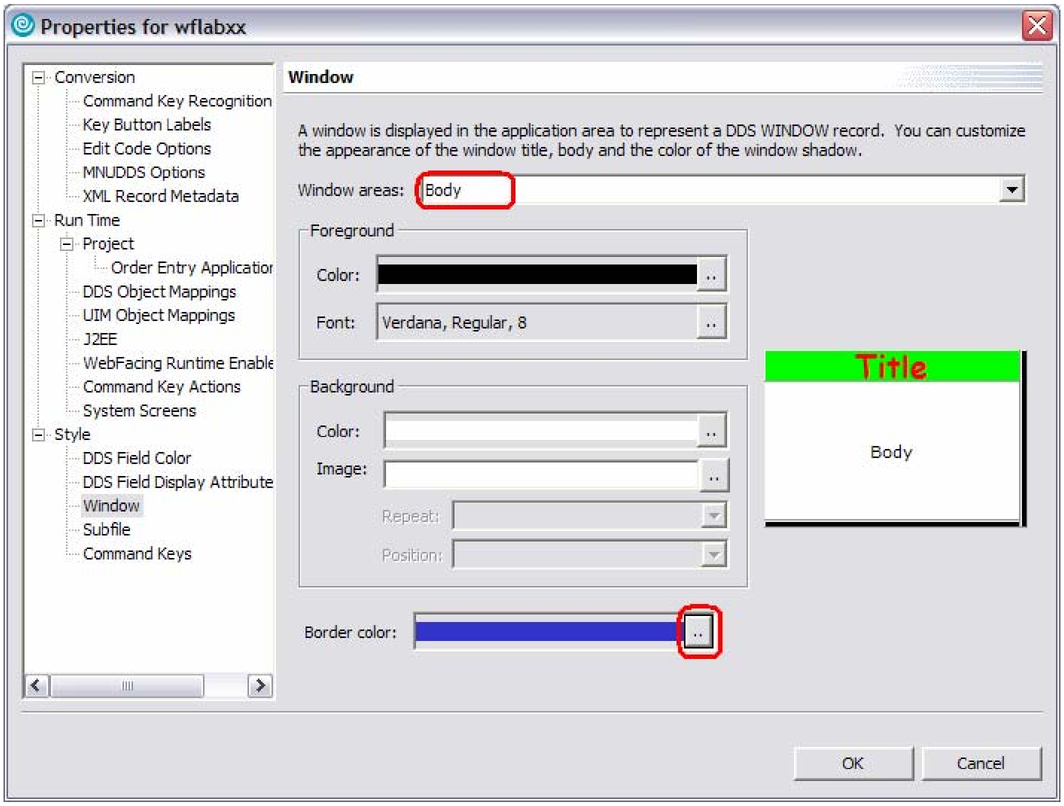
5. Select **Title** from the **Windows areas** list.
6. Under **Foreground**, click the .. push button to the right of the **Color** list.
7. On the Color Chooser dialog, click **Red**.
8. Then click **OK** in the Color Chooser dialog.

   The Window Properties page reappears.

9. Click the .. push button to the right of the **Font** list.
10. Select font **Comic Sans MS**, size **16**.
11. Click **OK** on the Font dialog.
12. Under **Background**, click the .. push button to the right of the **Color** list.
13. On the Color Chooser dialog, click **Green**.
14. Click **OK** on the Color Chooser dialog.

    The Windows properties dialog reappears.

15. Select **Body** in the **Windows areas** list at the top of the page (see Figure 127).



Window

A window is displayed in the application area to represent a DDS WINDOW record. You can customize the appearance of the window title, body and the color of the window shadow.

Window areas: Body

Foreground

Color: ■

Font: Verdana, Regular, 8

Background

Color:

Image:

Repeat:

Position:

Border color: 🟦

*Figure 127. Select **Body** in the Windows areas list.*

16. In the **Border color** list at the bottom of the page, click the .. push button to the right of the **Border color** list.

The Color Chooser dialog opens.

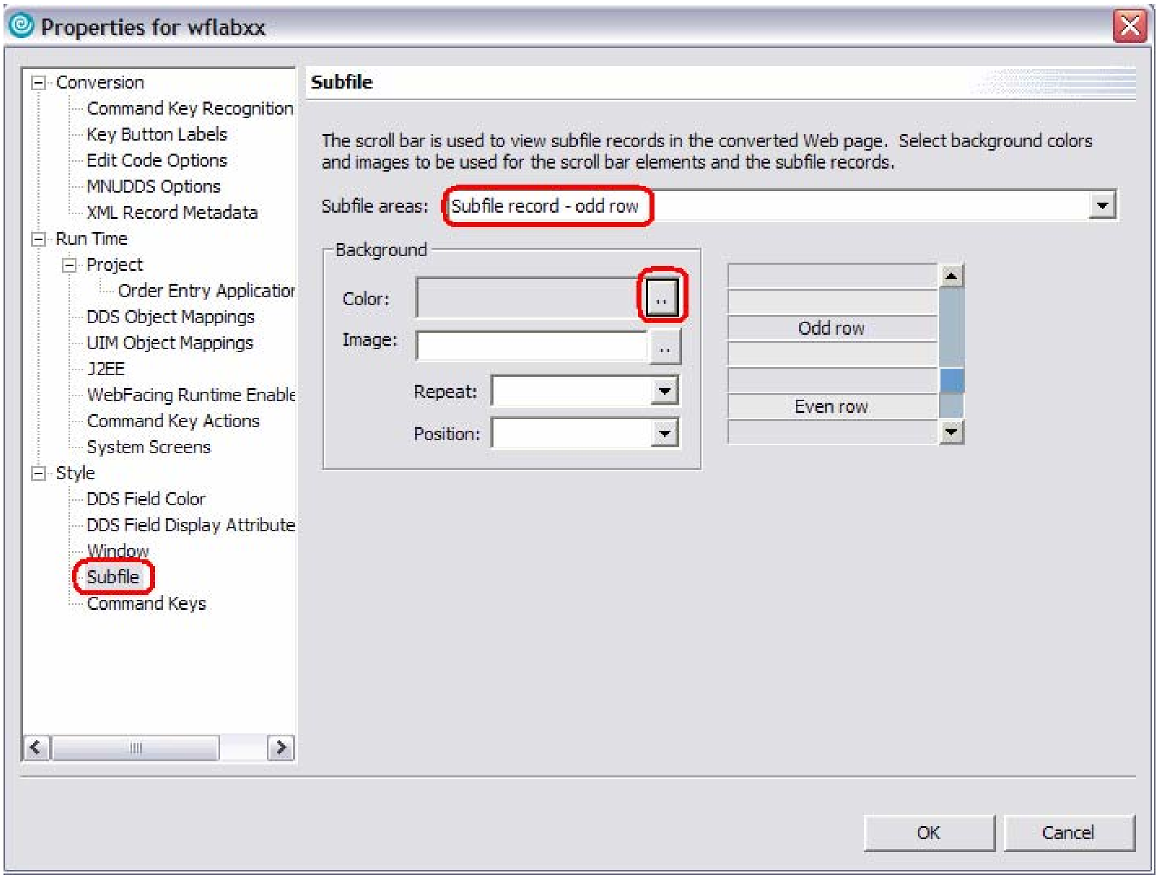
17. Select **Blue**.
18. Click **OK** in the Color Chooser dialog.

Exercise 8.2: Changing the appearance of the subfile

To change the appearance of the subfile:

1. Back in the Properties dialog, under **Style**, select **Subfile**.

   The Subfile properties page opens on the right pane of the dialog (see Figure 128).



The scroll bar is used to view subfile records in the converted Web page. Select background colors and images to be used for the scroll bar elements and the subfile records.

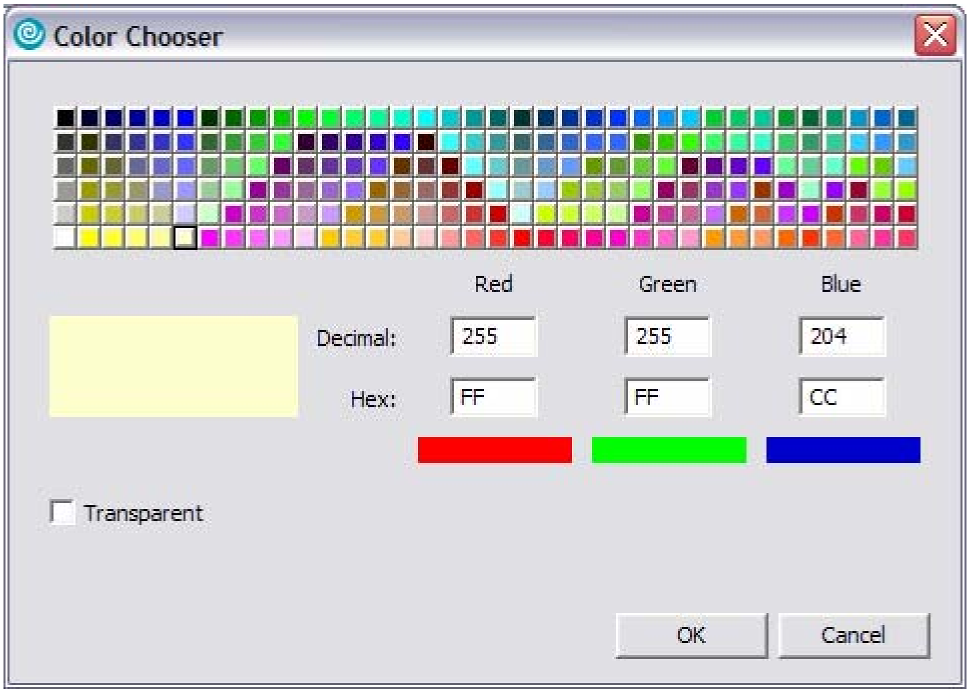
Subfile areas: Subfile record - odd row

Figure 128. Subfile properties page

2. Select **Subfile record - odd row** in the **Subfile areas** list.
3. Under Background, click the .. push button to the right of the **Color** list.

   The Color Chooser dialog opens.

4. Select the light yellow indicator box (see Figure 129).



	Red	Green	Blue
Decimal:	255	255	204
Hex:	FF	FF	CC

□ Transparent

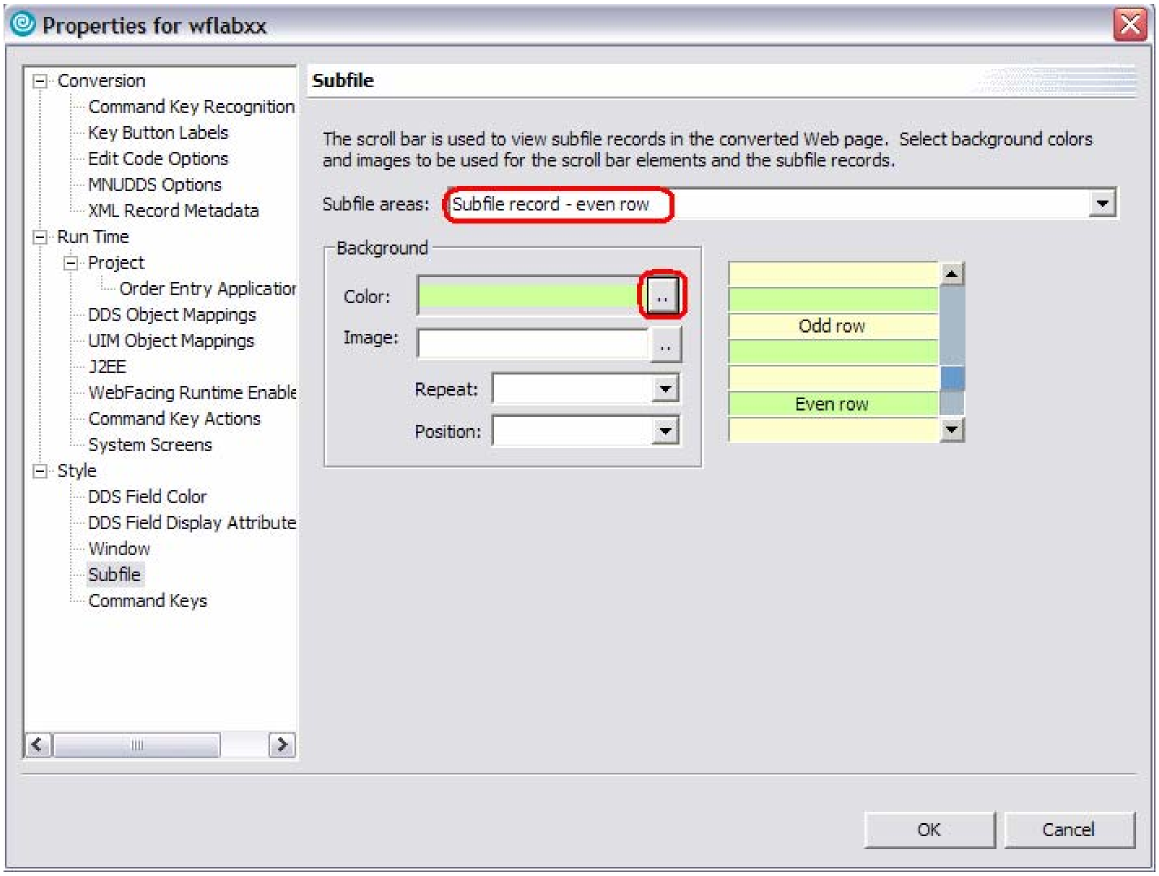
Figure 129. Select light yellow in the ***Color Chooser*** *dialog.*

Tip: If you are interested in the hex value of certain colors as they are used in the style classes, the Color Chooser dialog displays the hex values. See the values: **FF**, **FF**, and **CC** (Red, Green and Blue respectively).

5. Click **OK** in the Color Chooser dialog.

   You return to the Properties dialog.

6. Select **Subfile record - even row** in the **Subfile areas** list at the top of the page (see Figure 130).



The scroll bar is used to view subfile records in the converted Web page. Select background colors and images to be used for the scroll bar elements and the subfile records.

Subfile areas: Subfile record - even row

*Figure 130. Select **Subfile record - even row** in the **Subfile areas** list*

7. Under Background, click the .. push button to the right of the **Color** list.

   The Color Chooser dialog opens.

8. Select a light green.
9. Click **OK** on the Color Chooser dialog.

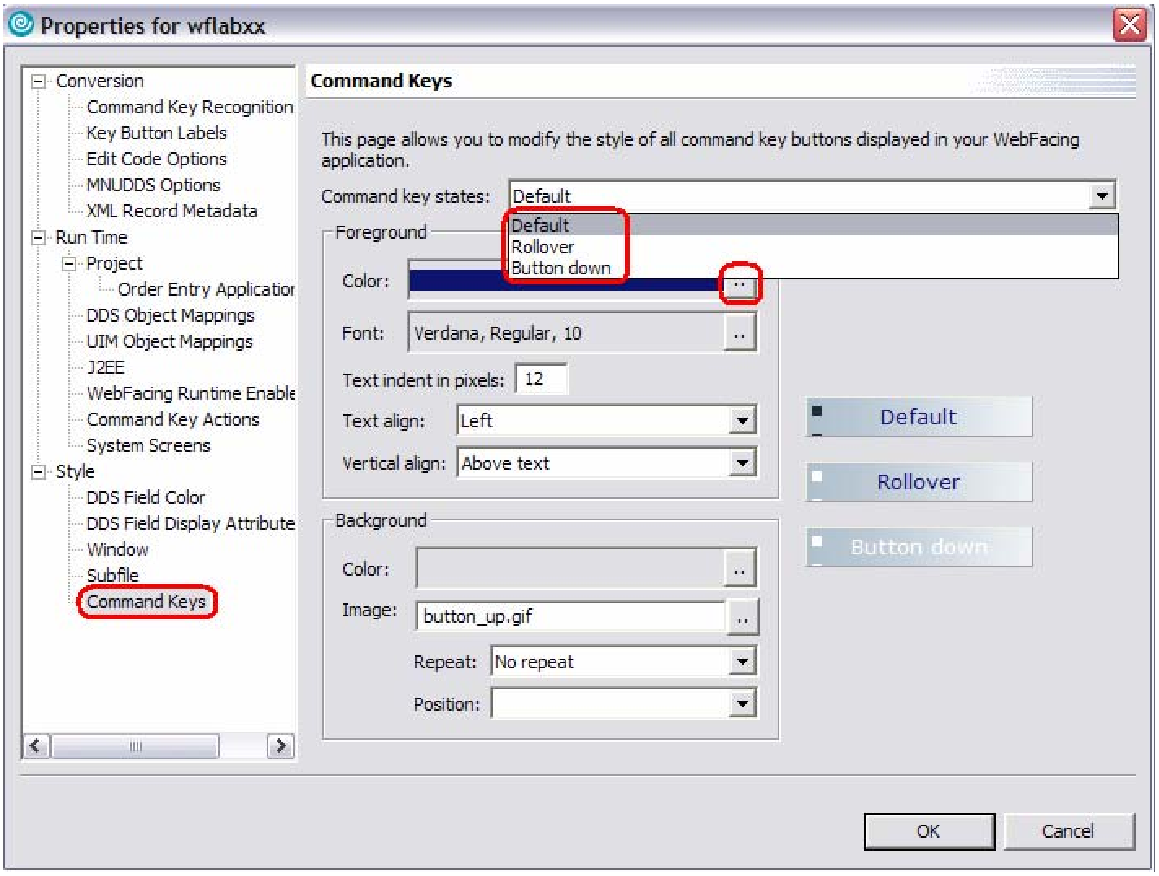
   Now, at run time, the odd and even records are displayed with different background colors. Next, you customize the appearance of the command key push buttons.

Exercise 8.3: Customizing the appearance of command keys

To customize appearance of command keys:

1. In the left pane of the Properties dialog under **Style**, select **Command keys**.

   The Command Keys properties opens on the right pane of the dialog (see Figure 131).



Properties for wflabxx

Command Keys

This page allows you to modify the style of all command key buttons displayed in your WebFacing application.

Command key states: Default

Foreground

Color:

Font: Verdana, Regular, 10

Text indent in pixels: 12

Text align: Left

Vertical align: Above text

Background

Color:

Image: button_up.gif

Repeat: No repeat

Position:

Figure 131. The Command Keys properties open.

2. Change the Foreground color for all three states of the buttons:

   - Default
   - Rollover
   - Button down

   Now you have finished customization.

3. Click **OK** on the Properties dialog.

Exercise 8.4: Checking the new style

Observe the new look of your application.

To check the new style:

1. Right-click wflabxx project.
2. Click **Run>Run on Server** on the pop-up menu.
3. Run the application.
4. Click **Prompt**.

   The application window shows different colors, and the subfile records are colored. The command key push buttons change color, depending on their state.

   If you do not see the changed colors, you need to restart the project, restart the Application server, and then run the application again.

Recap

You have completed “Working with more style properties.” You now have the information to understand how to:

- Start the Style Properties dialog
- Work with three different style properties
- Check the changes by running the application

You have tested your WebFacing application and for convenience have used your User ID and password as the default to avoid having to sign-on every time you started the application. Now you need to move the application into production and you need the WebFacing application to prompt the users for sign-on information when it starts.

Continue to “Adding authentication.”

Adding authentication

In this chapter, you learn how to secure your application by forcing authentication before the application is invoked. You learn how to change the run-time behavior of your application. Instead of using the default User ID and password, the one you used when you selected the members to convert your DDS source, you now force the user to provide their own sign-on information. You learn how to remove the User ID and password from the CL command properties.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 9.1: Removing the default User ID and password
- Exercise 9.2: Testing the authentication dialog

The exercises in this chapter must be completed in order. Start with “Exercise 9.1: Removing the default User ID and password” when you are ready to begin.

Length of time

This chapter take approximately five minutes to complete.

Exercise 9.1: Removing the default User ID and password

You use the CL command properties page of the run time to remove the default User ID and password information.

In the WebFacing perspective, within the WebFacing project view:

1. Right-click wflabxx project.
2. Click **Properties** on the pop-up menu.

   The Properties for wflabxx dialog opens.

3. Under **Run Time\Project** in the left pane of the Properties dialog, select **Order Entry Application** (see Figure 132).

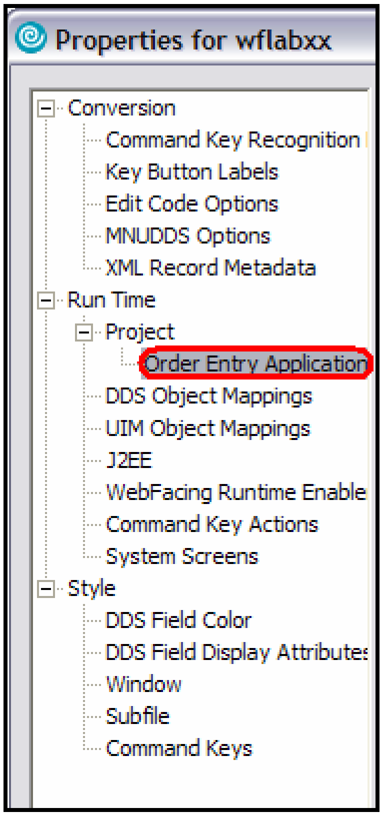
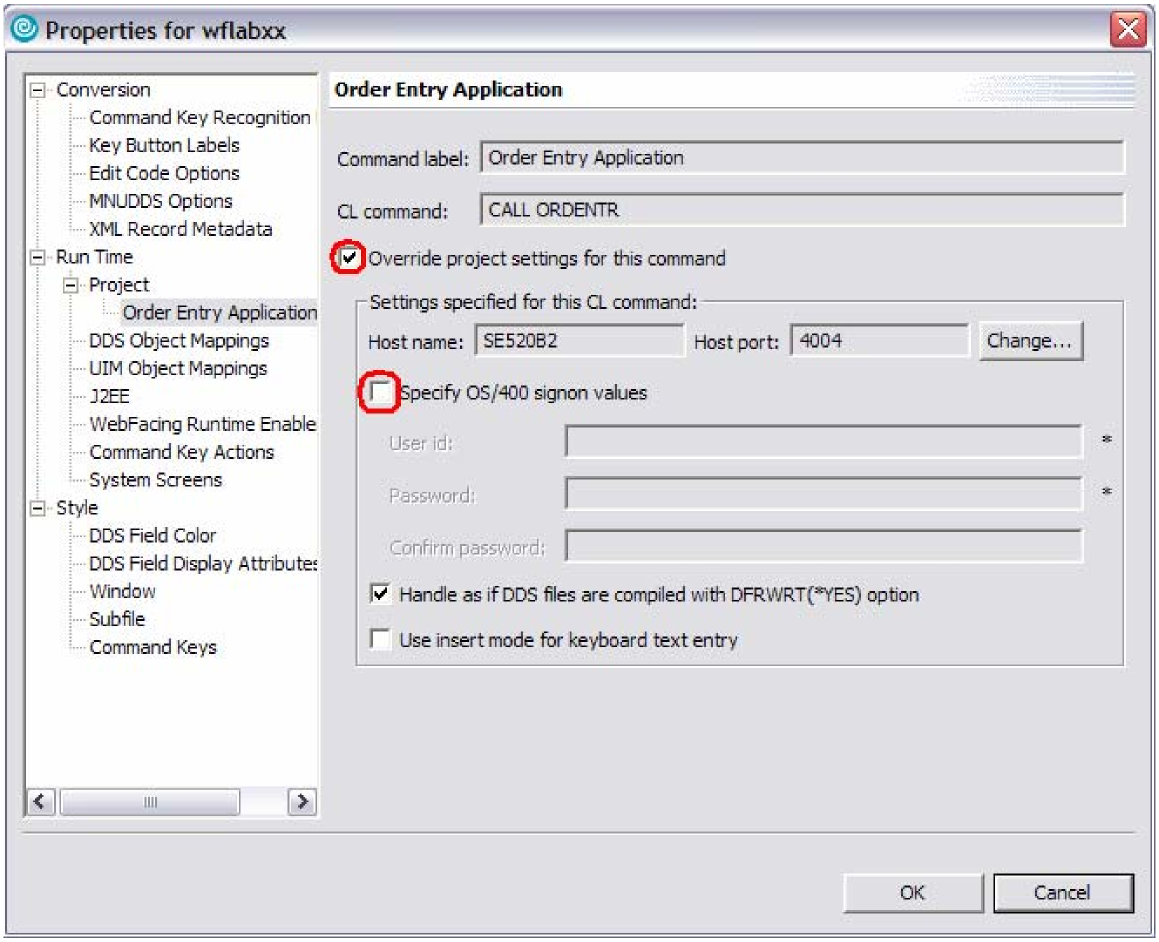


*Figure 132. Select **Order Entry Application**.*

The Order Entry application properties panel opens (see Figure 133).

4. Make sure the **Override project settings for this command** check box is selected.
5. Clear the **Specify OS/400 signon values** check box.
6. Click **OK** at the bottom of the Properties dialog.



Order Entry Application

Command label: Order Entry Application

CL command: CALL ORDENTR

☑ Override project settings for this command

Settings specified for this CL command:

Host name: SE520B2 Host port: 4004

☐ Specify OS/400 signon values

User id: *

Password: *

Confirm password:

☑ Handle as if DDS files are compiled with DFRWRT(*YES) option

☐ Use insert mode for keyboard text entry

Figure 133. Order Entry Application

You are back in the WebFacing project view. You have changed the authentication behavior for this link (application). In this WebFacing project, you only have one link, but imagine that you have multiple entry points into your application like a menu. You can specify a different authentication behavior for each of these links. On the other hand, if you have multiple links in a WebFacing project and the authentication is supposed to be the same for each different link, then you can specify an authentication rule on the project level. To do this, just go to the project properties page instead of the CL command properties page and specify the desired authentication behavior there.

Exercise 9.2: Testing the authentication dialog

As before, you run your application in the WebSphere Application Server test environment.

To test the authentication dialog:

1. Right-click wflabxx project.
2. Click **Run>Run on Server** on the pop-up menu.
3. Click the **Order Entry** link.

   You see the authentication dialog (see Figure 134).

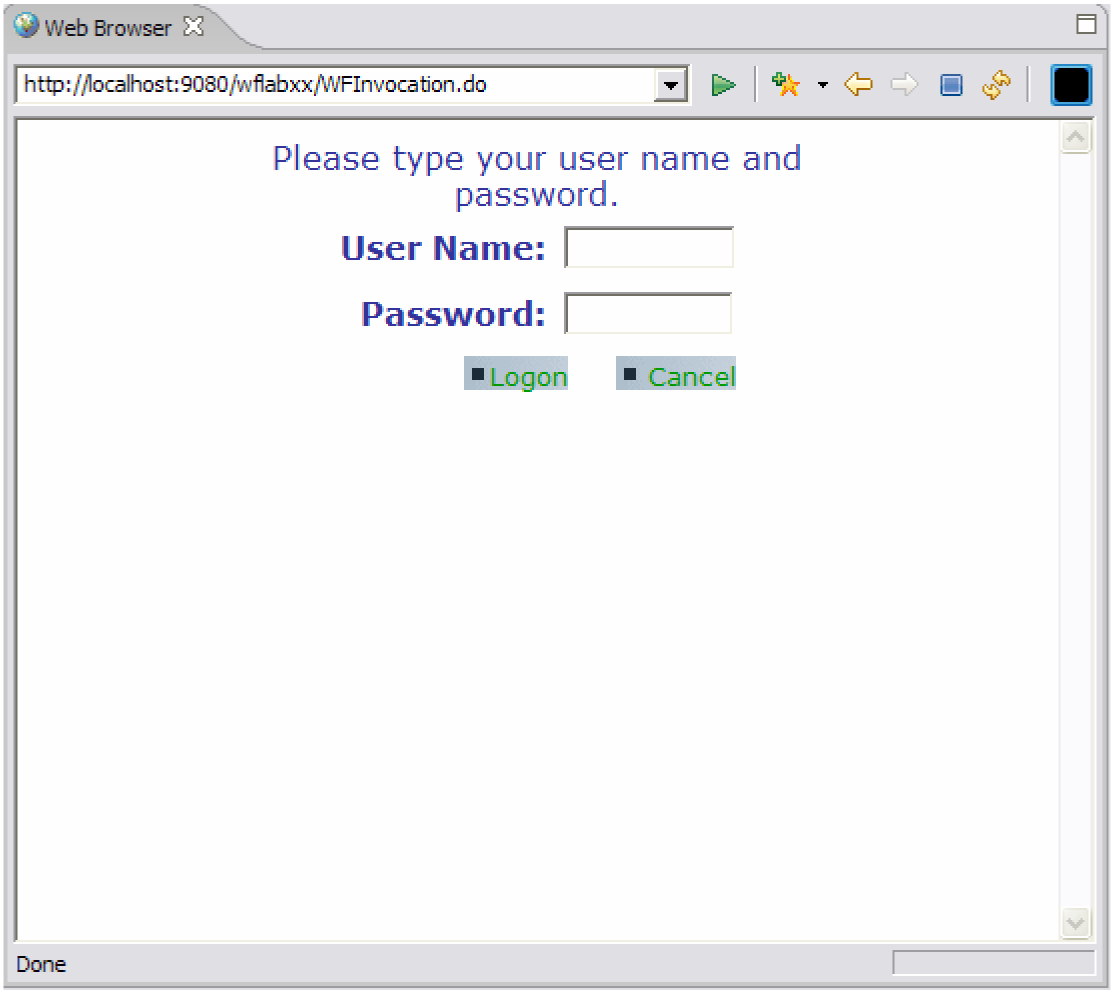

Web Browser
http://localhost:9080/wflabxx/WFInvocation.do
Please type your user name and password.
User Name:
Password:
Logon
Cancel
Done

Figure 134. The authentication dialog

4. Specify your **User Name** and **Password**.
5. Click the **Logon** push button on the Authentication dialog.

   The application runs as before, if the job environment for the User ID you specified is set up correctly for this application.

Note: If the LOGON dialog does not appear, restart the Application Server.

1. Click the **Servers** tab at the bottom view in the workbench.
2. Right-click the server icon.
3. Click **Restart>Start** on the pop-up menus (see Figure 135).

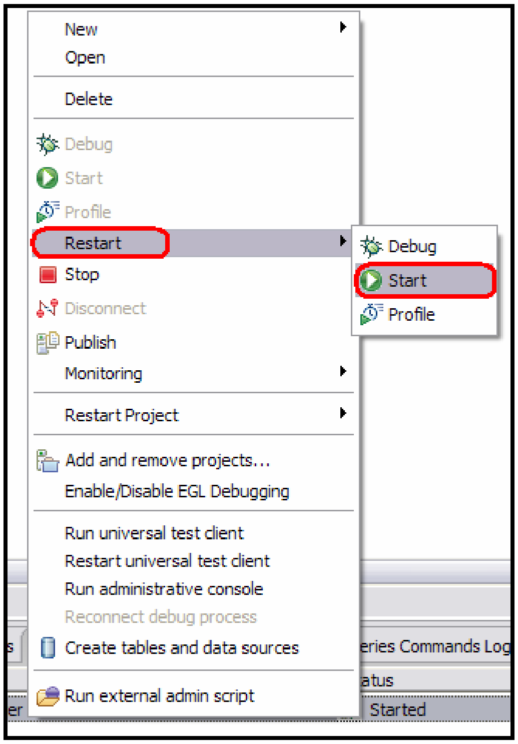

New
Open
Delete
Debug
Start
Profile
Restart
Stop
Disconnect
Publish
Monitoring
Restart Project
Add and remove projects...
Enable/Disable EGL Debugging
Run universal test client
Restart universal test client
Run administrative console
Reconnect debug process
Create tables and data sources
Run external admin script
Debug
Start
Profile
eries Commands Log
atus
Started

*Figure 135. Click **Restart>Start** on the pop-up menus.*

4. Run the application again.

Recap

You have completed “Adding authentication.” You now have the information to understand how to accomplish the following:

- Use the project properties dialog for the Order Entry application to remove the default User ID and password information.
- Run the application and where forced to provide authentication before the WebFacing run time starts the WebFacing application.

You have created a WebFacing application and now want to enhance the look of the index.jsp page to give the start page of this application a professional appearance. The WebSphere Development Studio Client workbench contains several tools to design and enhance Web pages. Also, with the workbench you have access to sample pictures, icons, animations, and more. You use some of these tools and samples and enhance the page so it becomes more attractive.

Continue to “Enhancing index.jsp page.”

Enhancing index.jsp page

In this chapter, you learn how to use the Web tools in the WebSphere Development Studio Client workbench to update the index.jsp file. The index.jsp file created by the IBM WebFacing Tool, which allows you to start your WebFacing application, is plain. In this chapter, you add some color to it, as well as some pictures to make it a more interesting Web page. You learn how to test the changed page.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 10.1: Opening Page Designer
- Exercise 10.2: Working with page properties
- Exercise 10.3: Linking a cascading style sheet to a Web page
- Exercise 10.4: Designing and adding a logo
- Exercise 10.5: Adding a heading 1 tag to the page
- Exercise 10.6: Adding a picture to the page
- Exercise 10.7: Adding moving text to the page
- Exercise 10.8: Changing the text color
- Exercise 10.9: Deleting default text from the page

The exercises in this chapter must be completed in order. Start with “Exercise 10.1: Opening Page Designer” when you are ready to begin.

Length of time

This chapter takes approximately 30 minutes to complete.

Exercise 10.1: Opening Page Designer

To open Page Designer:

1. Open the WebFacing perspective, if not already open.

   You have the WebFacing Projects view in your workbench environment (see Figure 136).

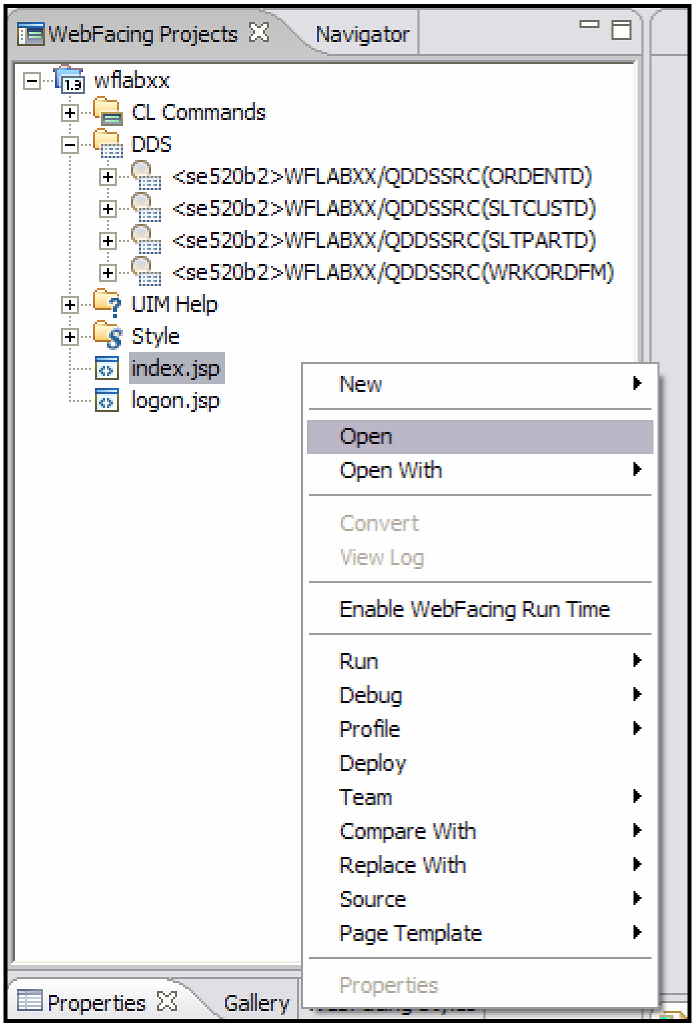


Figure 136. WebFacing Projects open

2. Expand the wflabxx WebFacing project.
3. Right-click the index.jsp icon.
4. Click **Open** on the pop-up menu.

The Page Designer opens in the upper right pane of the workbench and shows the index.jsp page as the IBM WebFacing Tool created it (see Figure 137).

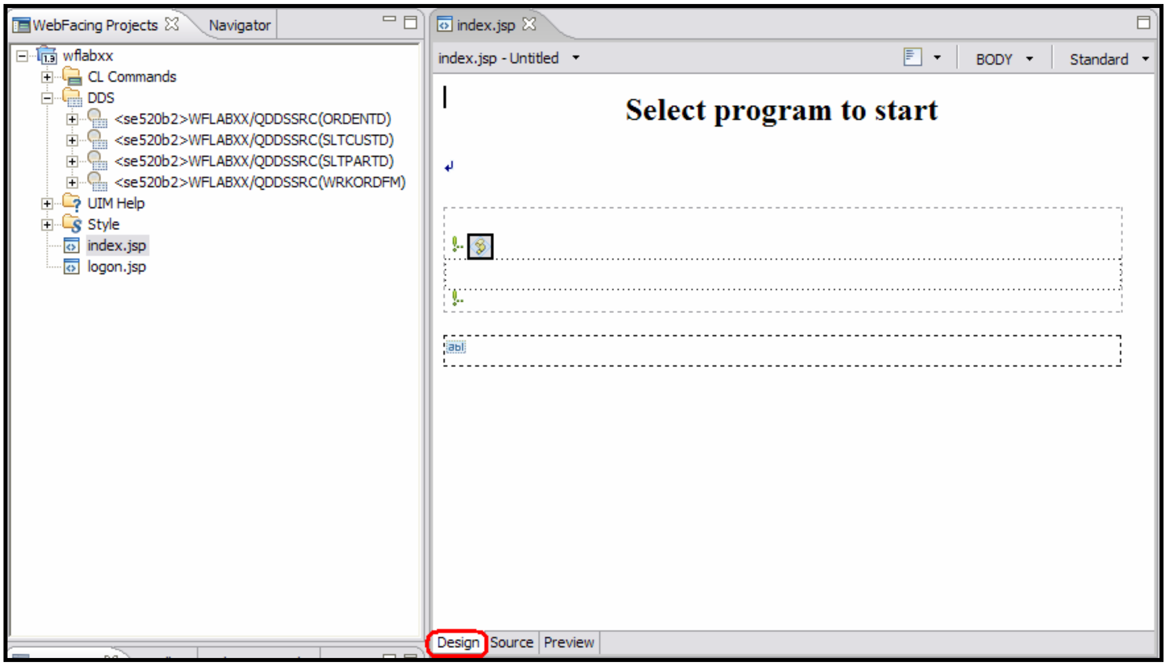


Figure 137. The ***index.jsp*** *page*

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

5. Click the **Design** tab.

Exercise 10.2: Working with page properties

Next, you insert some blank lines at the top of our page design. This moves the existing links further down the web page panel.

To work with page properties:

1. Position your cursor for editing by clicking at the first position of the line containing the **Select program to start** title text (see Figure 138).

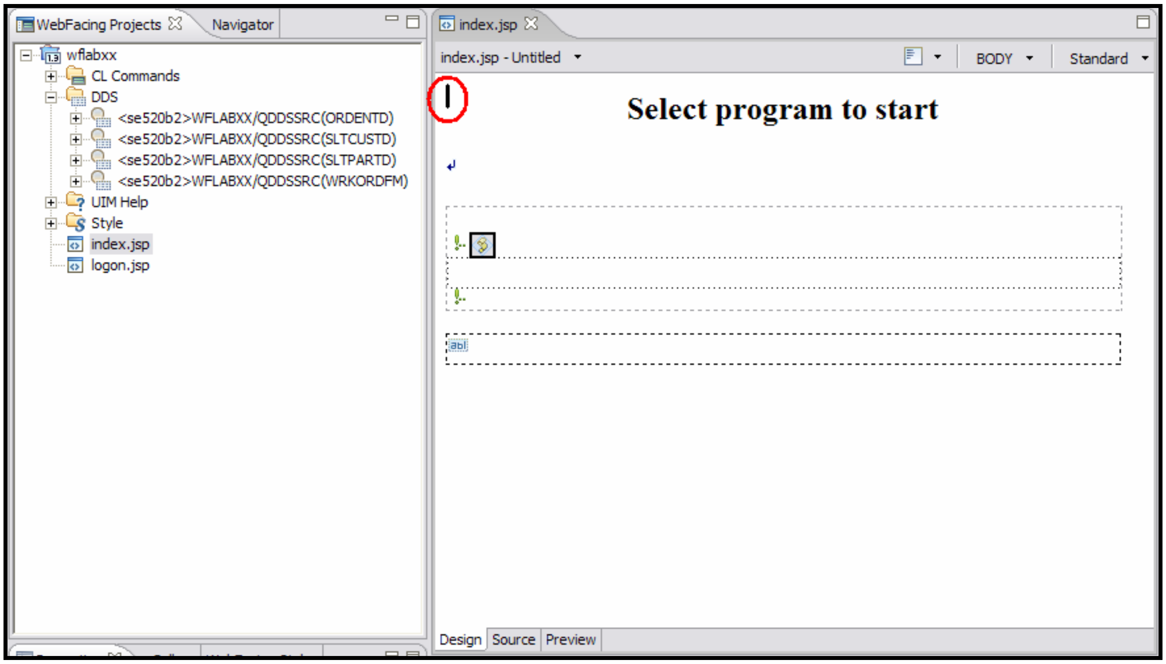


*Figure 138. Position cursor at **Select program to start** title text.*

2. Press the **Enter** key a number of times (see Figure 139).

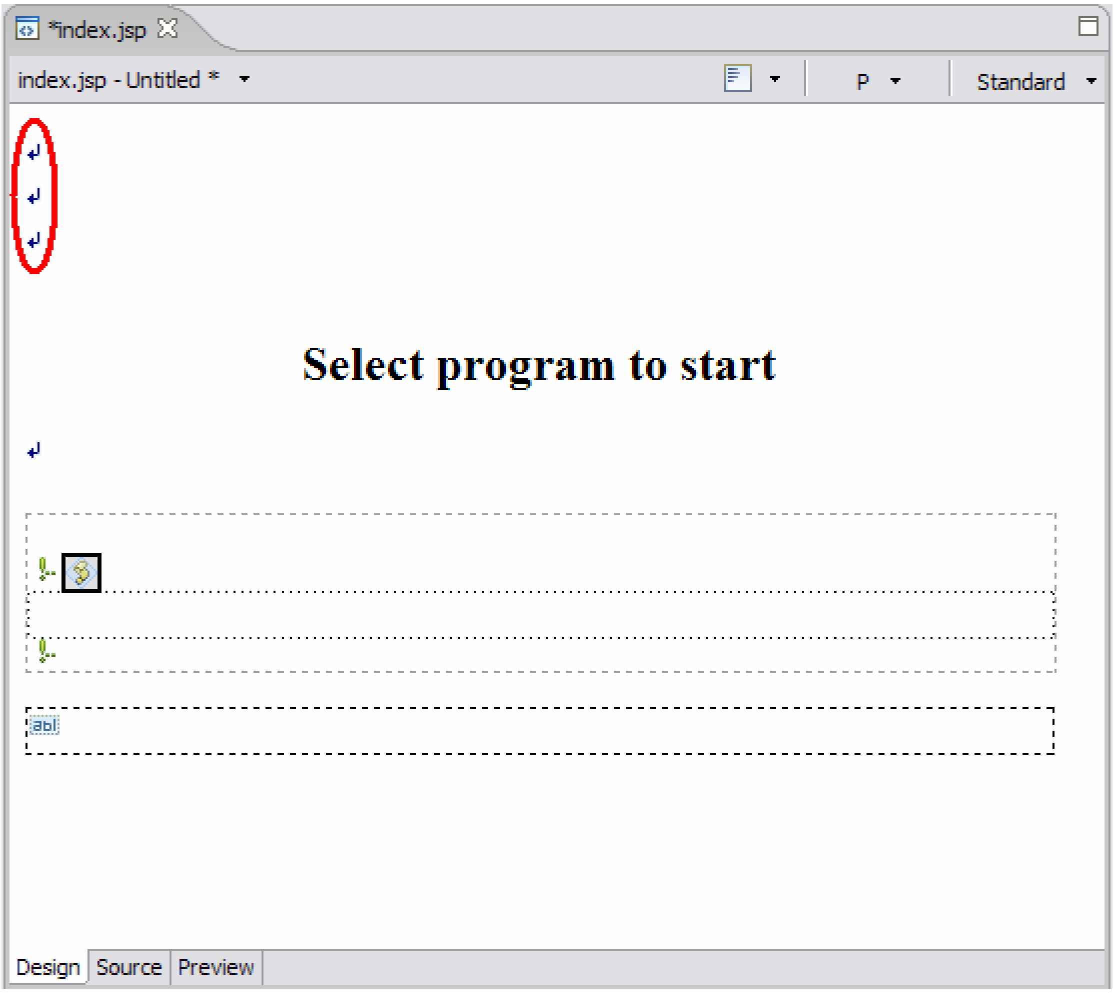


*Figure 139. Press the **Enter** key a number of times.*

3. Right-click the background of the index.jsp page in Page Designer.

4. Click **Page Properties** on the pop-up menu (see Figure 140).

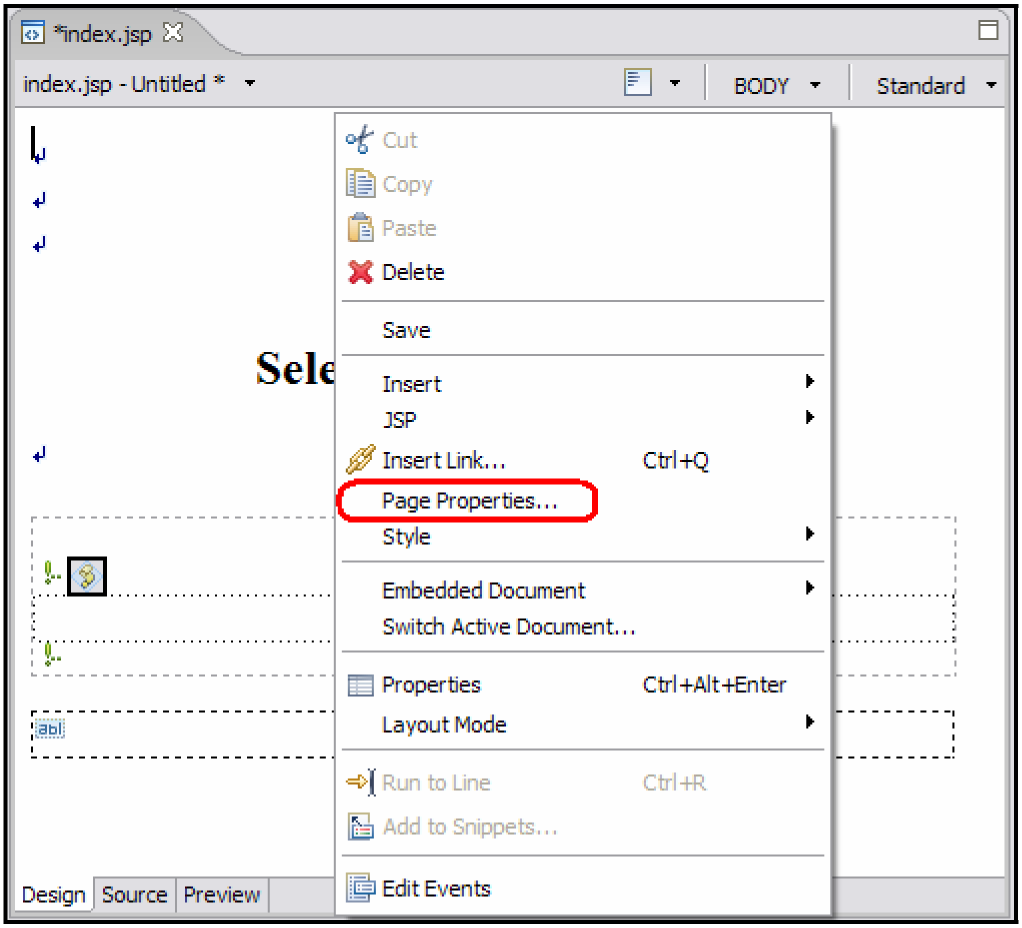
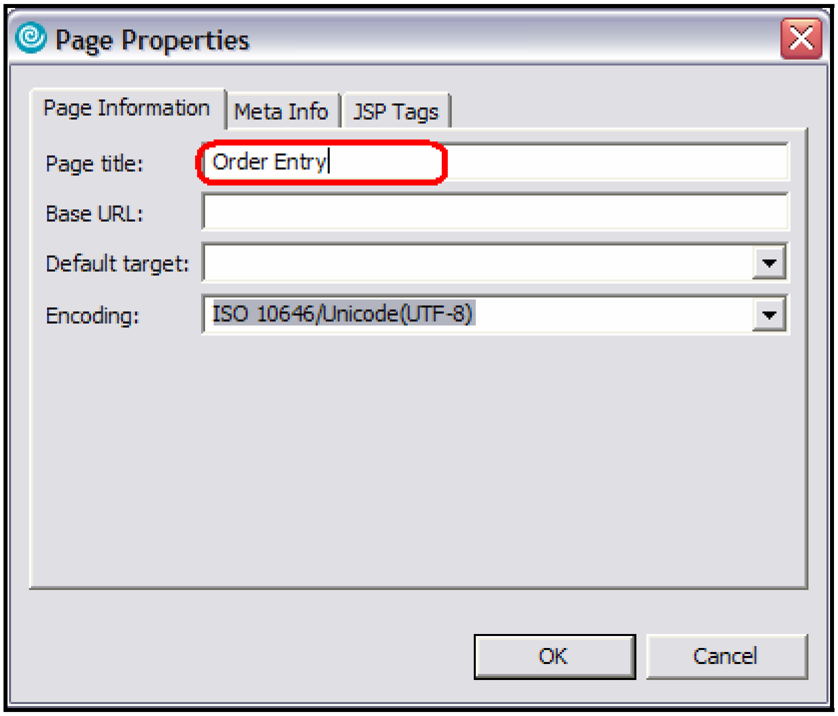


Figure 140. Page Properties

The Page Properties dialog opens (see Figure 141).



Page title: Order Entry
Base URL:
Default target:
Encoding: ISO 10646/Unicode(UTF-8)

Figure 141. Page Properties dialog

5. In the **Page title** field, type Order Entry.
6. Click **OK** on the Page Properties dialog.

When this page is shown in a browser, the Window title bar of the browser now displays **Order Entry**.

Exercise 10.3: Linking a cascading style sheet to a Web page

Now, you add a style to the Web page. You can use a style that is used in your company or you can use one of the sample style sheets that are provided in WebSphere Development Studio Client.

Open the Gallery view in the workbench.

To link a style sheet, follow steps 1 through 8 (see Figure 142):

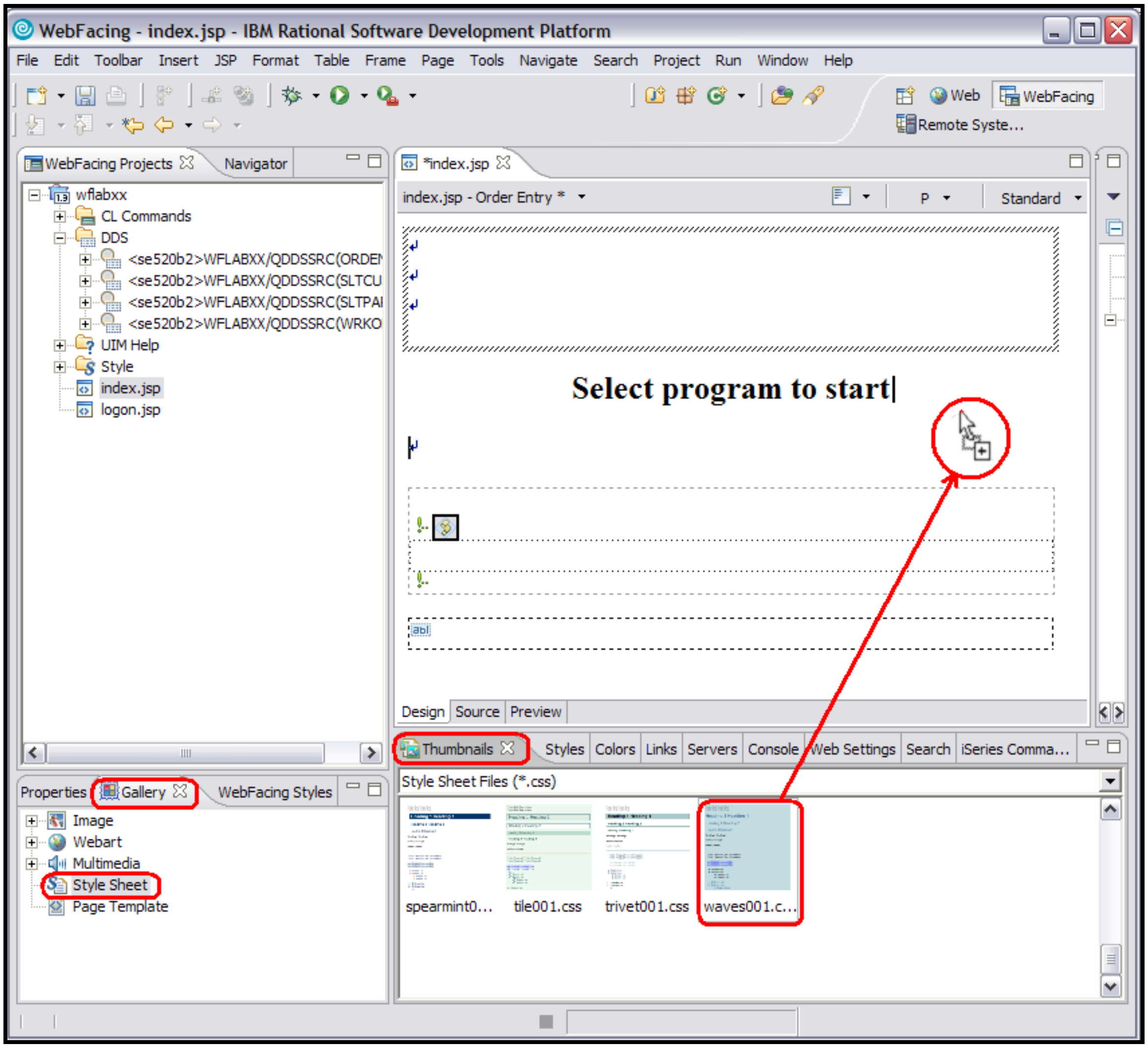


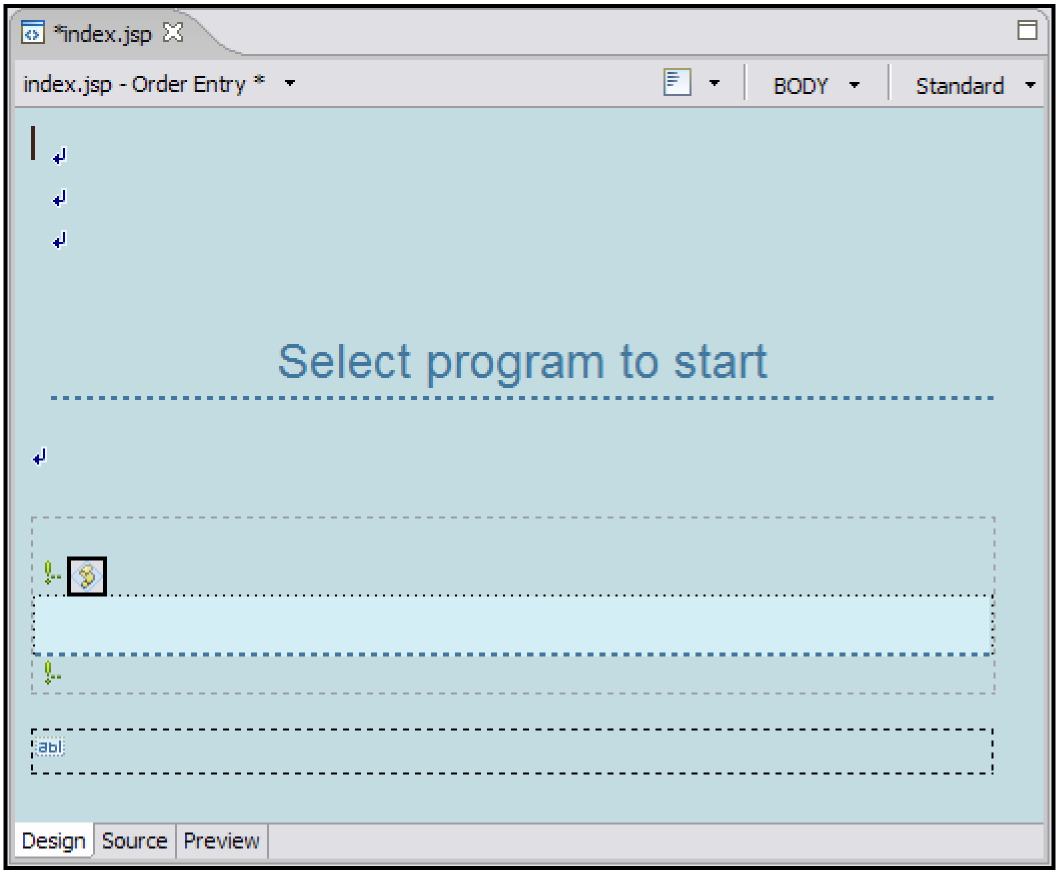
Figure 142. Linking a style sheet

1. Click the **Gallery** tab, if the Gallery view is not already displayed.
2. Scroll to the bottom of the Gallery view, until you can see the **Style Sheet** icon.
3. Click the **Style Sheet** icon, to select it.
4. Click the **Thumbnails** tab on the right bottom pane in the workbench.

   You see thumbnail icons of all the styles available.

5. In the thumbnail view, scroll down to the bottom, until you see style sheet waves001.css in the list, or select a style sheet that you like best.
6. Click the thumbnail picture of waves001.css.
7. Hold the mouse button down and drag the cursor to the Page Designer window.
8. The cursor changes from this shape ⊘, to this shape . When the latter cursor shape appears in the Page Designer window, release the left mouse button.

   After a short time, the style sheet properties are applied and the colors in the page changes to the style sheet definitions (see Figure 143).



Select program to start

Figure 143. Style sheet properties applied with new color

Exercise 10.4: Designing and adding a logo

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

Starting the WebArt Designer

To start the WebArt Designer, perform these two steps (see Figure 144):

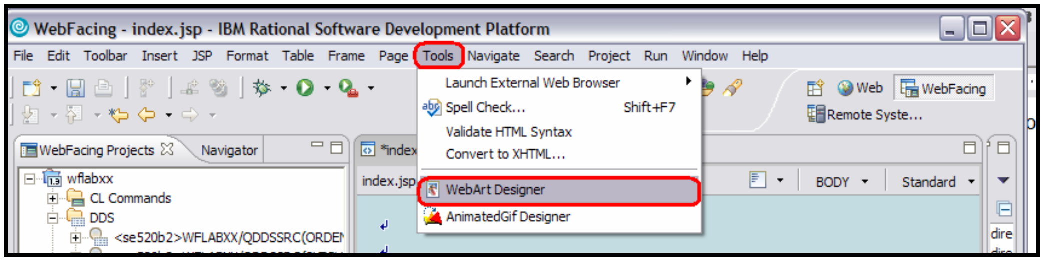

WebFacing - index.jsp - IBM Rational Software Development Platform
File Edit Toolbar Insert JSP Format Table Frame Page Tools Navigate Search Project Run Window Help
Launch External Web Browser
Spell Check... Shift+F7
Validate HTML Syntax
Convert to XHTML...
WebArt Designer
AnimatedGif Designer
Web
WebFacing
Remote Syste...
WebFacing Projects
Navigator
wflabxx
CL Commands
DDS
<se520b2>WFLABXX/QDDSSRC(ORDEI
index.jsp
BODY
Standard

Figure 144. Starting the WebArt Designer

1. Click **Tools** in the workbench menu.
2. Click **WebArt Designer** on the pop-up menu.

The WebSphere Studio WebArt Designer opens (see Figure 145).

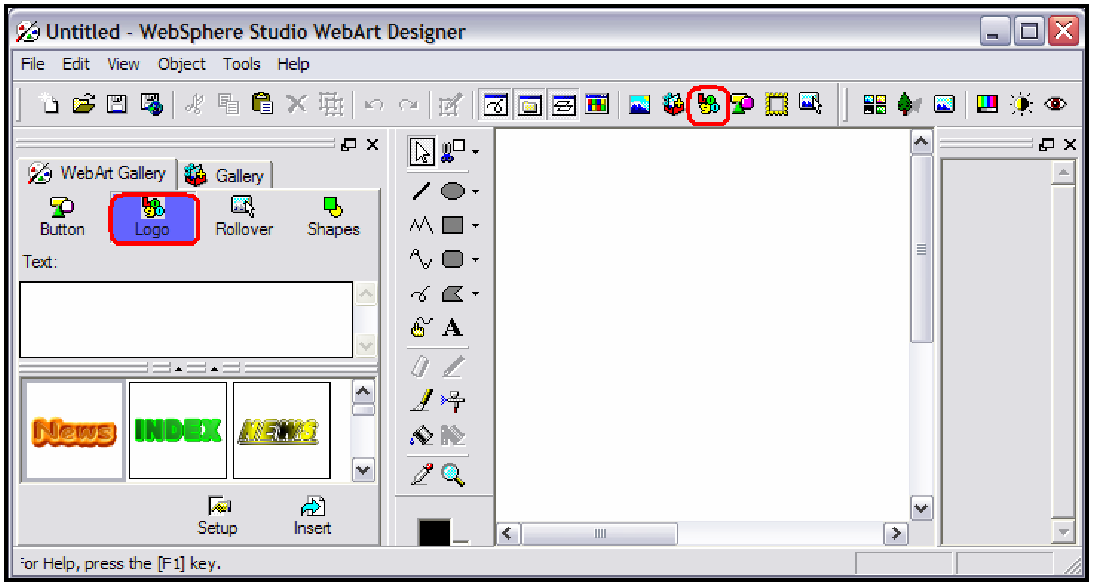

Untitled - WebSphere Studio WebArt Designer
File Edit View Object Tools Help
WebArt Gallery
Gallery
Button
Logo
Rollover
Shapes
Text:
News
INDEX
NEWS
Setup
Insert
For Help, press the [F1] key.

Figure 145. The WebSphere Studio WebArt Designer opens.

The WebArt Designer shows the Template Gallery on the left, where there are samples of logos, buttons, rollovers, images and more. The big white area in the middle of the dialog is the canvas that is used to work with objects that you want to create or change. Although you can select a logo from the template gallery as the base for your own logo, in this exercise, you create your own, instead.

Creating the logo

To create the logo:

1. Click the **Create Logo** button above the canvas or use the **Object > Create Logo** menu options (see Figure 146).

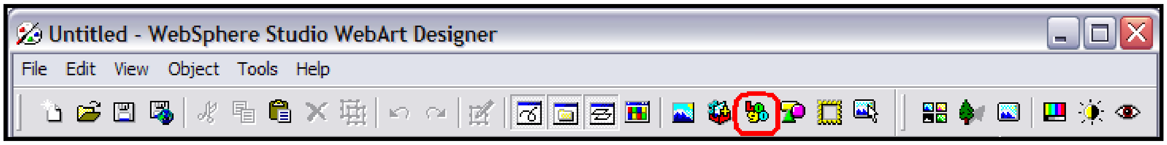
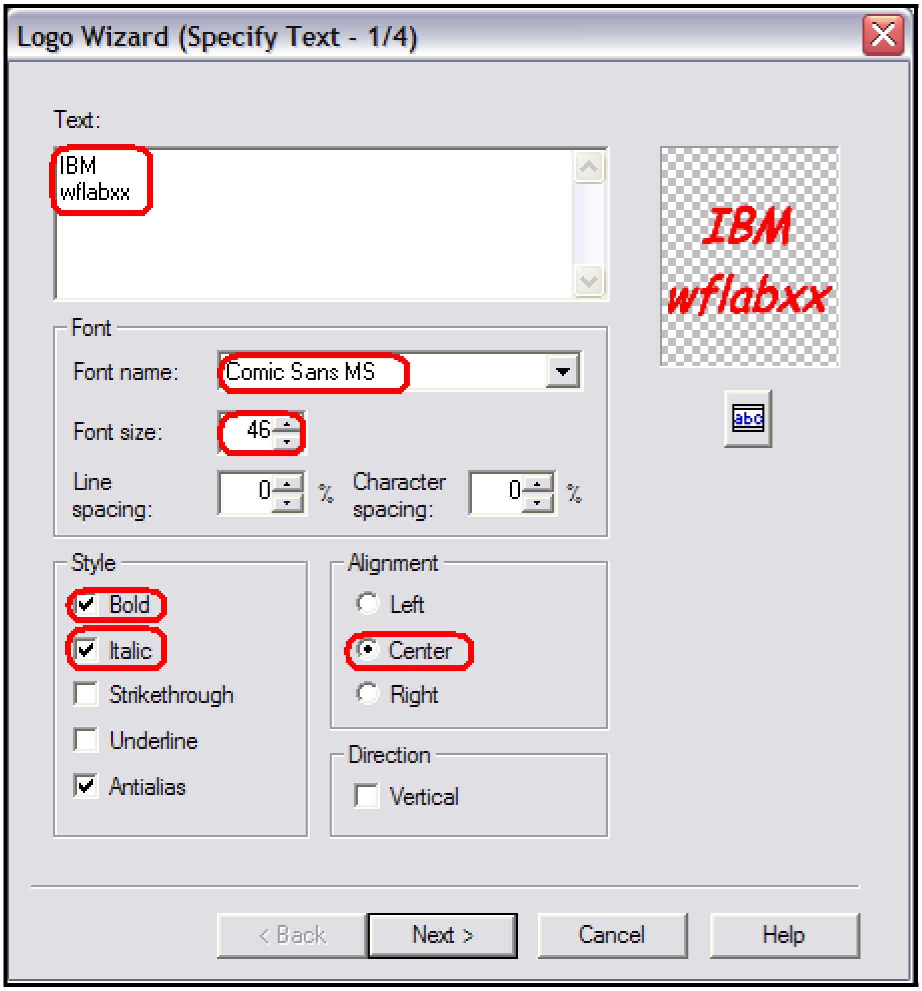


*Figure 146. Click the **Create Logo** button.*

The Logo Wizard opens (see Figure 147).



Text:
IBM
wflabxx

Font
Font name: Comic Sans MS
Font size: 46
Line spacing: 0 %
Character spacing: 0 %

Style
☑ Bold
☑ Italic
□ Strikethrough
□ Underline
☑ Antialias

Alignment
○ Left
● Center
○ Right

Direction
□ Vertical

Figure 147. The Logo Wizard opens.

2. Enter your company name in the **Text** field, and in the next line of the **Text** field enter the project name.
3. Select **Comic Sans MS** in the **Font name** list.
4. Select **46** in the **Font size** list.
5. Under **Style**, select **Bold** and **Italic** check boxes.
6. Leave the **Antialias** check box selected.
7. Click the **Center** radio button under **Alignment**.

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

8. Click **Next** to go to the next page of the wizard.

   The Select Color page of the wizard opens (see Figure 148).

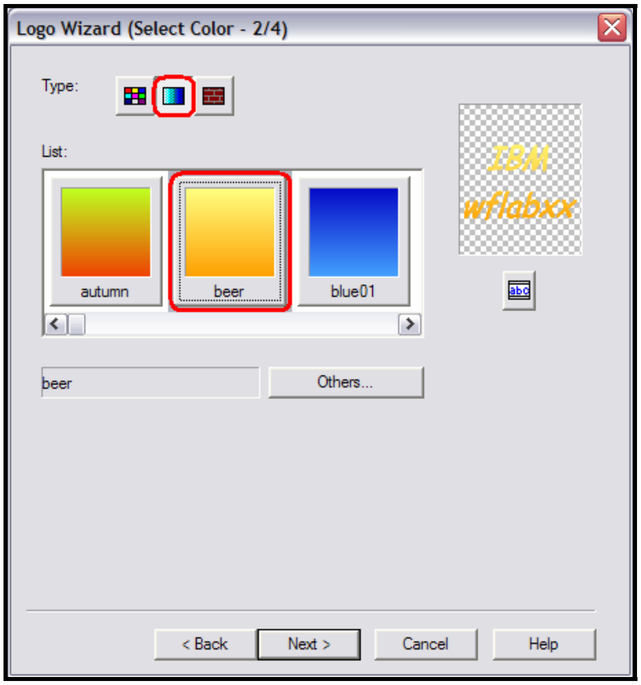


Figure 148. The Select Color page of the wizard

9. Click the gradation **Type** push button, the middle one of the three type push buttons.

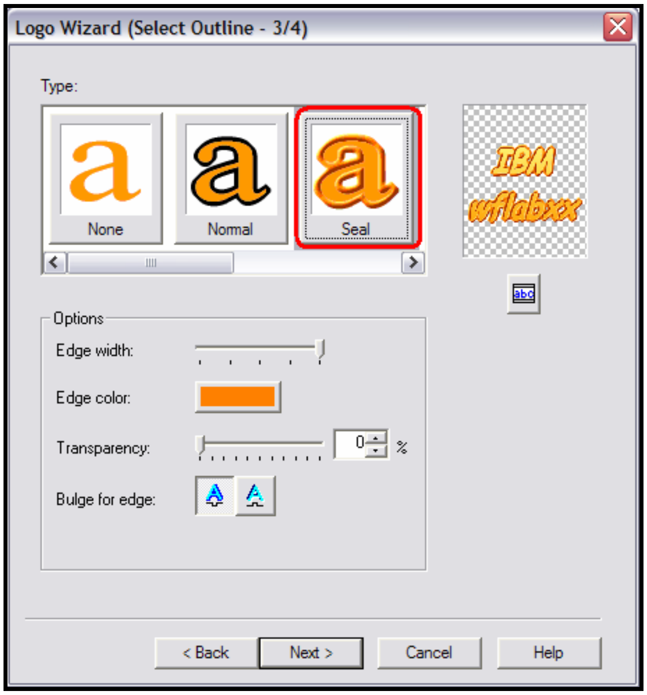
   Note: The other push buttons select color types: solid and textured.

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

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

11. Click **Next** to move forward to the next page.

The Select Outline page opens (see Figure 149).



Logo Wizard (Select Outline - 3/4)

Type:

Options

Edge width:

Edge color:

Transparency: 0 %

Bulge for edge:

Figure 149. The Select Outline page

12. Select the **Seal** outline from the list or any outline you like best.
13. Click **Next**.

The Select Text Effect page opens (see Figure 150).

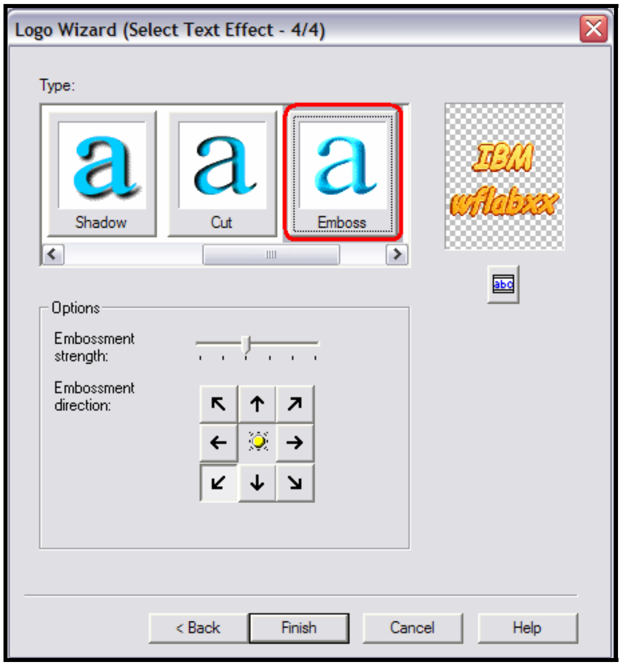


Figure 150. The Select Text Effect page

14. Select the **Emboss text effect** or one that you like best.
15. Click **Finish**.

You return to the WebArt Designer window with the new logo object (see Figure 151).

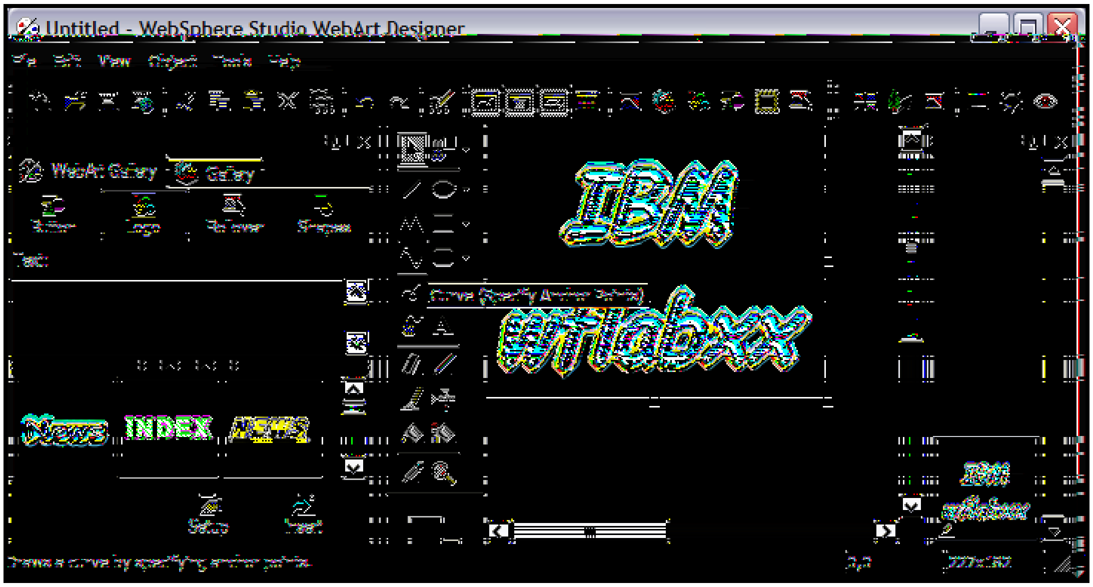

Untitled - WebSphere Studio WebArt Designer
File
Edit
View
Object
Tools
Help
WebArt Gallery
Gallery
Button
Logo
Rollover
Shapes
Text:
News
INDEX
NEWS
Setup
Insert
Curve (Specify Anchor Points)
IBM
wflabxx
Draws a curve by specifying anchor points.

Figure 151. WebArt Designer window with the new logo object

Resizing the logo

To resize the logo object on the canvas:

1. Click the logo object to select it.
2. Move the cursor to the rectangle at the right bottom corners of the object, and watch the cursor changing shape.
3. Drag the rectangle up and to the left so the object becomes smaller (see Figure 152).

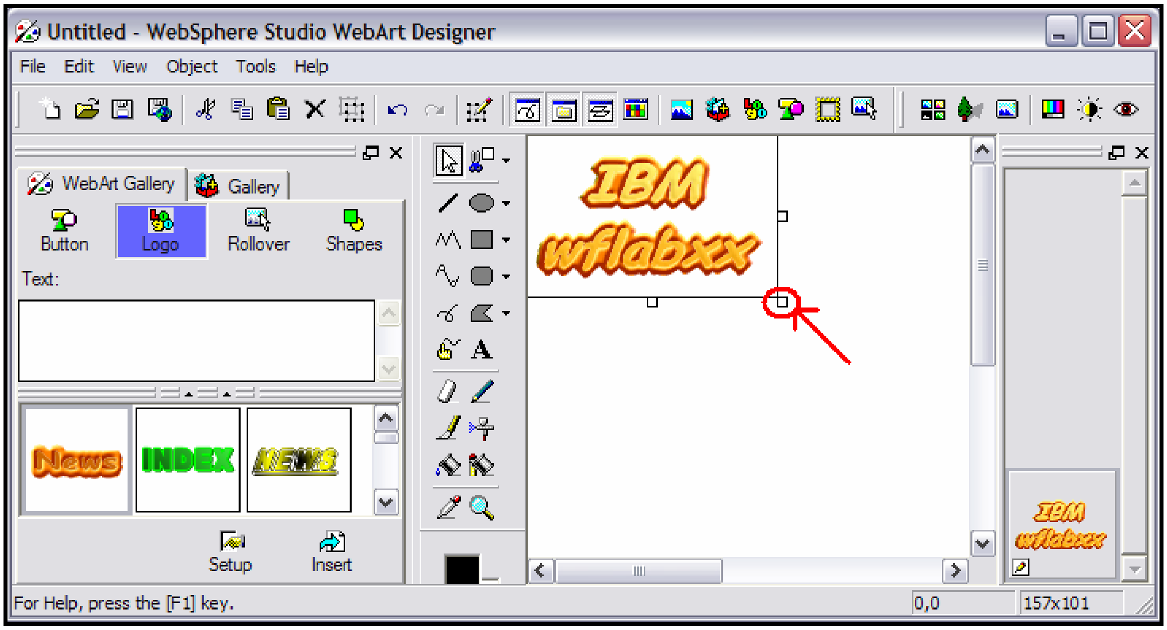


***Figure** 152. Drag the rectangle up and to the left so the object becomes smaller.*

Now you need to save this object. First, you want save it as a WebArt object, which allows you to work with the object later in WebArt Designer. However, you cannot use that format for your Web page, so you must also save it in another format.

Saving the logo as a WebArt object

To save the logo as a WebArt object (see Figure 153):

1. Click **File** on the WebArt Designer menu.

2. Click **Save Canvas As** on the pop-up menu.

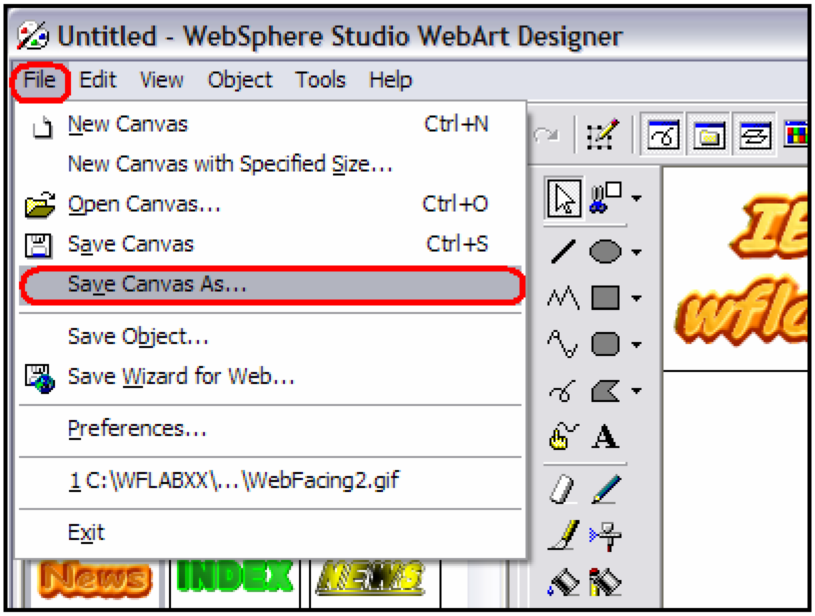


*Figure 153. Click **Save Canvas As**.*

The Save Canvas As dialog opens (see Figure 154).

Note: Make sure you save the canvas in your workspace.

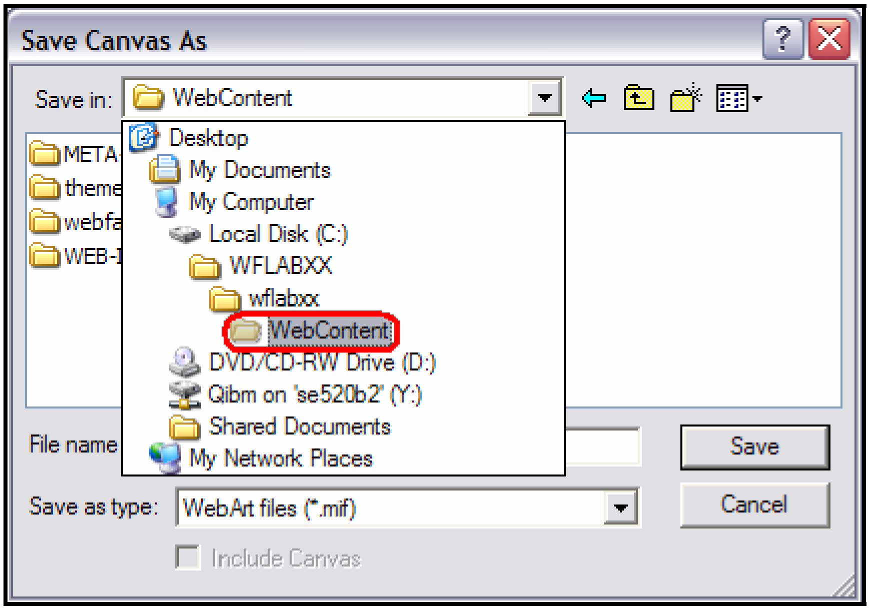
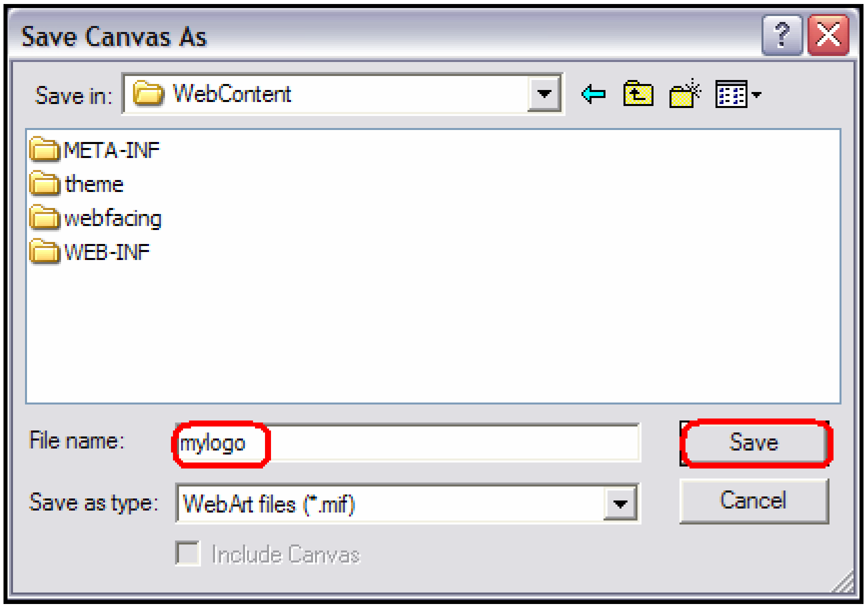


Figure 154. Save the canvas in your workspace.

3. Type `mylogo` in the **File name** field (see Figure 155).



Save Canvas As

Save in: WebContent

META-INF
theme
webfacing
WEB-INF

File name: mylogo

Save as type: WebArt files (*.mif)

□ Include Canvas

*Figure 155. Type mylogo in the **File name** field.*

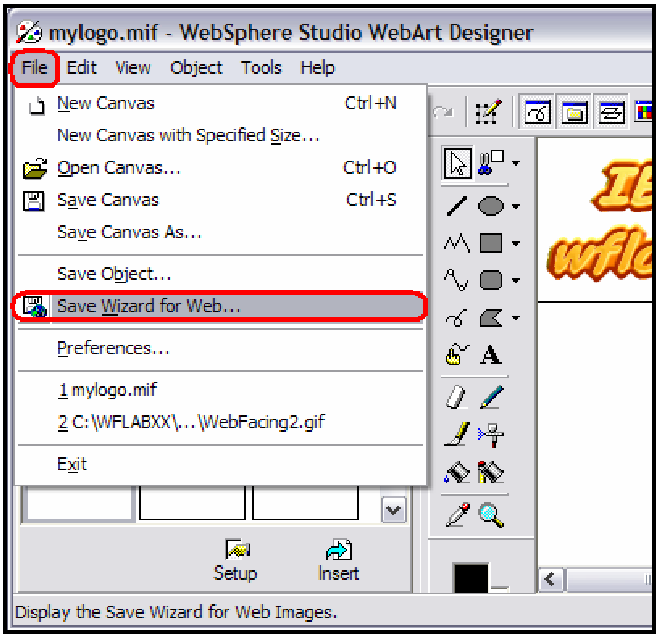
4. Click **Save**.

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

Saving the object for a Web page

To save the logo object for a Web page (see Figure 156):

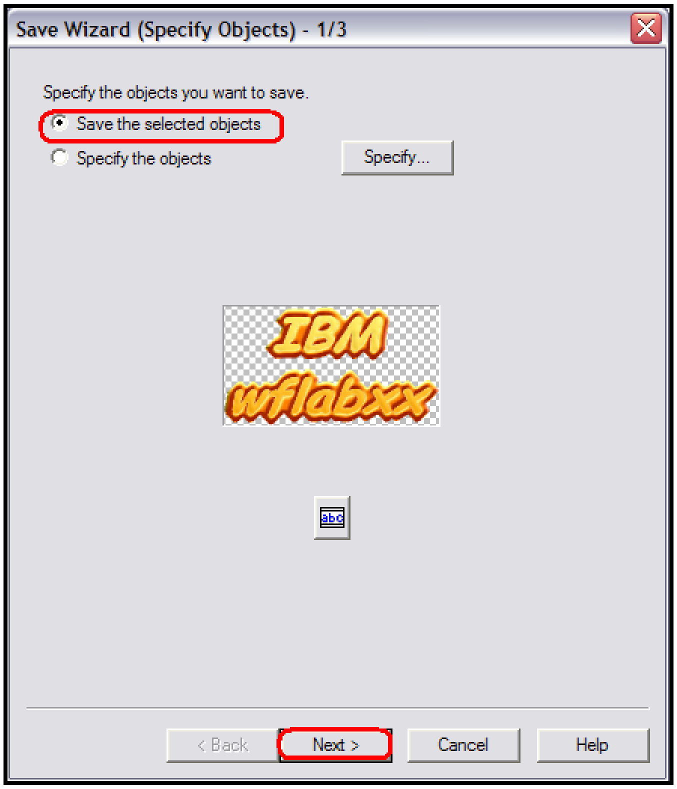
1. Click **File** from the WebArt Designer menu again.
2. Click **Save Wizard for Web** on the pop-up menu.



Display the Save Wizard for Web Images.

*Figure 156. Click **Save Wizard for Web**.*

The Save Wizard dialog opens (see Figure 157).



Save Wizard (Specify Objects) - 1/3

Specify the objects you want to save.

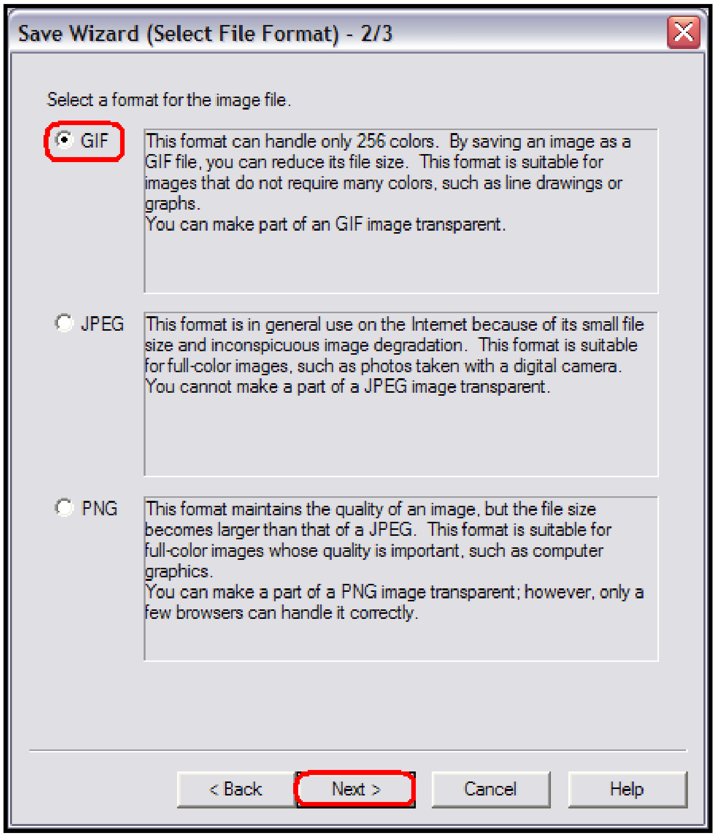
● Save the selected objects

○ Specify the objects

Figure 157. The Save Wizard dialog

3. Leave the **Save the selected objects** radio button selected.
4. Click **Next**.

Now the Select File Format page opens (see Figure 158).



Select a format for the image file.

● GIF

This format can handle only 256 colors. By saving an image as a GIF file, you can reduce its file size. This format is suitable for images that do not require many colors, such as line drawings or graphs.
You can make part of an GIF image transparent.

○ JPEG

This format is in general use on the Internet because of its small file size and inconspicuous image degradation. This format is suitable for full-color images, such as photos taken with a digital camera.
You cannot make a part of a JPEG image transparent.

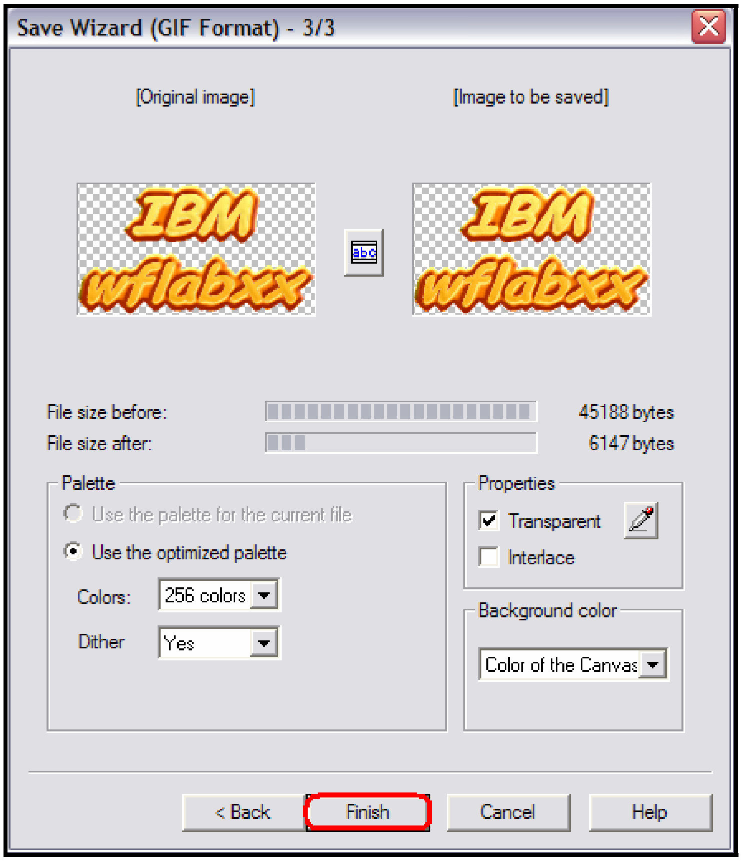
○ PNG

This format maintains the quality of an image, but the file size becomes larger than that of a JPEG. This format is suitable for full-color images whose quality is important, such as computer graphics.
You can make a part of a PNG image transparent; however, only a few browsers can handle it correctly.

Figure 158. The Select File Format page

5. Leave the **GIF** radio button selected.
6. Click **Next**.

The GIF Format page opens (see Figure 159).



Save Wizard (GIF Format) - 3/3

File size before: 45188 bytes

File size after: 6147 bytes

Palette

○ Use the palette for the current file

● Use the optimized palette

Colors: 256 colors

Dither: Yes

Properties

☑ Transparent

□ Interlace

Background color: Color of the Canvas

Figure 159. The GIF Format page

7. Click **Finish**.

The Save As dialog opens (see Figure 160).

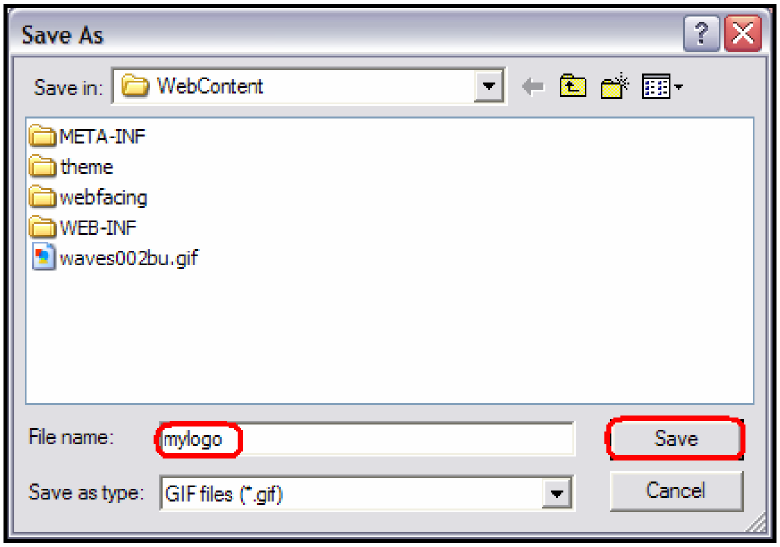


Figure 160. The Save As dialog

8. Type `mylogo` in the **File name** field.
9. Click **Save**.

Next, close the WebArt Designer.

10. Click **File** from the menu.

11. Click **Exit** on the pop-up menu (see Figure 161).

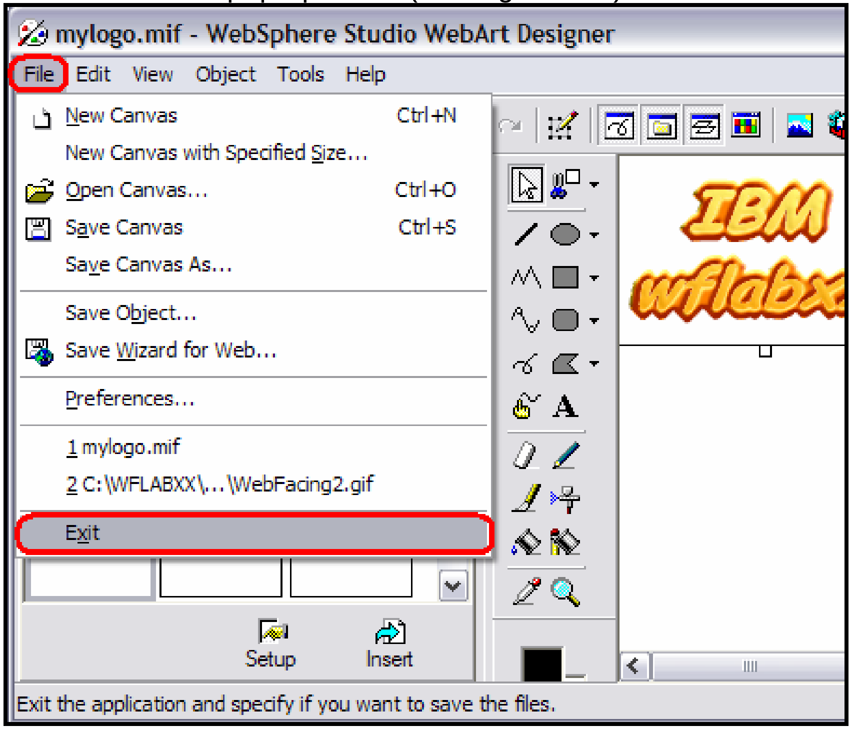


*Figure 161. To get back to the workbench, click **Exit***

You are back in the workbench in the WebFacing perspective showing the WebFacing Projects view.

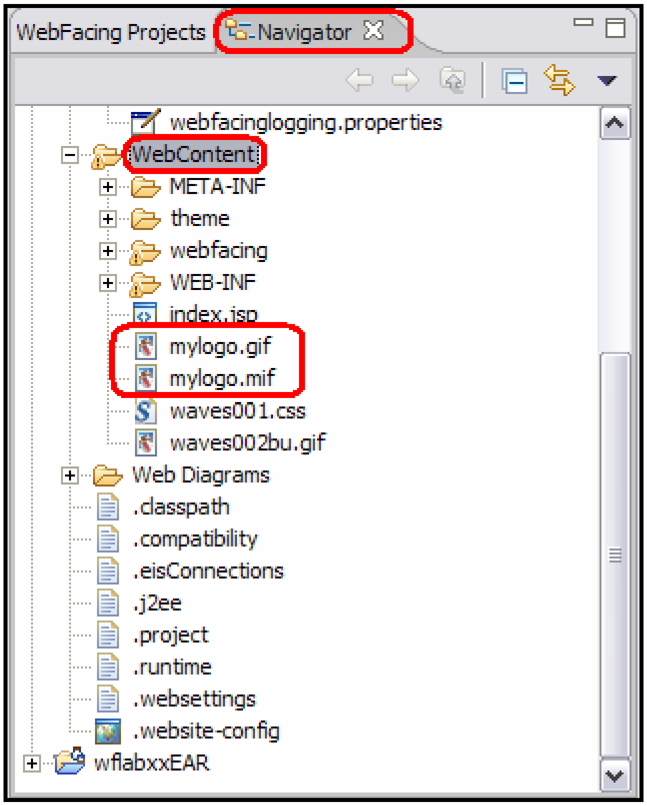
Next, switch to the Navigator view to locate the logo object mylogo.gif and to place the logo on the Design page.

Note: You might have to click **Window > Show view > Navigator** from the workbench menu.

Placing the logo on the Design page

To place the logo on the Design page (see Figure 162):

1. Click the **Navigator** tab.
2. Expand the wflabxx Web project if not already expanded.
3. Expand the **WebContent** folder if not already expanded.



```
webfacinglogging.properties
WebContent
  META-INF
  theme
  webfacing
  WEB-INF
  index.jsp
  mylogo.gif
  mylogo.mif
  waves001.css
  waves002bu.gif
Web Diagrams
.classpath
.compatibility
.eisConnections
.j2ee
.project
.runtime
.websettings
.website-config
wflabxxEAR
```

Figure 162. Placing the logo on the Design page

The mylogo.gif file appears in the list. If it does not appear, refresh the list (see Figure 163).

4. Right-click the **WebContent** folder icon.
5. Click **Refresh** on the pop-up menu.

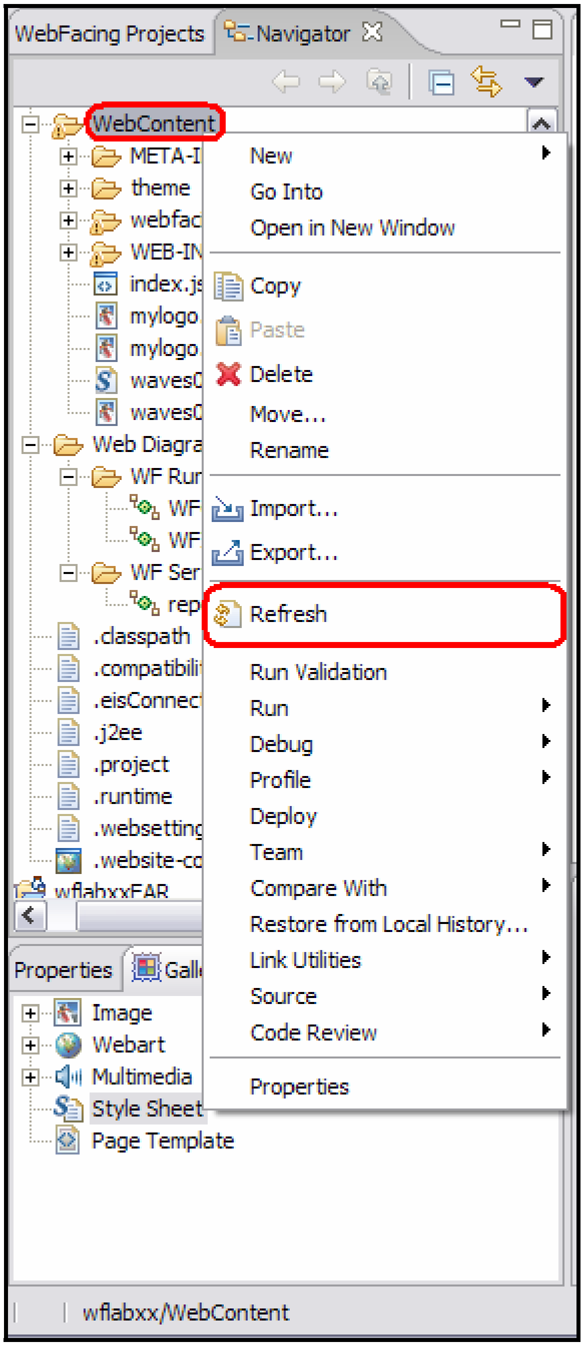


*Figure 163. Refresh the list if **mylogo.gif** file does not appear*

Hopefully, you can see the file now in the Web Content folder, if not, go to Windows Explorer and search for the mylogo.gif on your hard drive. Move it to the Web Content folder in the WebSphere Development Studio Client workspace. If you did not use the default workspace location, do a search for the workspace and the wflabxx directory in it. Move the mylogo.gif into the WebContent directory under the wflabxx directory. Refresh the Navigator view again.

Now, you can take the logo and put it on your Web page that is still open in Page Designer. If Page Designer has been closed, just open the index.jsp file. Make sure you are on the Design page, not the Preview page.

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

The logo is placed on the Design page (see Figure 164).

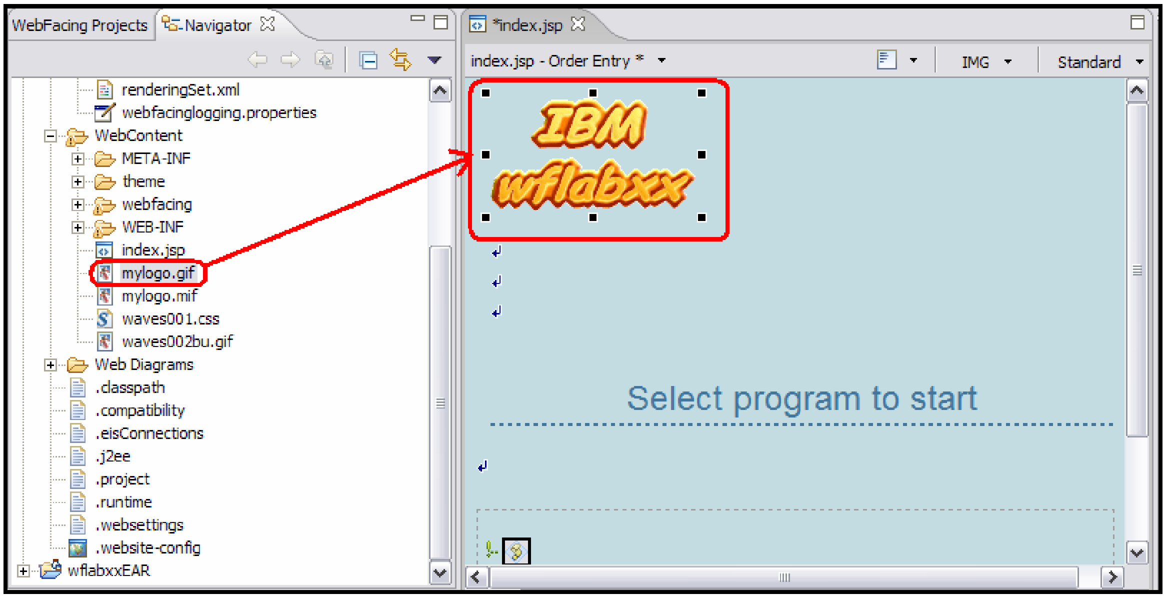


Figure 164. The logo is placed on the Design page.

Exercise 10.5: Adding a Heading 1 tag to the page

Now, you want to insert a heading below the logo.

To add a Heading 1 tag:

1. Position the cursor just below the Logo at the first BR tag (see Figure 165).

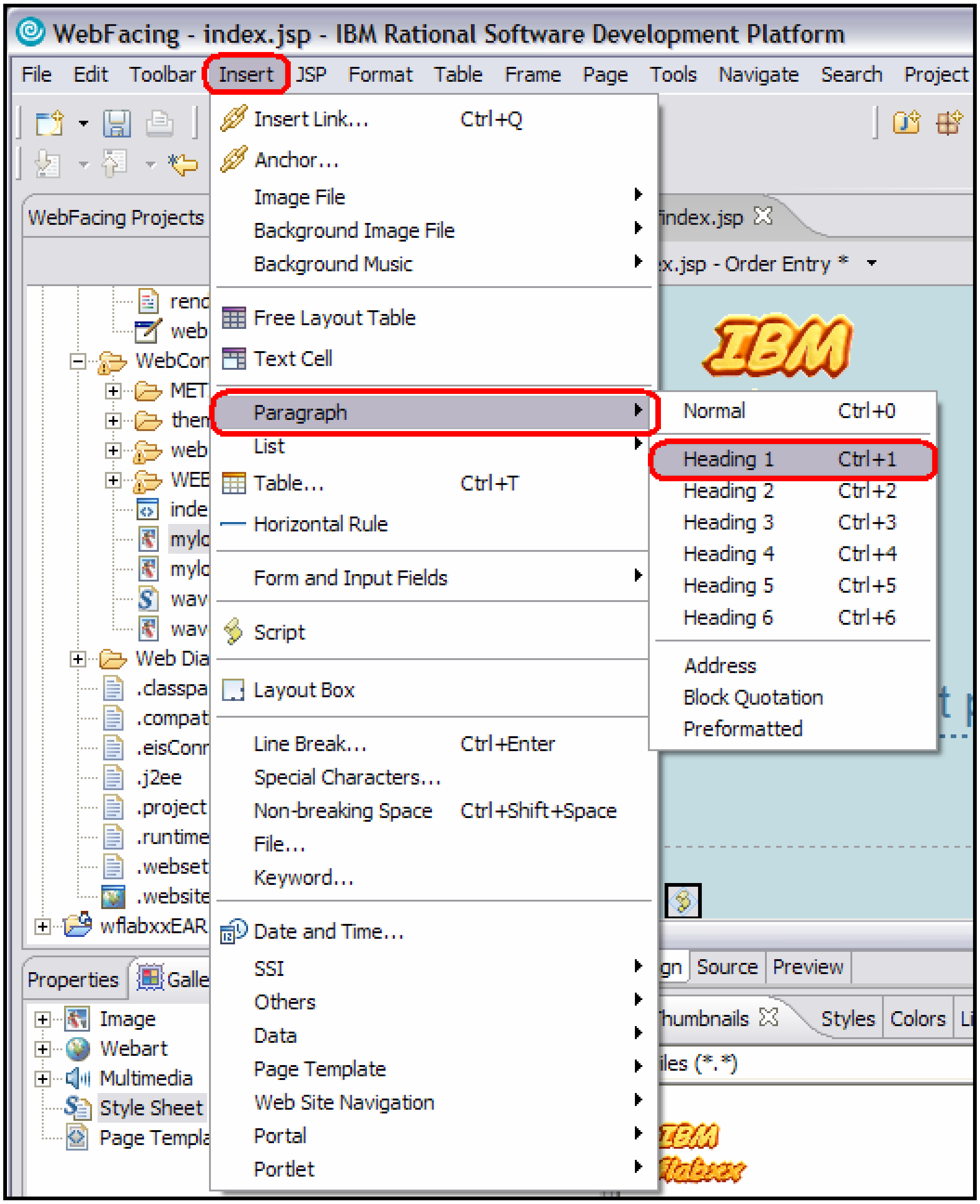


Figure 165. Add a ***Heading 1*** *tag*

2. Click **Insert** from the workbench menu.

3. Click **Paragraph > Heading 1** on the pop-up menu.

   A frame appears that allows you to enter text.

4. Enter your company name followed by **Order Entry Application** (see Figure 166).

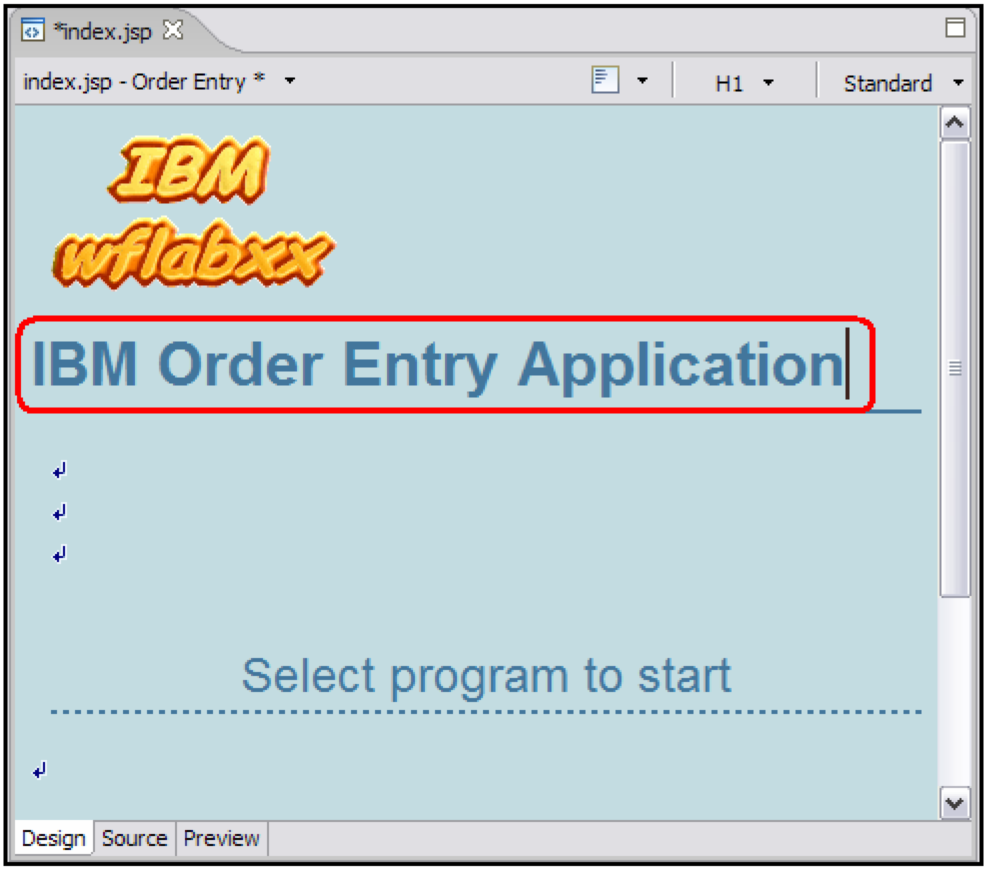


Figure 166. Enter your company name Order Entry Application.

Now, you want to use one of the sample pictures included with WebSphere Development Studio Client and place this picture on the page.

Exercise 10.6: Adding a picture to the page

To add a picture to the page, follow the steps below (see Figure 167):

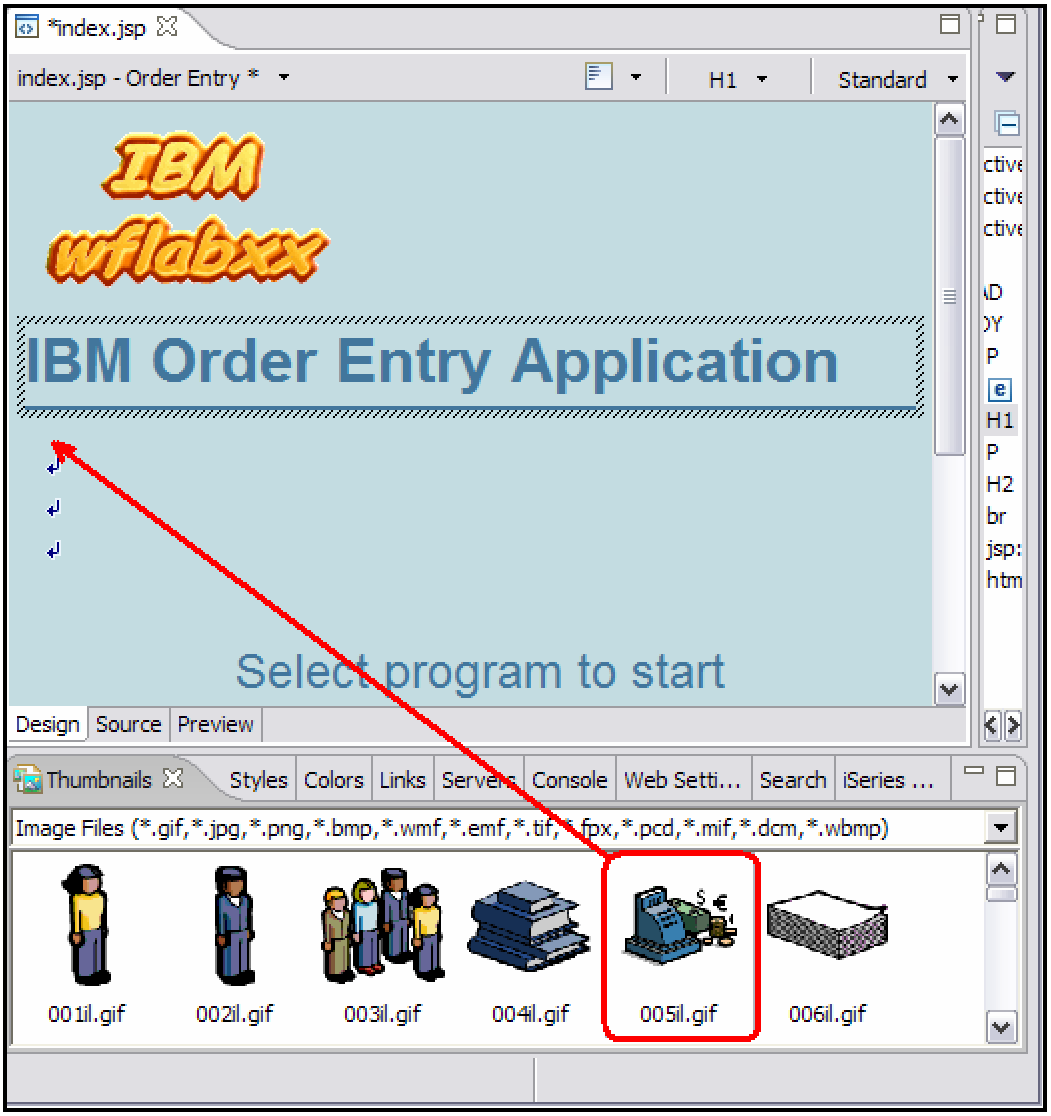


Figure 167. Adding a picture to the page

1. In the workbench in the Navigator view, select the **Gallery** tab.
2. Expand the **Image** folder.
3. Select the **Illustration** folder.
4. In the Thumbnails view beside the Gallery view, select one of the sample illustrations, for example, use file 005il.gif.

   Note: The sample illustrations change from release to release so 005il.gif might not be available in your environment.

5. Drag a picture onto the Design page below the heading.

The Design page now contains a picture (see Figure 168).

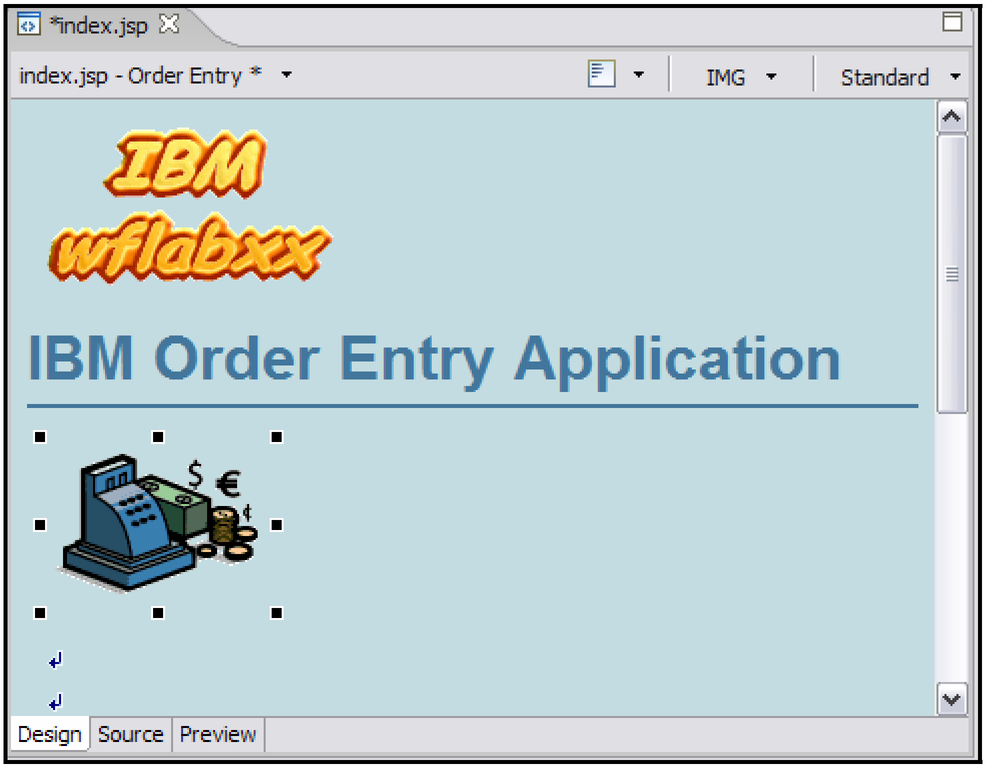


Figure 168. The Design page contains a picture.

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

Exercise 10.7: Adding moving text to the page

Now you add moving text to the page.

To add moving text (see Figure 169):

1. Position the cursor in the Design page underneath the picture.
2. Click **Insert** from the workbench menu.
3. Click **Paragraph > Heading 3** on the pop-up menu.

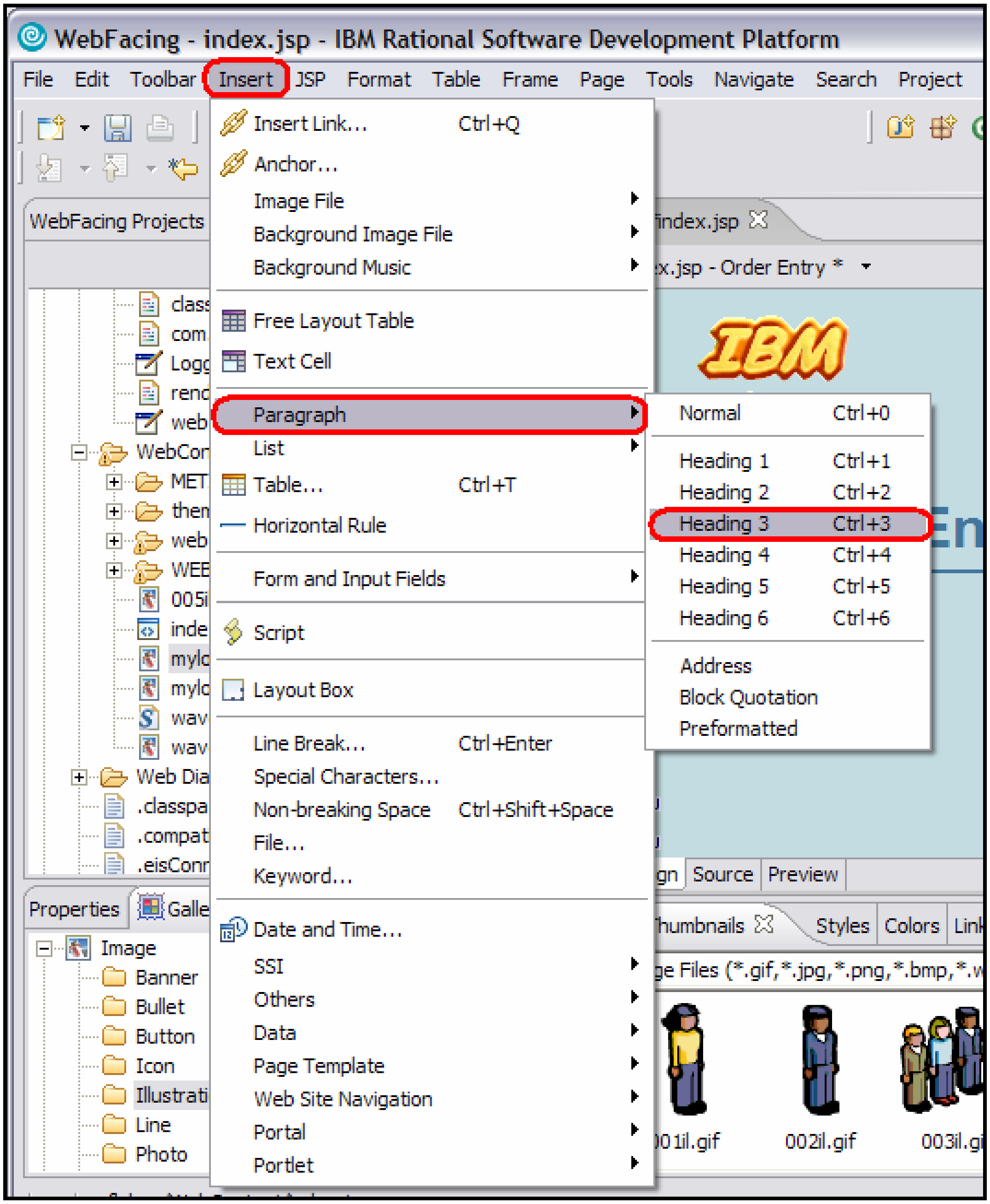


Figure 169. Adding moving text

4. Leave the cursor positioned inside the heading 3 frame (see Figure 170):

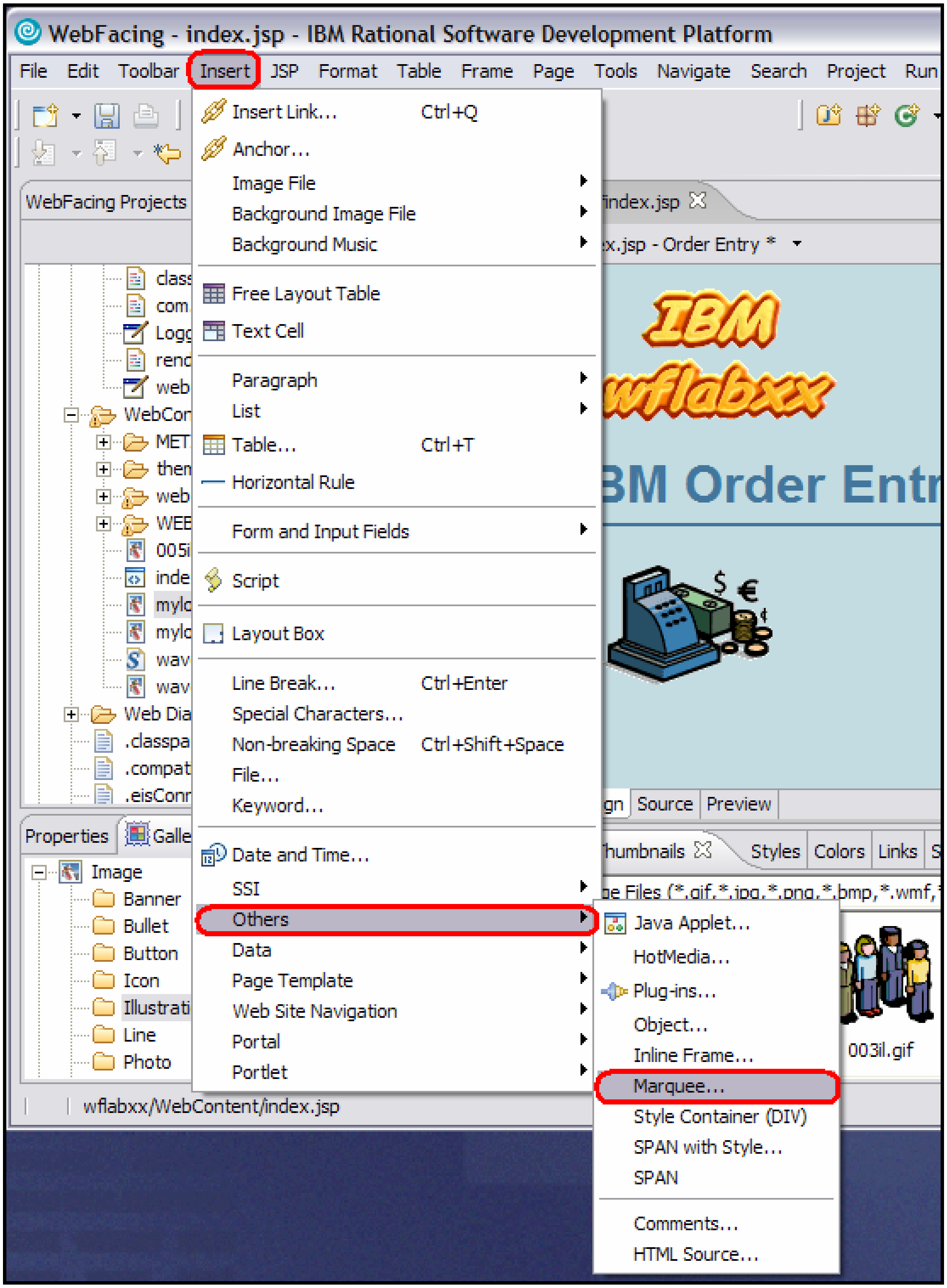
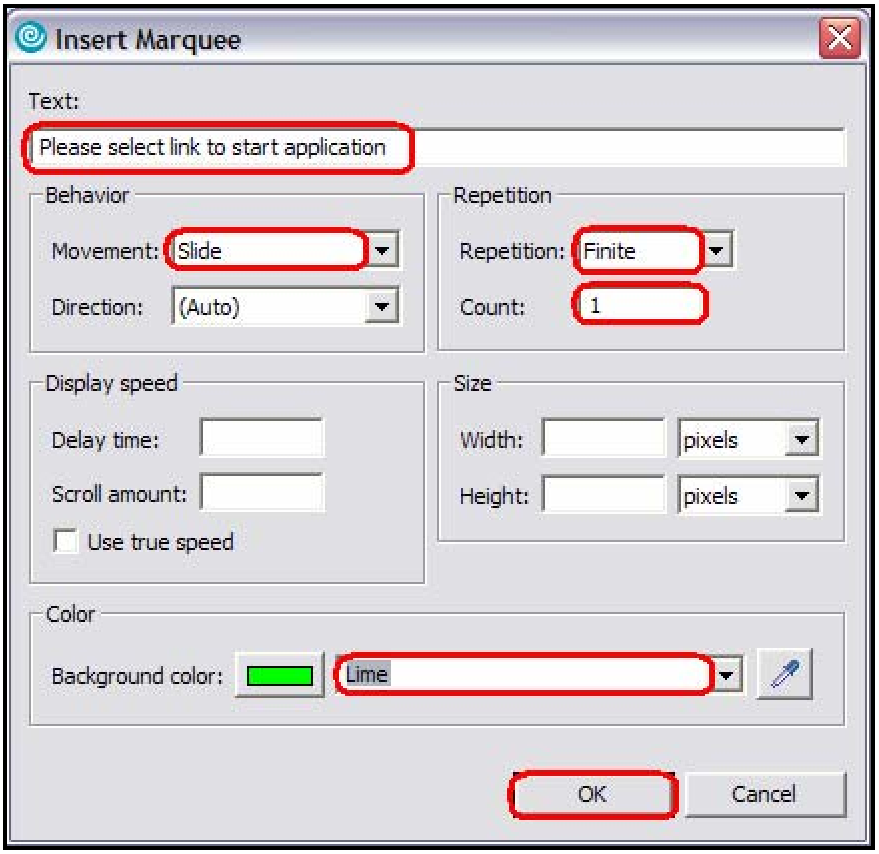


*Figure 170. Leave the cursor positioned inside the **Heading 3** frame.*

5. Click **Insert** from the workbench menu.

6. Click **Others > Marquee** on the pop-up menu.

   The Insert Marquee dialog opens (see Figure 171).



Insert Marquee

Text: Please select link to start application

Behavior
Movement: Slide
Direction: (Auto)

Repetition
Repetition: Finite
Count: 1

Display speed
Delay time:
Scroll amount:
□ Use true speed

Size
Width: pixels
Height: pixels

Color
Background color: 🟩 Lime

Figure 171. The Insert Marquee dialog

7. Into the **Text** field, type `Please select link to start application`.
8. Under **Behavior**, select **Slide** in the **Movement** list.
9. Under **Repetition**, select **Finite** in the **Repetition** list.
10. Under **Repetition**, type `1` in the **Count** field.

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

11. Select **Lime** in the **Background color** list. Click **OK** on the color palette.
12. Click **OK** on the Insert Marquee dialog.

The Design page look similar to Figure 172.

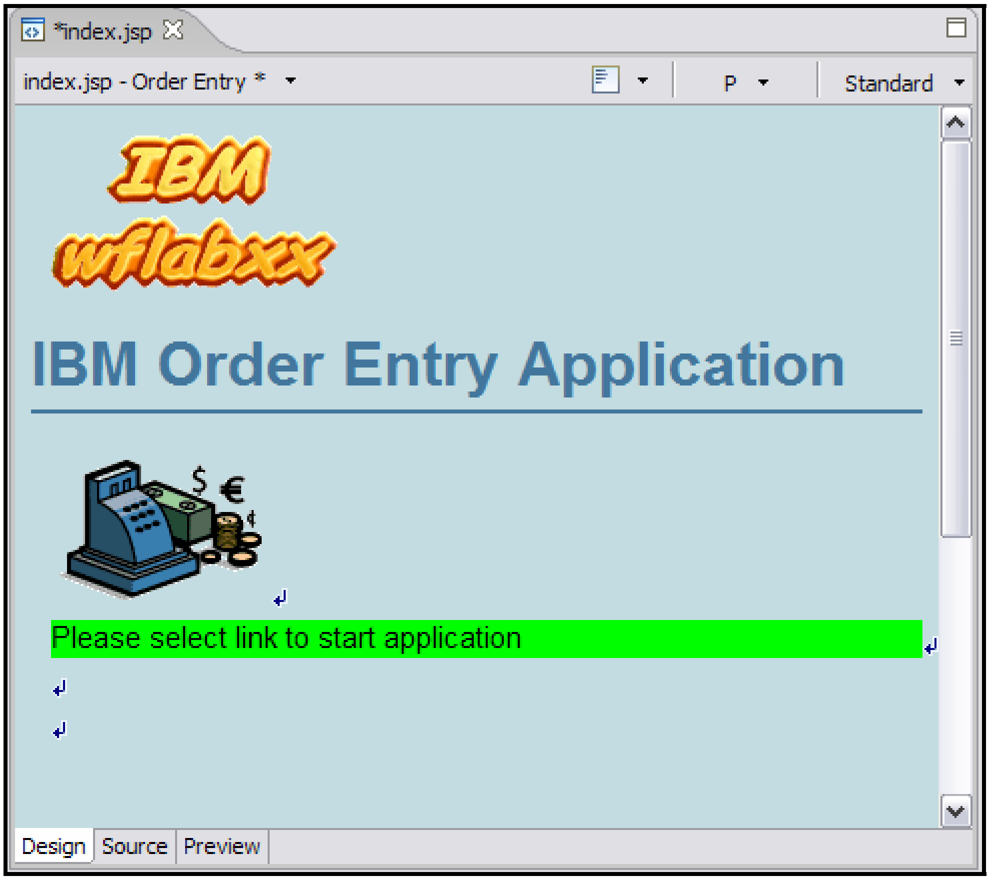
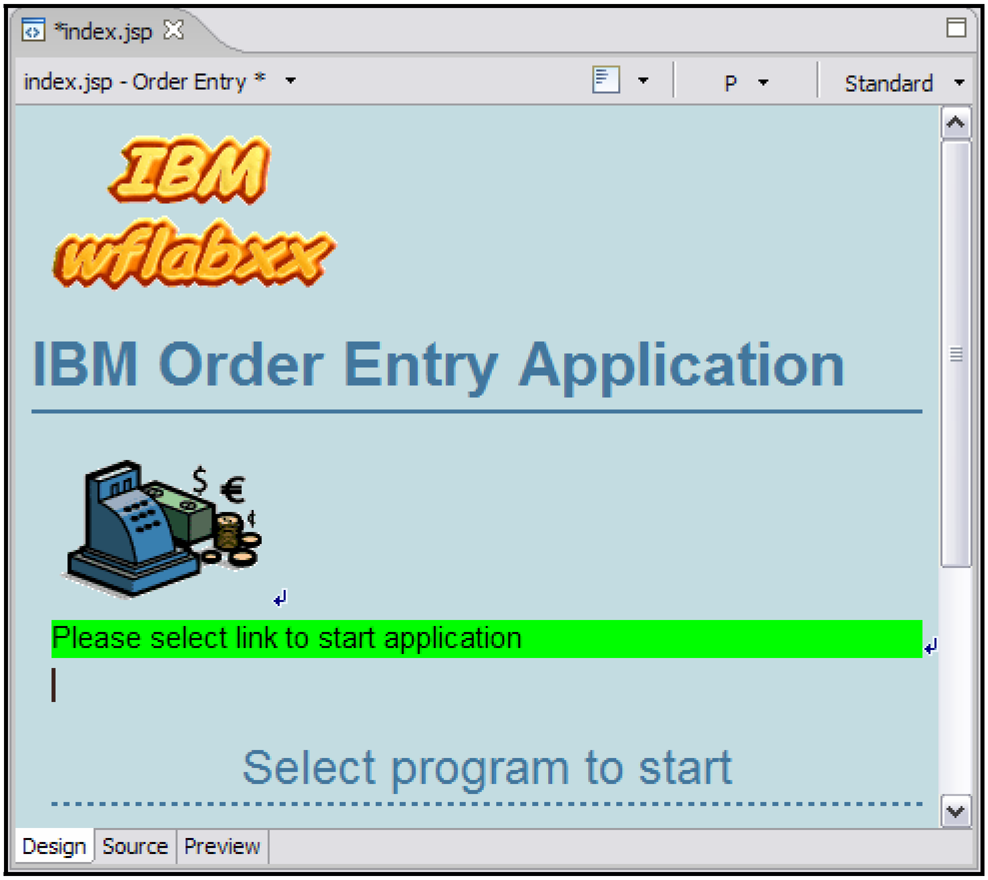

*index.jsp
index.jsp - Order Entry *
P
Standard
IBM
wflabxx
IBM Order Entry Application
Please select link to start application
Design
Source
Preview

Figure 172. The Design page

Next, you save some space by removing some of the line break tags.

13. Position the cursor on the BR tag.
14. Press the **Delete** key until the “Select program to start” title appears on your page (see Figure 173).



IBM Order Entry Application

Please select link to start application

Select program to start

Figure 173. Remove some of the line break tags.

Next, you look at the page, as it appears in a browser.

15. Click the **Preview** tab at the bottom of the Design page (see Figure 174).

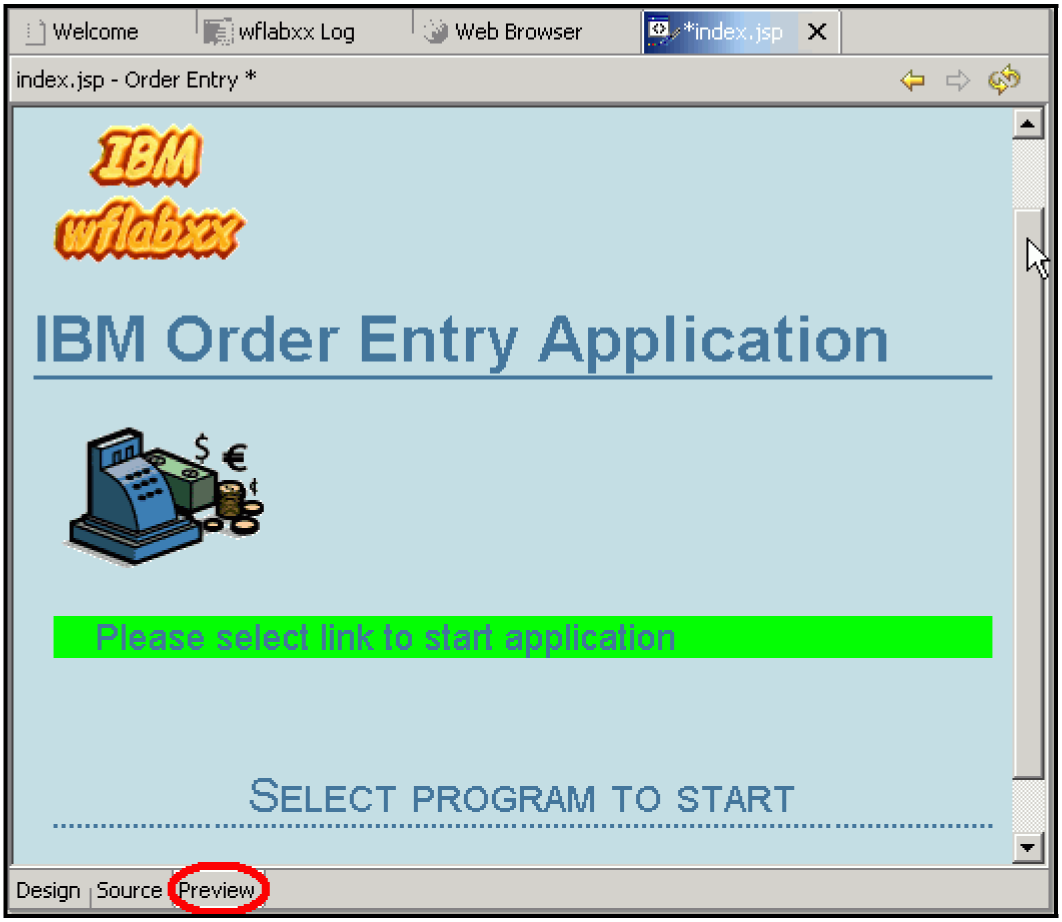
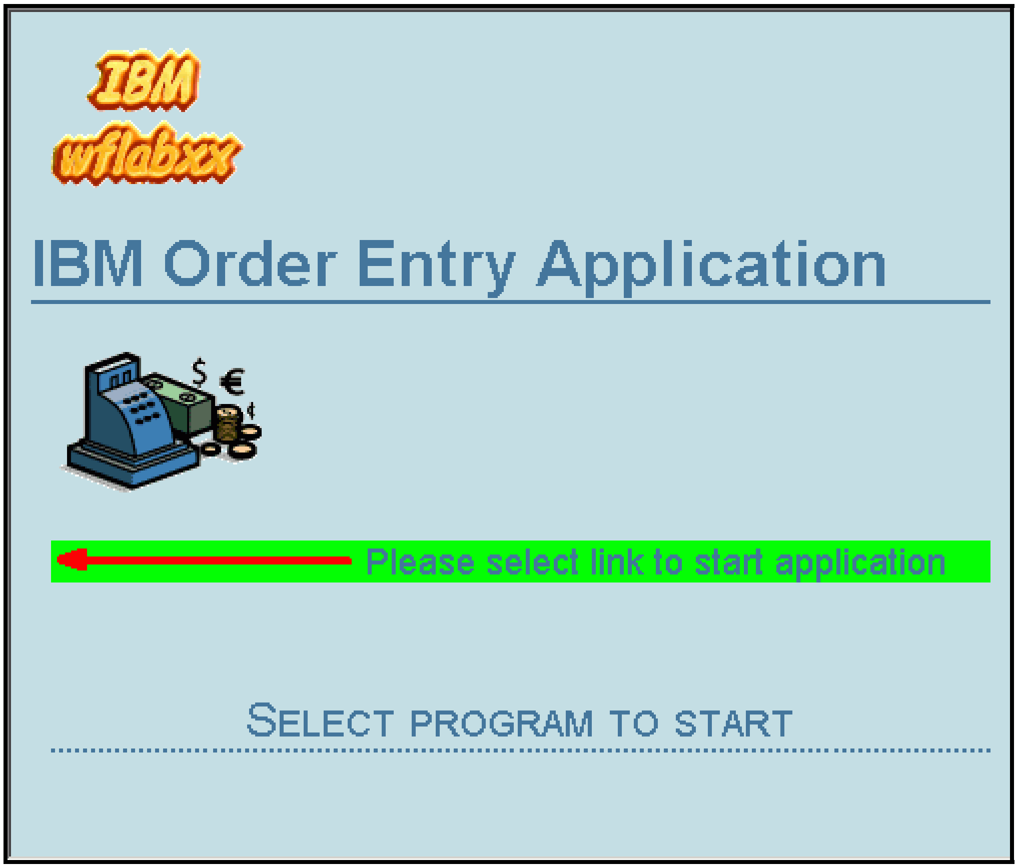


Figure 174. The SELECT PROGRAM TO START title appears.

Notice that your heading three text is sliding in (see Figure 175).



IBM
wflabxx
IBM Order Entry Application
Please select link to start application
SELECT PROGRAM TO START

Figure 175. Heading 3 text slides in

Exercise 10.8: Changing the text color

You notice that the Text color and the background are not easy to distinguish from each other. One way of changing this, is to apply another color to a certain area of the text. To do that, you have to return to the Design page.

To change the text color:

1. Click the **Design** tab.

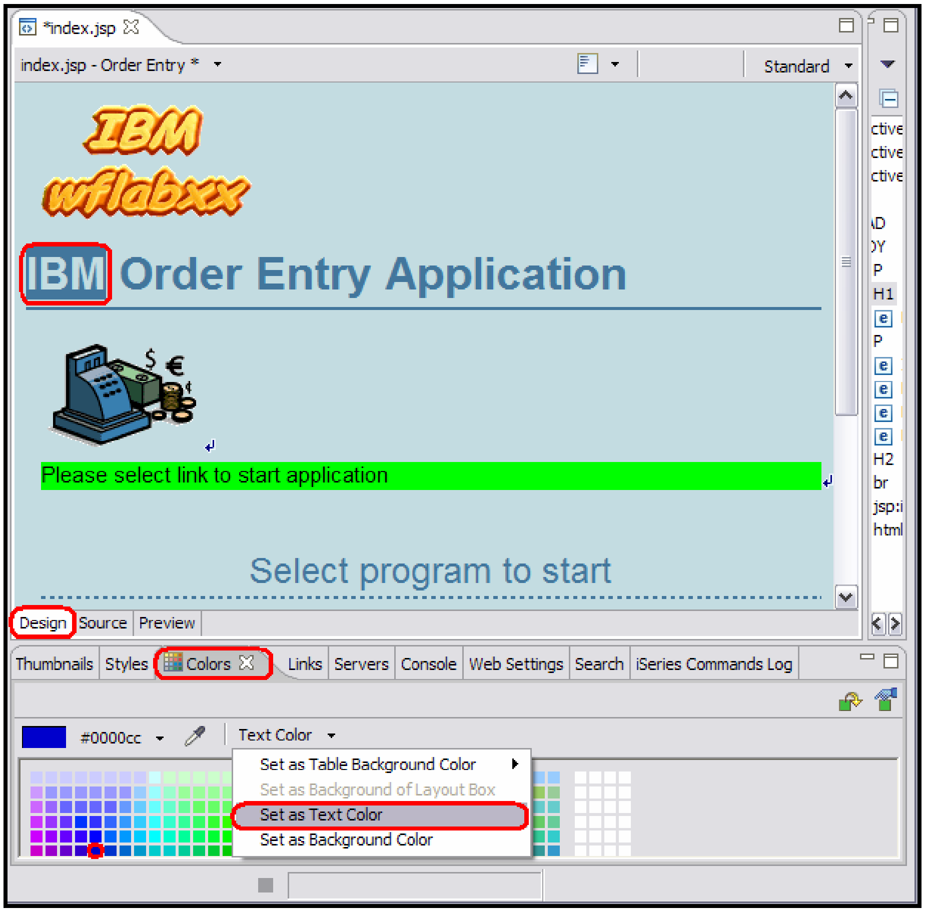


*Figure 176. Click the **Design** tab*

2. Select the company text (IBM) for which you want to change color by selecting the text area with the mouse cursor. After you have selected a text area, the Properties view opens.
3. Under the Design page, select the **Colors** tab.
4. In the Colors view, select a color with your mouse cursor.
5. Click the down arrow next to the **Text Color** from the Colors view.
6. Click **Set as Text Color** on the pop-up menu.

   You are done; now the selected text is displayed with the color you selected for it.

Exercise 10.9: Deleting default text from the page

Now, you want to remove the default text that was added by the IBM WebFacing Tool.

To delete default text from the page, follow the steps below (see Figure 177):

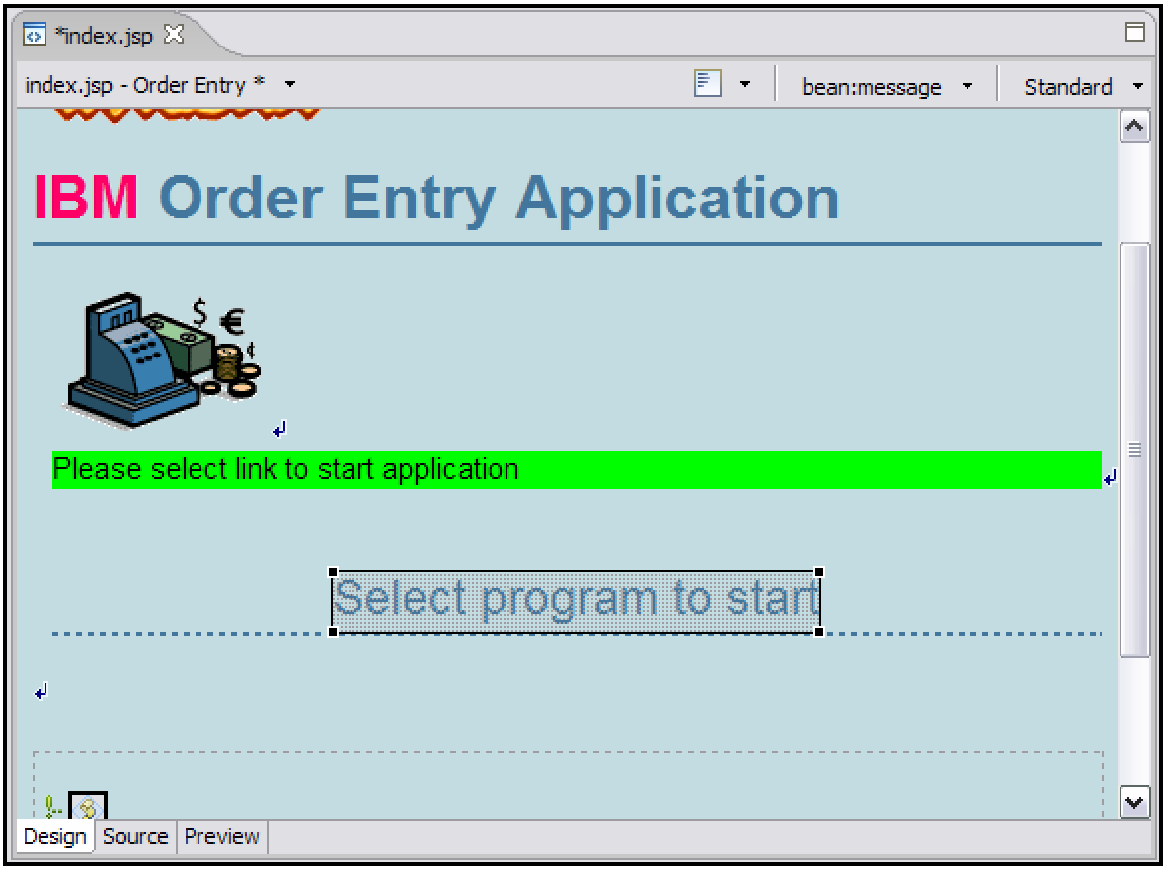

*index.jsp
index.jsp - Order Entry *
bean:message
Standard
IBM Order Entry Application
Please select link to start application
Select program to start
Design
Source
Preview

Figure 177. Deleting default text from page

1. Select the text **Select program to start**.
2. Press the **Delete** key.

3. Select the **Preview** tab to view your completed page (see Figure *178*).

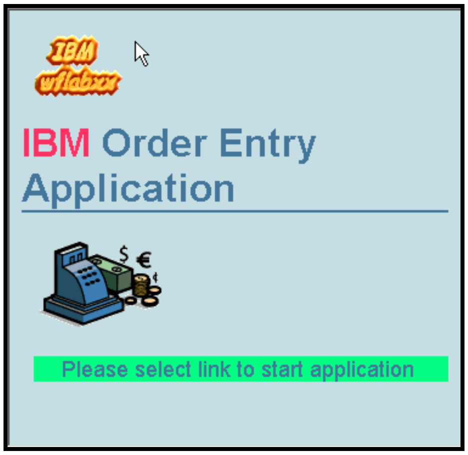

IBM
wflabxx
IBM Order Entry Application
$ €
Please select link to start application

Figure 178. View your completed page.

4. Close the file index.jsp.
5. Click **Yes** to save the changes.

   Page Designer is closed.

6. In the Navigator view or WebFacing Projects view, right-click **wflabxx**.
7. Click **Run>Run on Server** on the pop-up menu.

Your newly designed index.jsp page opens to run your application (see Figure 179).

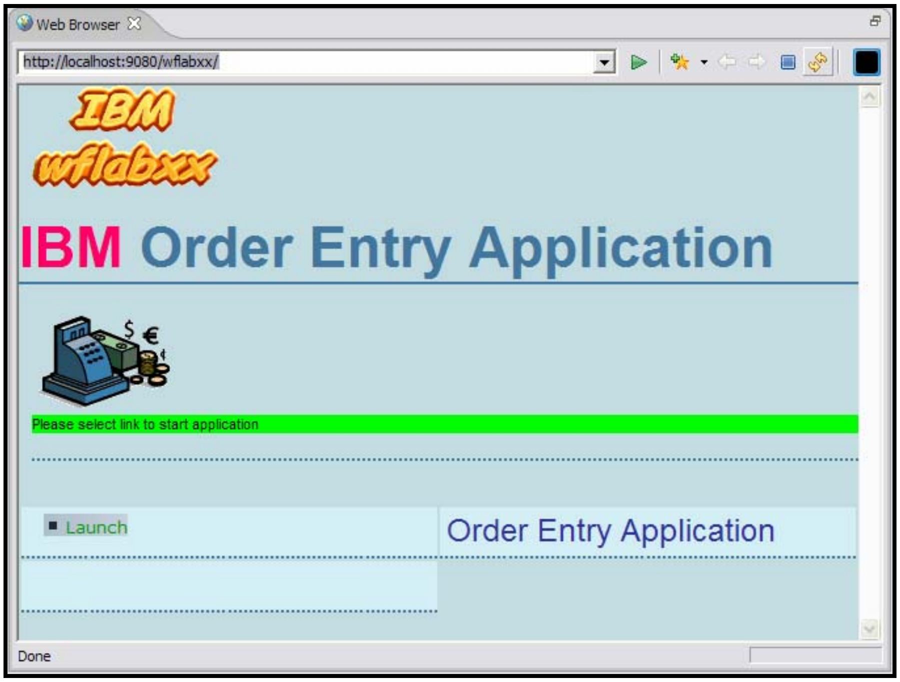

Web Browser
http://localhost:9080/wflabxx/
IBM
wflabxx
IBM Order Entry Application
Please select link to start application
Launch
Order Entry Application
Done

Figure 179. The ***index.jsp*** *page*

Recap

You have completed "Enhancing index.jsp page." You now have the information to understand how to:

- Open Page Designer
- Work with page properties
- Link to a cascading style sheet
- Create a heading
- Add an image
- Create a logo and add it
- Add a marquee
- Change the color of text
- Delete default text

Export/Import IBM WebFacing Tool projects in WebSphere Development Studio Client

There are several ways to export and import IBM WebFaced Applications including .WAR (Web Archive), .EAR (Enterprise Archive), and .Zip file formats. The .WAR and .EAR files are J2EE specifications and can be used to deploy WebFaced Applications to servers. These file formats can also be imported into WebSphere Development Studio Client and be viewed as WebProjects from a Web perspective. However, imported .WAR and .EAR file projects are not visible in the WebFacing Perspective so it is not possible to change to these types of imported projects using the IBM WebFacing Tool.

Customers who wish to be able to export or import IBM WebFacing Tool projects and still have them visible in the WebFacing Perspective needs to use the .ZIP file format. This example focuses on how to export an IBM WebFacing Tool project such that it can be imported into WebSphere Development Studio Client and still be worked from within the WebFacing Environment.

To accomplish these learning objectives, several steps are involved, including:

- Export your IBM WebFacing Tool project from WebSphere Development Studio Client
- Clean up WebSphere Development Studio Client
- Import your IBM WebFacing Tool project into WebSphere Development Studio Client

Length of time

This chapter takes approximately 10 minutes to complete.

Exercise 11.1: Export the IBM WebFacing Tool project

In this step, you are exporting your IBM WebFacing Tool project from WebSphere Development Studio Client. It is important that you follow the instructions correctly because the file you create is used to recreate your project later in this section.

Export your IBM WebFacing Tool project

Follow these steps to export your WebFacing Tool project:

1. From the WebFacing Perspective, select **File** and **Export** (see Figure 180).

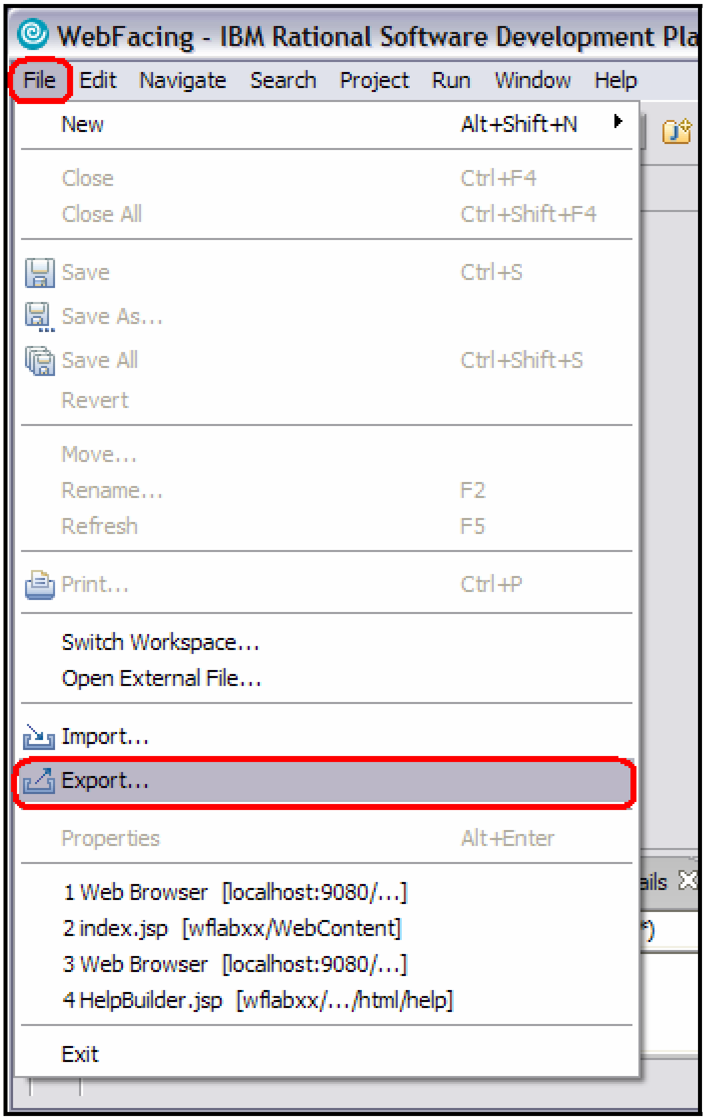


Figure 180. Export your IBM WebFacing Tool project

2. This displays a dialog box. Select **Zip File** in the Dialog Box, and then click **Next** (see Figure 181).

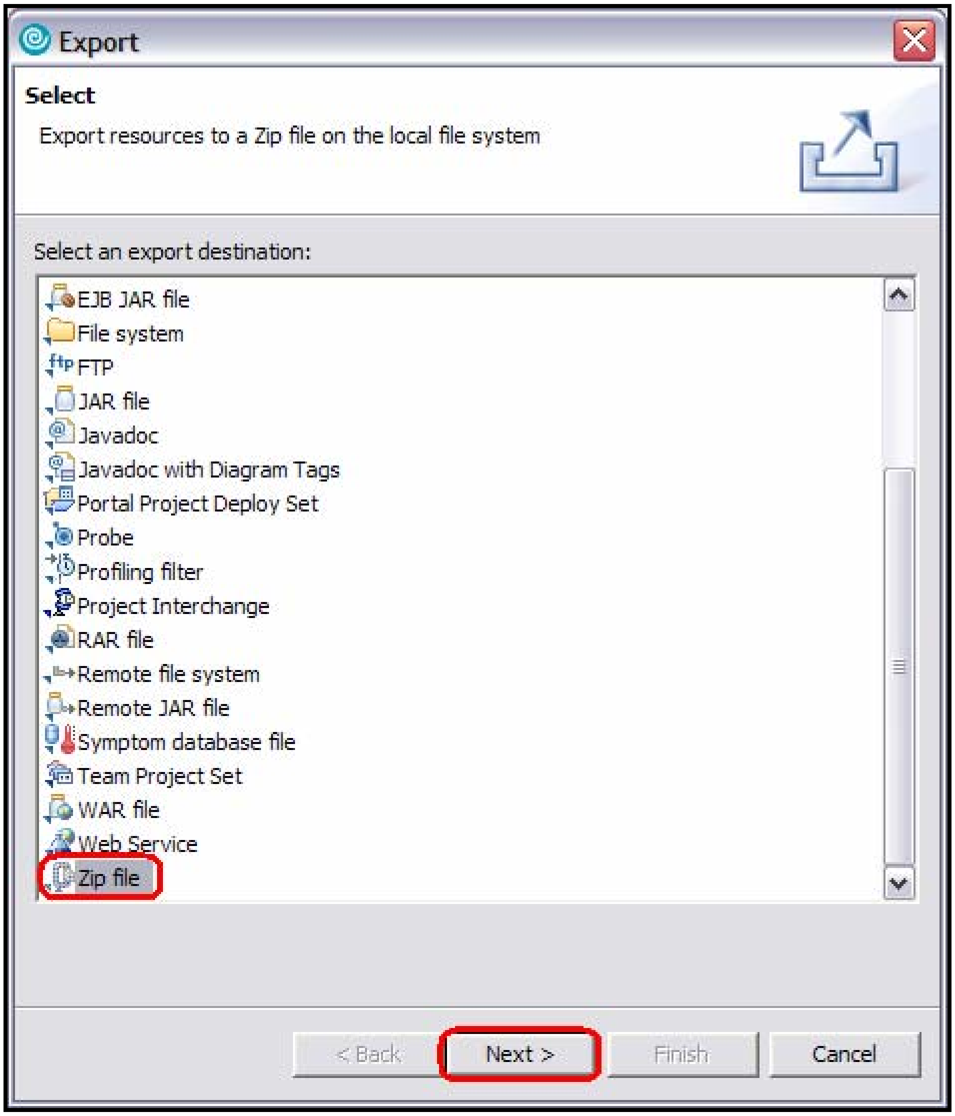
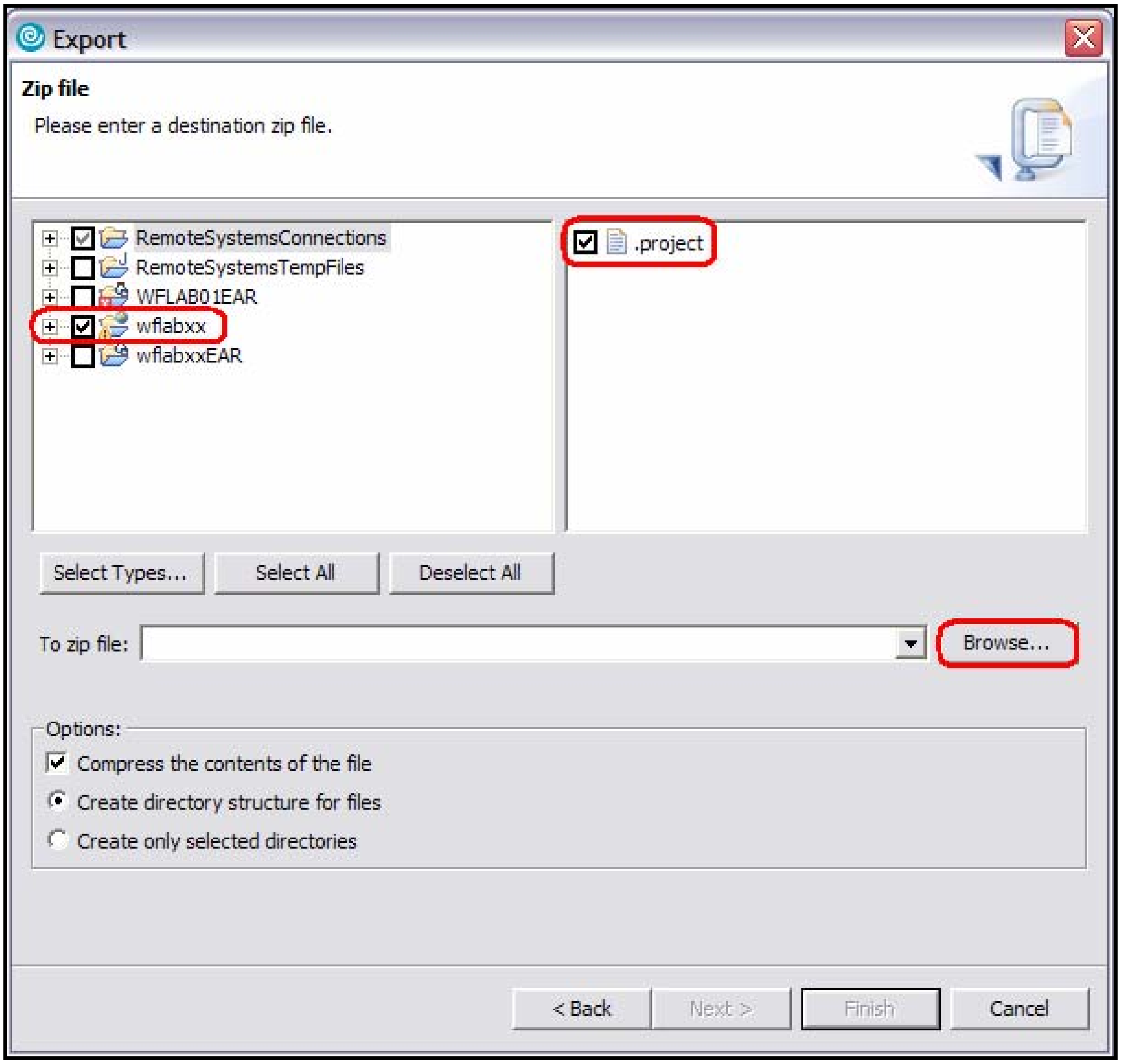


*Figure 181. Select **Zip file***

3. On the next dialog box that is displayed, check the box next to the **wflabxx** library. Also, check the check box next to **.project**. Then click **Browse** next to the **To zip file:** input field (see
Figure ***182***).



Export

Zip file

Please enter a destination zip file.

- ☑ RemoteSystemsConnections
- □ RemoteSystemsTempFiles
- □ WFLAB01EAR
- ☑ wflabxx
- □ wflabxxEAR

☑ .project

To zip file:

Options:

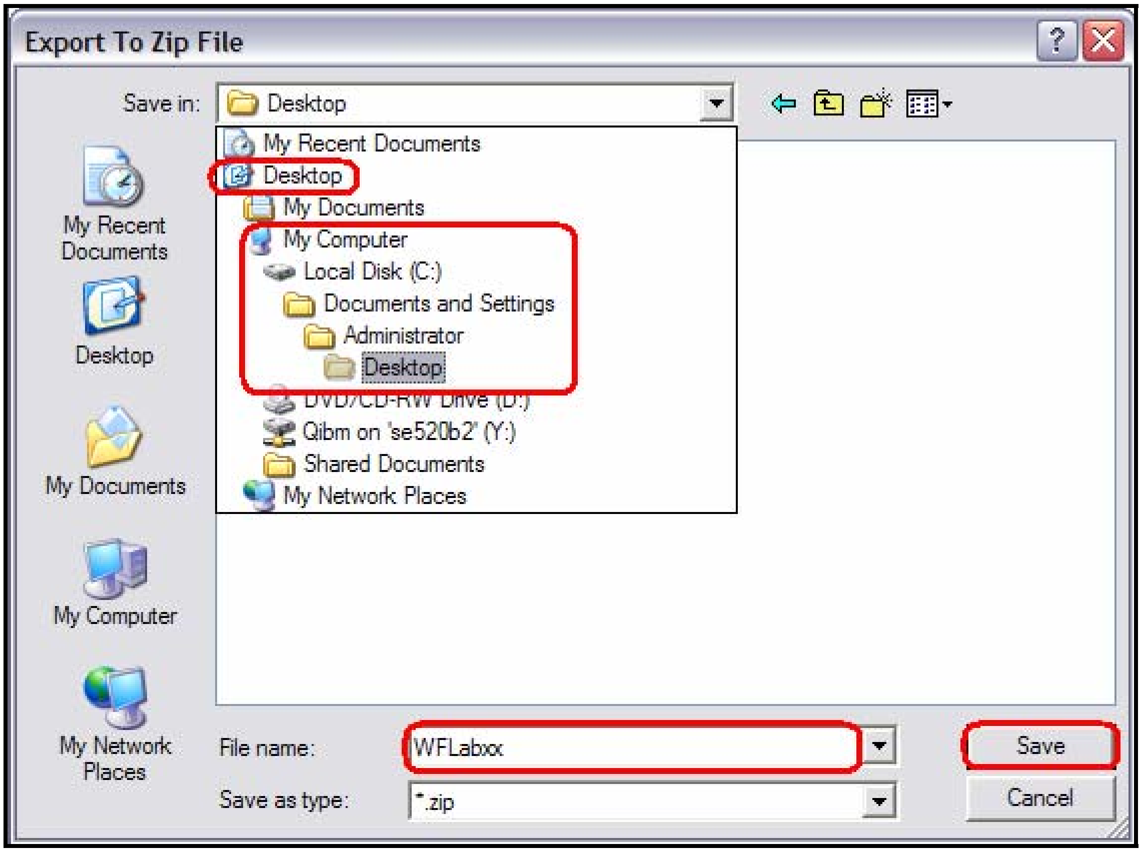
☑ Compress the contents of the file

● Create directory structure for files

○ Create only selected directories

*Figure 182. Check boxes for **wflabxx** and **.project**, then click **Browse***

4. Browse to the Desktop as shown, and type `wflabxx` into **the File name:** input field. The dialog box looks similar to
Figure ***183***. Click **Save**.

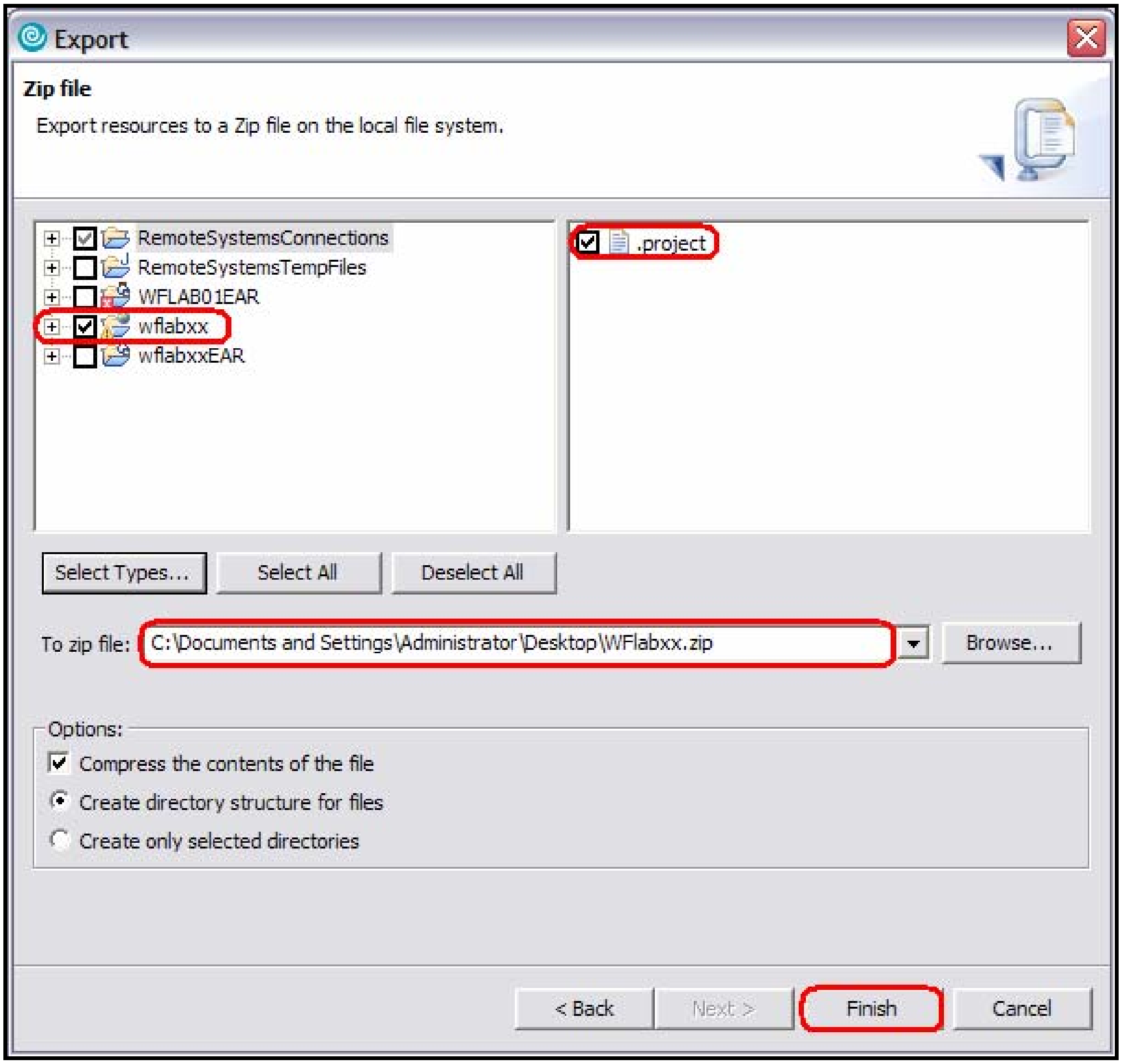


File name: WFLabxx

Save as type: *.zip

*Figure 183. Browse to **Desktop**; in **File name**, type **WFLabxx**.*

5. The **Export Zip file** dialog box now looks similar to Figure 184.



To zip file: C:\Documents and Settings\Administrator\Desktop\WFlabxx.zip

Options:

☑ Compress the contents of the file

● Create directory structure for files

○ Create only selected directories

Figure 184. The Export Zip file dialog

6. Click **Finish**. This creates a .zip file called wflabxx.zip on the Desktop. This file can be passed to another developer or customer who can then use it to import into the WebSphere Development Studio Client to be used in the WebFacing perspective.

Exercise 11.2: Clean up WebSphere Development Studio Client environment

Before you import your IBM WebFacing Tool project back into WebSphere Development Studio Client, you need to remove the project that is already there.

1. Go to the Navigator view in the WebFacing Perspective and highlight both the wflabxx and wflabxxEAR folders. Right-click the folders and select **Delete** (see Figure 185).

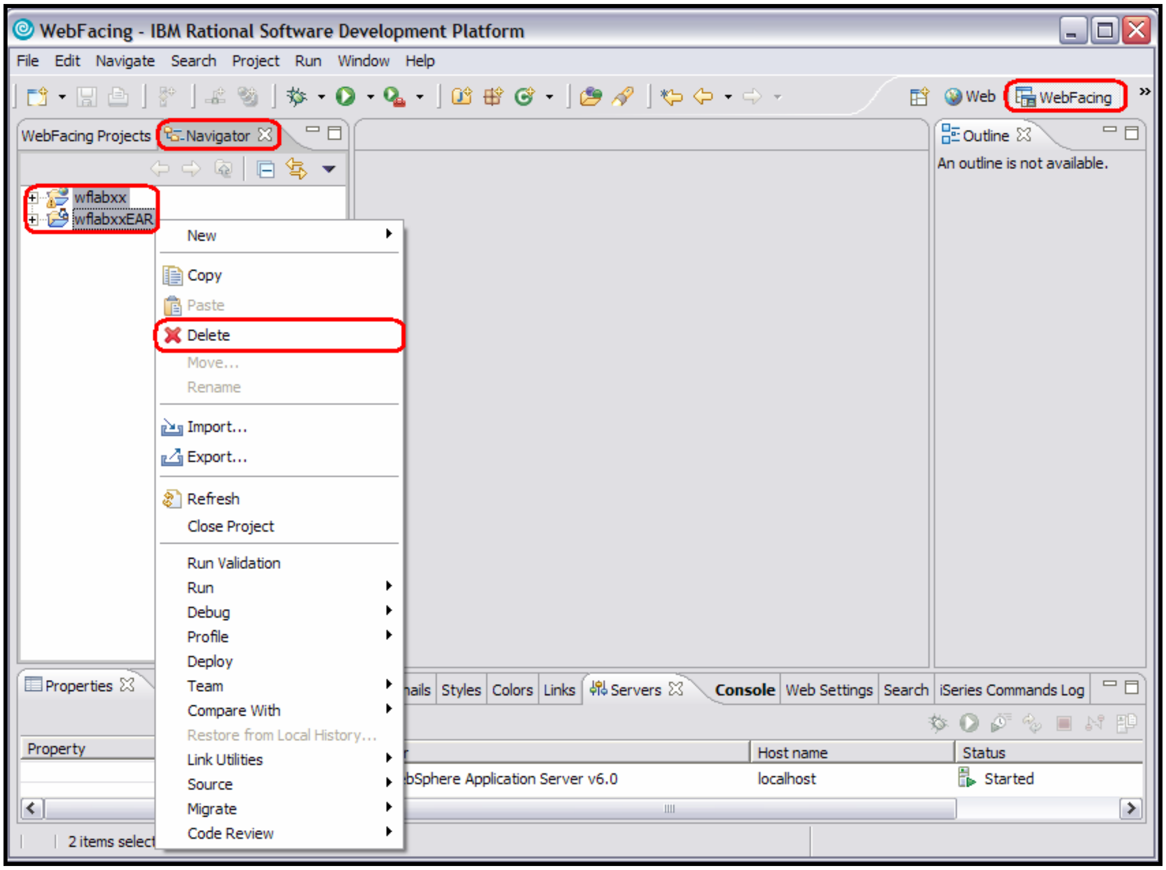
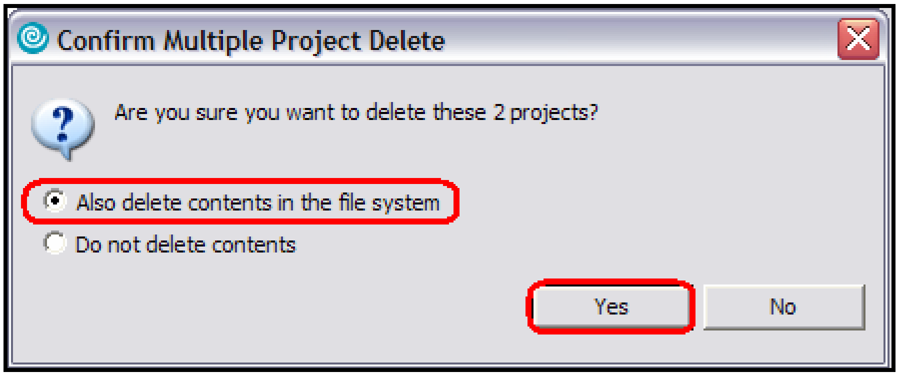


*Figure 185. Delete existing **wflabxx** and **wflabxxEAR** folders*

2. On the confirmation dialog box that follows, select **Also Delete the contents in the file system** as shown in Figure 186. Click **Yes.**



Confirm Multiple Project Delete

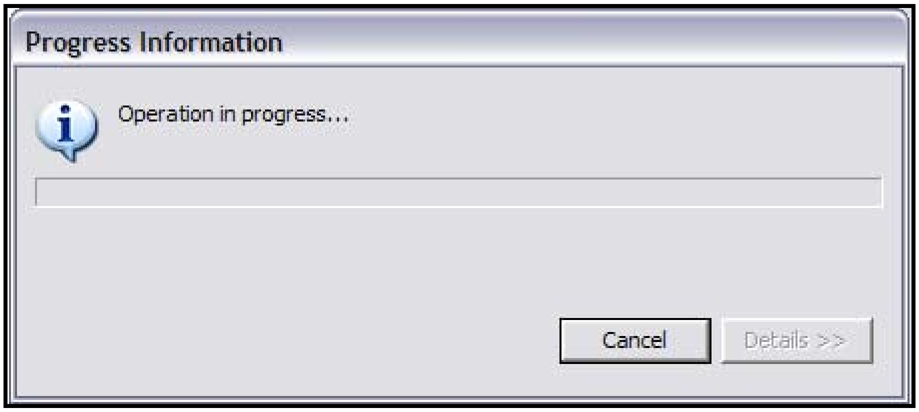
Are you sure you want to delete these 2 projects?

● Also delete contents in the file system

○ Do not delete contents

Figure 186. The Confirm Multiple Project Delete panel

3. A progress indicator panel is then displayed (see Figure 187).

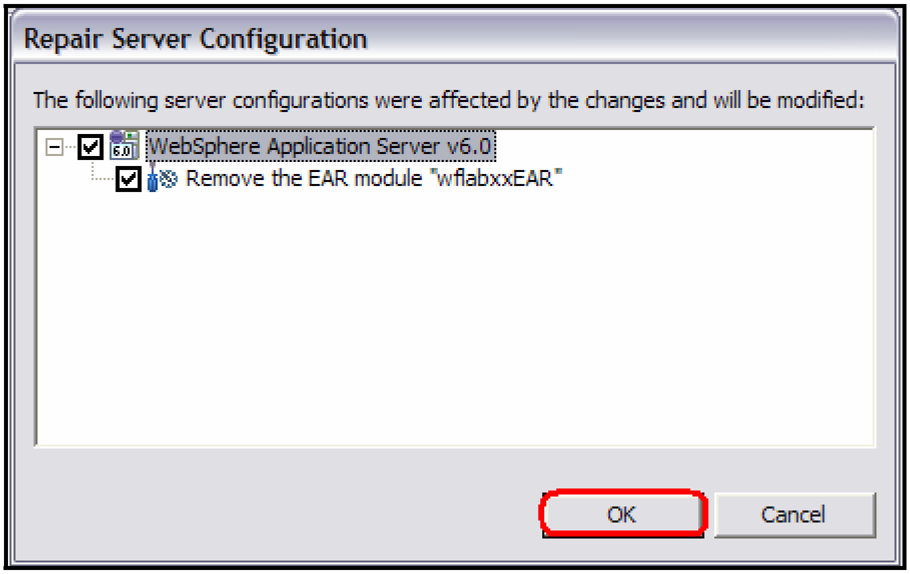


Progress Information

Operation in progress...

Figure 187. The Progress Information panel

4. Finally, a Repair Server Configuration panel is displayed. Select **OK** to proceed (see Figure 188).



Repair Server Configuration

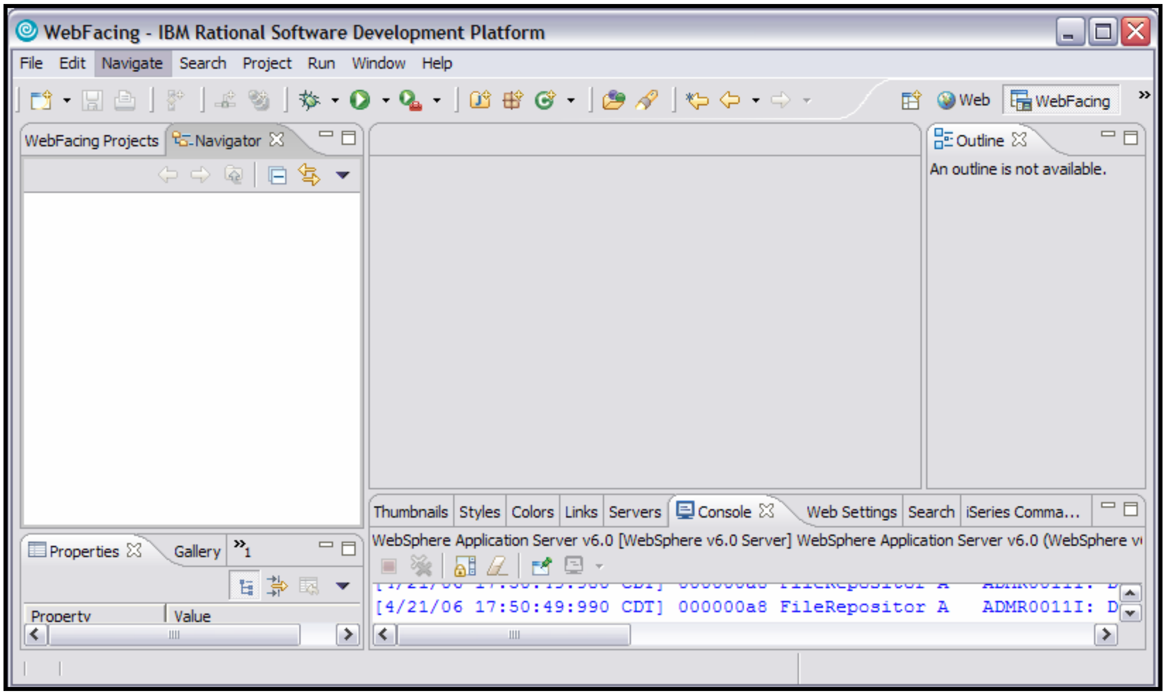
The following server configurations were affected by the changes and will be modified:

☑ WebSphere Application Server v6.0

☑ Remove the EAR module "wflabxxEAR"

Figure 188. The Repair Server Configuration panel

5. When the delete process completes, your existing project in WebSphere Development Studio Client no longer exists and your workspace is empty. The empty workspace is shown in Figure 189.



An outline is not available.

WebSphere Application Server v6.0 [WebSphere v6.0 Server] WebSphere Application Server v6.0 (WebSphere v

```
[4/21/06 17:50:49:990 CDT] 000000a8 FileRepositor A   ADMR0011I: D
```

Figure 189. The empty workspace

Exercise 11.3: Import IBM WebFacing Tool project into WebSphere Development Studio Client

You are now ready to import a WebFacing project into WebSphere Development Studio Client. Before that can happen, you must unzip the WebFacing project .ZIP file.

Import into WebSphere Development Studio Client

To import into WebSphere Development Studio Client, perform the following steps:

5. Double-click the .ZIP file created earlier and unzip this file in C:\temp. When unzipped, the directory structure within the temp subdirectoryfile lookS similar to Figure 190.

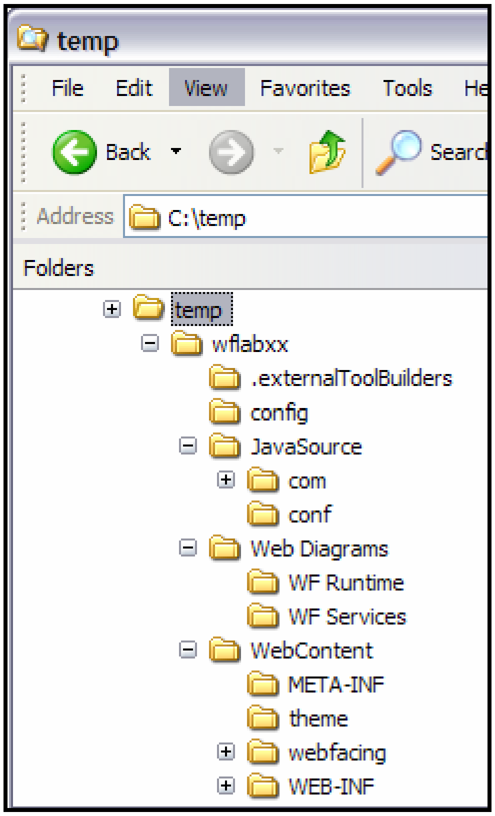


Figure 190. Import into WebSphere Development Studio Client

6. From WebSphere Development Studio Client, select **File > Import** (see Figure 191).

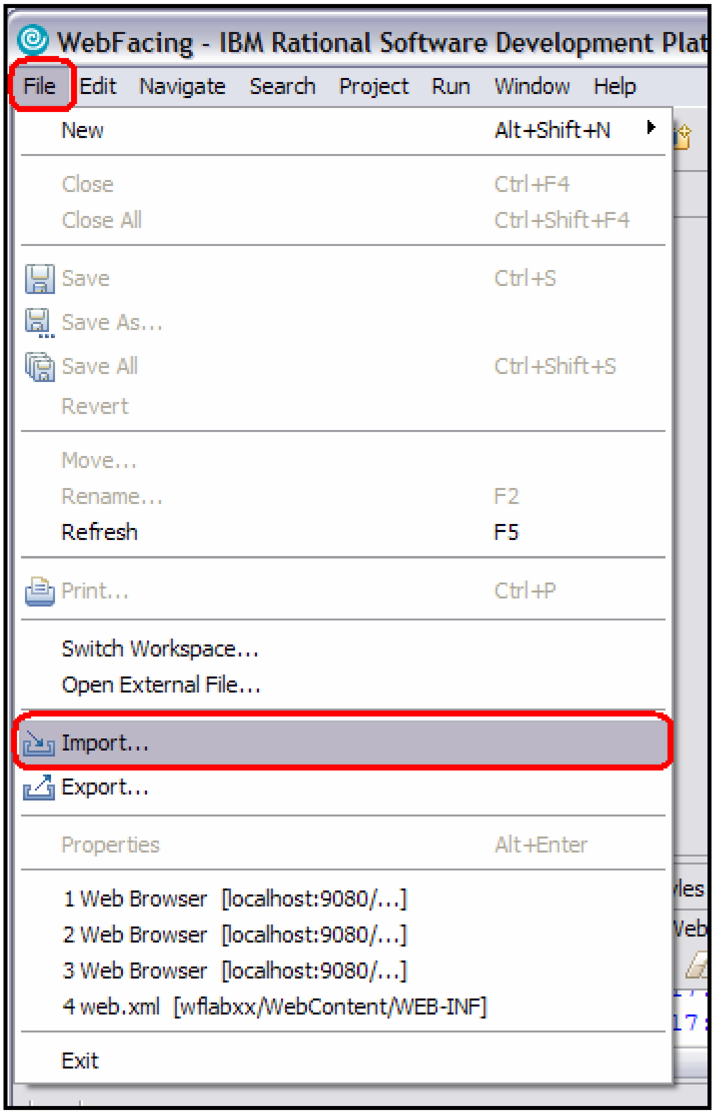


*Figure 191. Choose **Import***

7. On the displayed dialog box, select **WebFacing Project**. Click **Next** (see Figure ***192***).

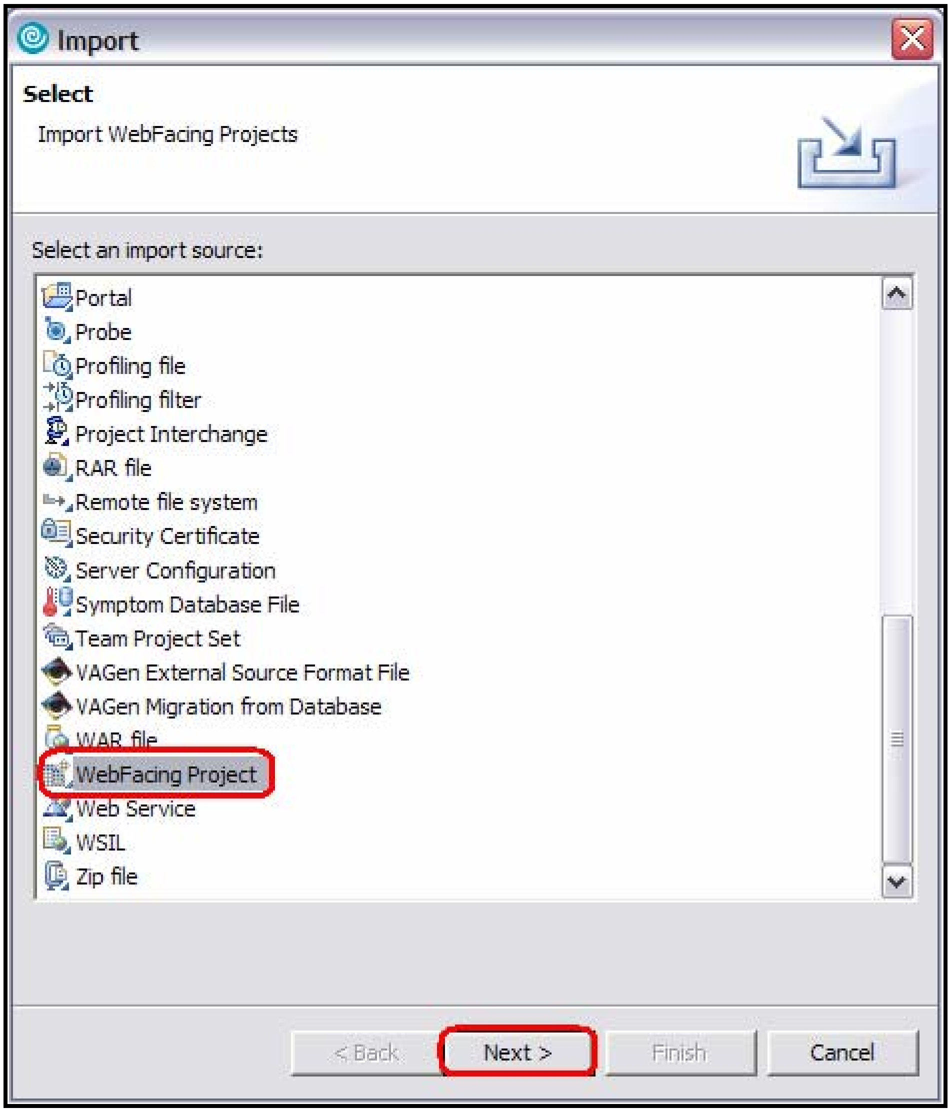
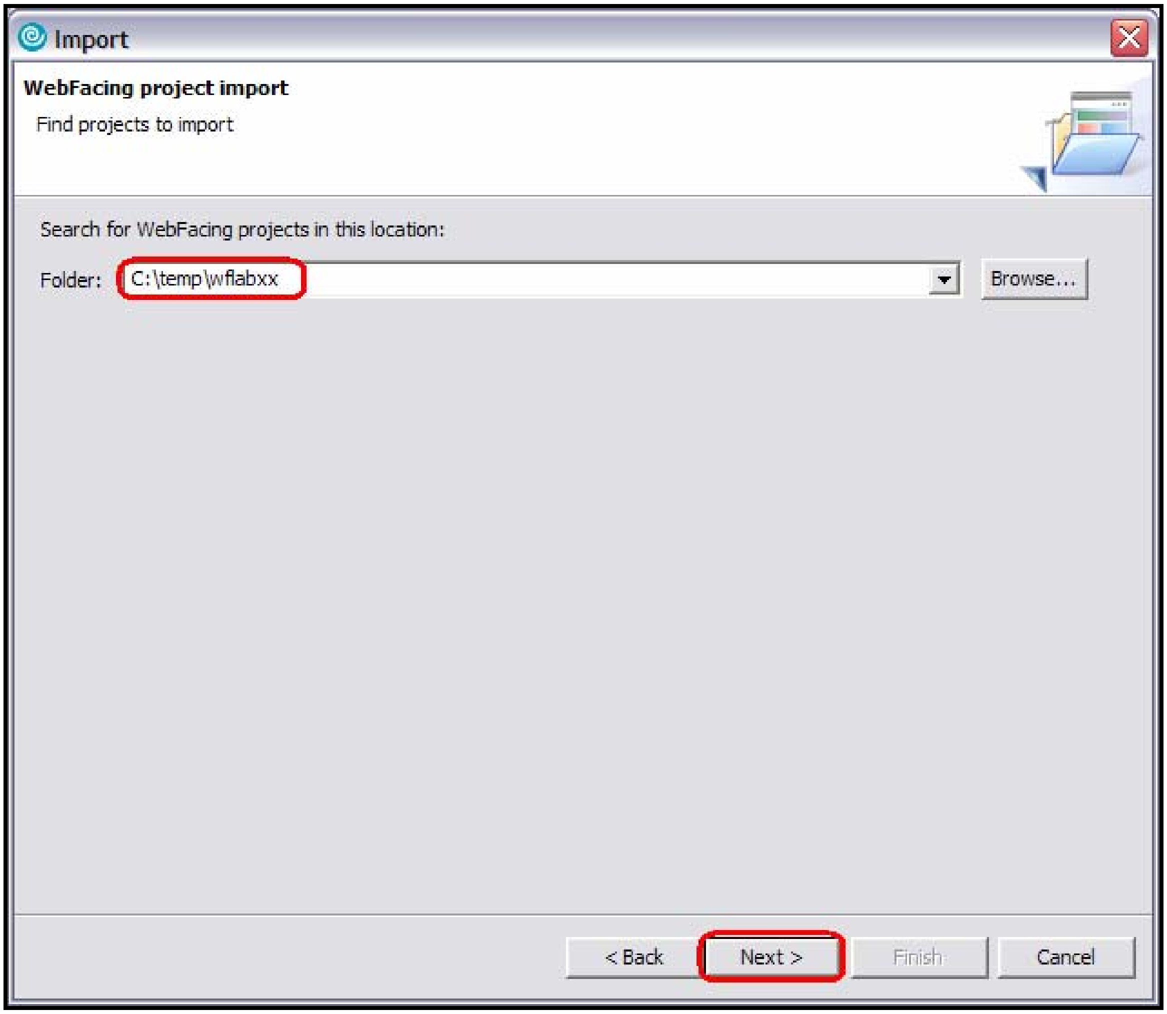


*Figure 192. Choose **WebFacing Project***

8. On the next dialog box, browse to the C:\temp\wflabxx directory and click **Next** (see Figure 193).



Import

WebFacing project import

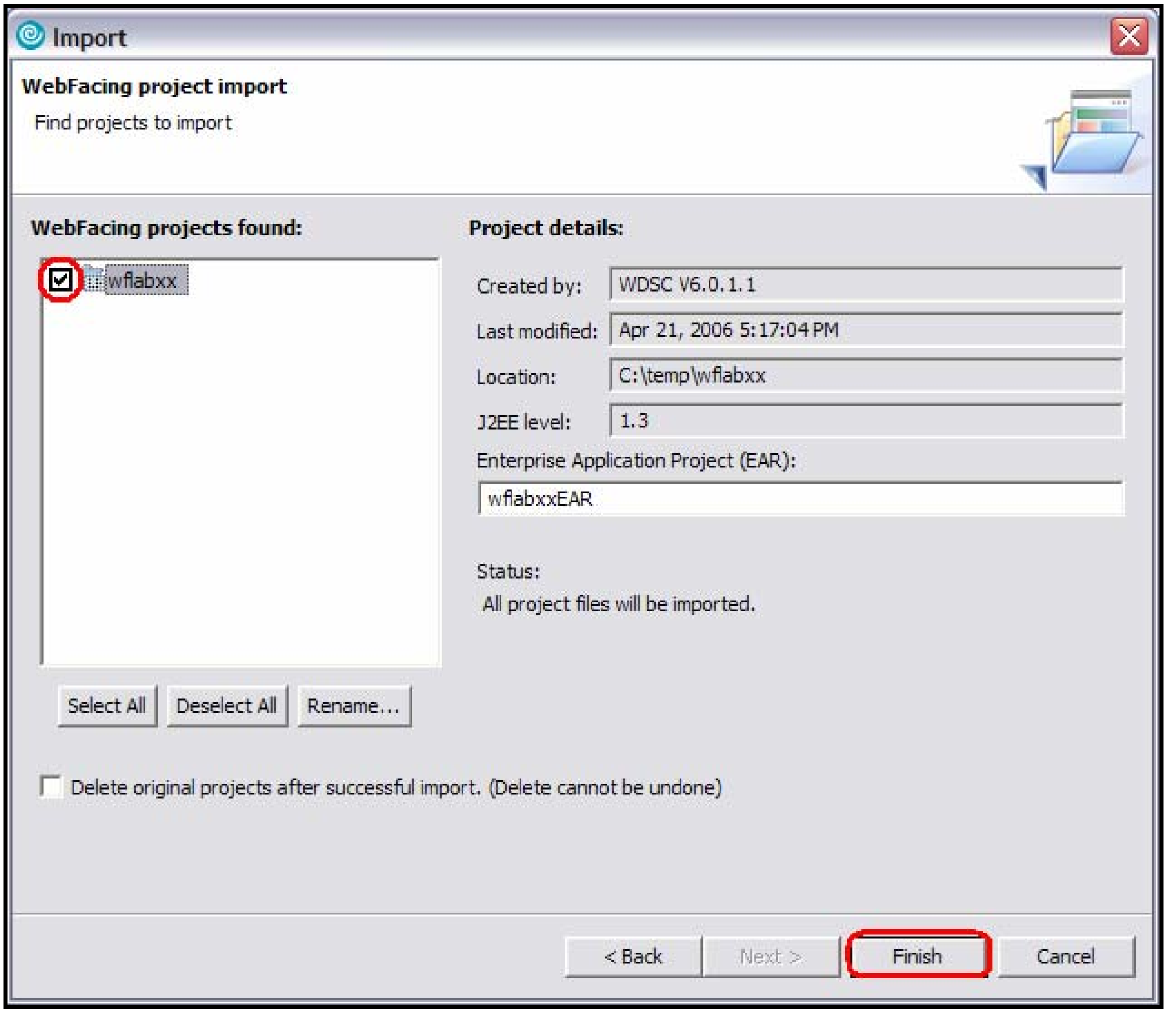
Find projects to import

Search for WebFacing projects in this location:

Folder: C:\temp\wflabxx

*Figure 193. Browse to the **C:\temp\wflabxx** directory and press **Next***

9. On the next dialog box, make sure the wflabxx WebFacing Project is selected. Click **Finish** (see Figure 194).



Import

WebFacing project import

Find projects to import

WebFacing projects found:

☑ wflabxx

Project details:

Created by: WDSC V6.0.1.1

Last modified: Apr 21, 2006 5:17:04 PM

Location: C:\temp\wflabxx

J2EE level: 1.3

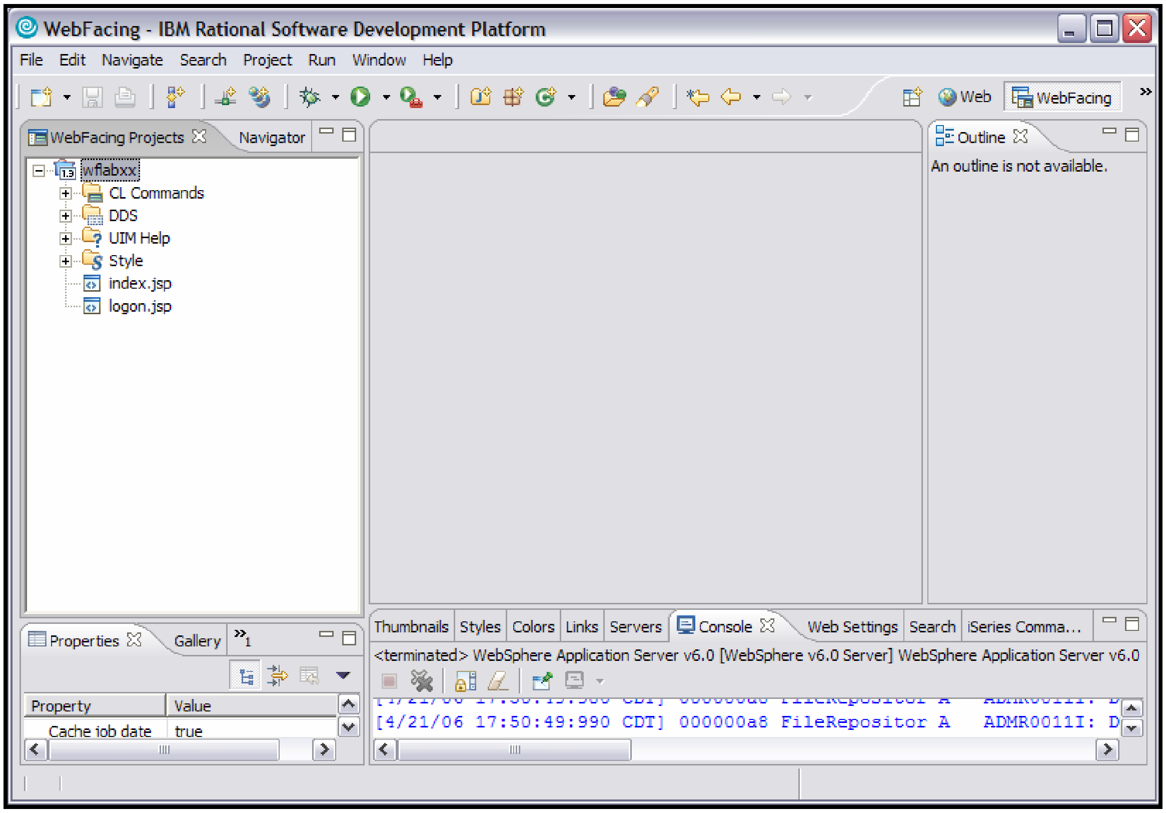
Enterprise Application Project (EAR): wflabxxEAR

Status:
All project files will be imported.

☐ Delete original projects after successful import. (Delete cannot be undone)

*Figure 194. Select the **wflabxx** WebFacing Project and click **Finish***

10. This imports the wflabxx WebFacing project back in to the WebSphere Development Studio Client environment. Go to WebSphere Development Studio Client and select the WebFacing perspective. The WebFacing project is now restored (see Figure 195).



An outline is not available.

Property	Value
Cache iob date	true

```
<terminated> WebSphere Application Server v6.0 [WebSphere v6.0 Server] WebSphere Application Server v6.0
[4/21/06 17:50:49:990 CDT] 000000a8 FileRepositor A   ADMR0011I: D
```

Figure 195. Restoring the WebFacing project

11. Delete the **wflabxx** subdirectory and all of its contents from within the **c:\temp** directory. This resets the c:\temp library to a default state, which makes it easier to follow the directions during the partial deployment exercise.

Recap

You have now completed the “Export/Import IBM WebFacing Tool projects in WebSphere Development Studio Client” section. You now have the information to understand how to:

- Export your IBM WebFacing Tool project from WebSphere Development Studio Client
- Clean up WebSphere Development Studio Client environment
- Import your IBM WebFacing Tool project into WebSphere Development Studio Client

Exporting to System i WebSphere Application Server Express V6.0.2.7

You have tested the WebFacing application in the WebSphere Application Server test environment and you are now ready to move it to your System i WebSphere Application Server production environment. You are running WebSphere Application Server V6.0.2.7 on your System i host server.

In this chapter, you learn how to publish your WebFacing files to a remote WebSphere Application Server V6.0.2.7 environment on a System i host server. You start the export tool in WebSphere Development Studio Client, specify files to export, install, and start the Web application in the WebSphere Application Server Administrative console.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 11.1: Exporting the files to the Web server
- Exercise 11.2: Exporting your EAR file
- Exercise 11.3: Installing the application
- Exercise 11.4: Testing the WebFacing application

The exercises in this chapter must be completed in order. Start with “Exercise 11.1: Exporting the files to the Web server” when you are ready to begin.

Length of time

This chapter takes approximately 30 minutes to complete.

Exercise 12.1: Exporting the files to the Web server

First, copy the WebFacing project files to a WebSphere Application Server (V6.0.2.7). These first steps are the same for WebSphere Application Server for iSeries (V6.0.2.7) and the WebSphere Application Server for iSeries Express (V6.0.2.71).

Before you export any files, you must map a network drive to the Root file system on the System i model where WebSphere Application Server is installed.

To map a network drive:

1. Right-click the **Network Neighborhood** or (**My Network Places**) icon on the desktop (see Figure 196).

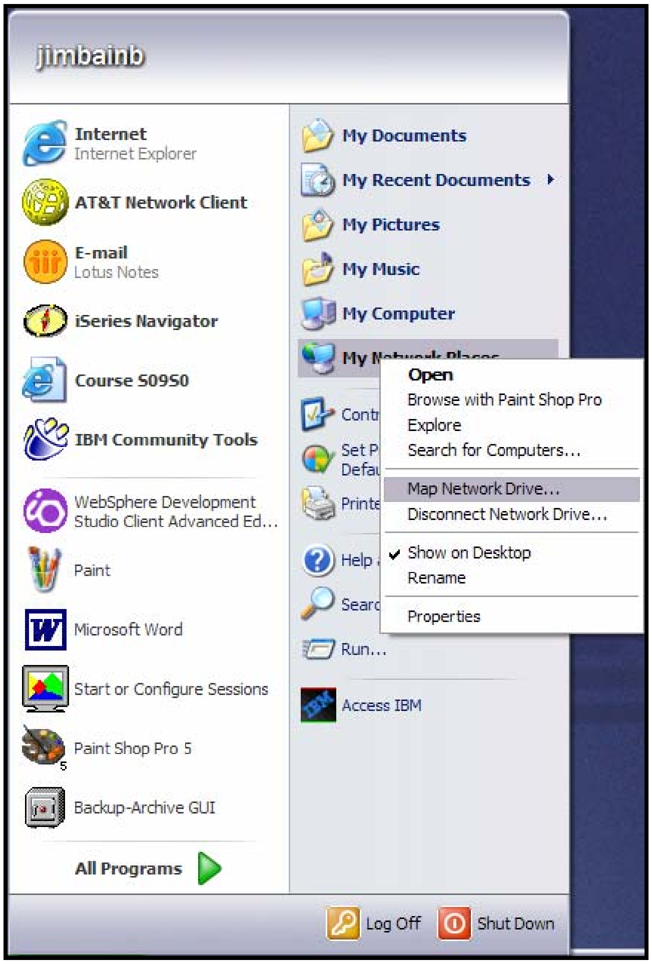
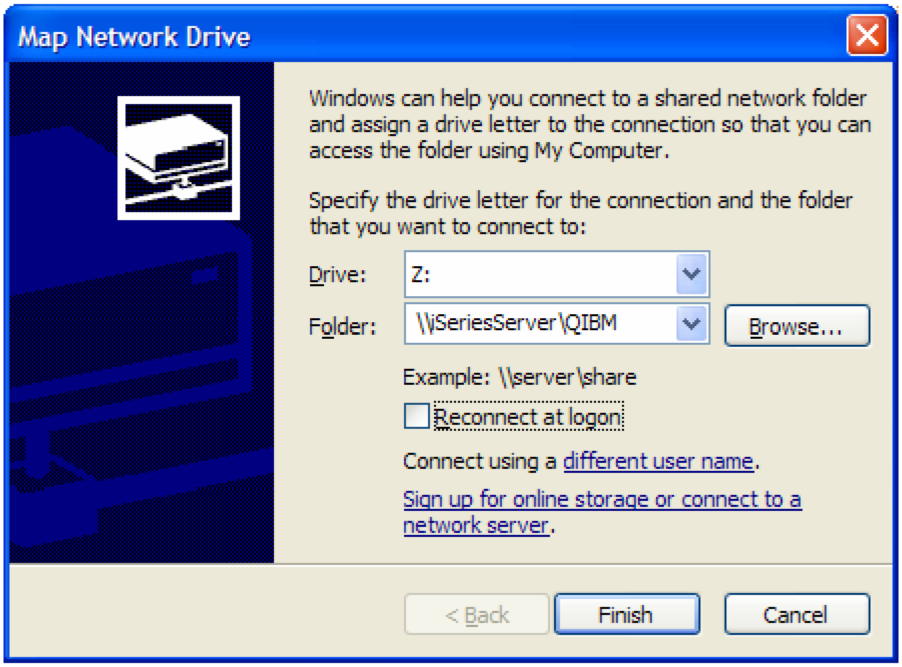


Figure 196. Exporting files to the Web server

2. Select **Map Network drive** on the pop-up menu.

   The Map Network Drive dialog opens (see Figure 197).



Map Network Drive

Windows can help you connect to a shared network folder and assign a drive letter to the connection so that you can access the folder using My Computer.

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Z:

Folder: \\iSeriesServer\QIBM

Example: \\server\share

□ Reconnect at logon

Connect using a different user name.

Sign up for online storage or connect to a network server.

Figure 197. The Map Drive dialog

3. In the **Folder** field, enter two backslashes \\ and the Netserver name of your Web server, and then **\QIBM**, for example: \\iSeriesServer\QIBM. This is a default share on System i models. If the share is not active, you have to activate it or create a new share using iSeries Navigator.
4. Click **Finish**.

   You now have an additional drive available on your workstation and are ready to export your WebFacing project files.

Exercise 12.2: Exporting your EAR file

The J2EE standard has the concept of an Enterprise Archive File (EAR). This file is a Zip file that contains all information about your Web application. In the workbench, this file is built automatically for Web projects. Because your WebFacing project is a Web project, the wflabxxEAR file exists already. You only have to move it to the application server, point to it, and then install the application. There is no need to know about the structure of the application. Everything is handled in the workbench. Return to the WebSphere Development Studio Client workbench.

To export your EAR file:

1. In the WebFacing perspective, click **File** from the workbench menu.
2. Click **Export** on the pop-up menu (see Figure 198).

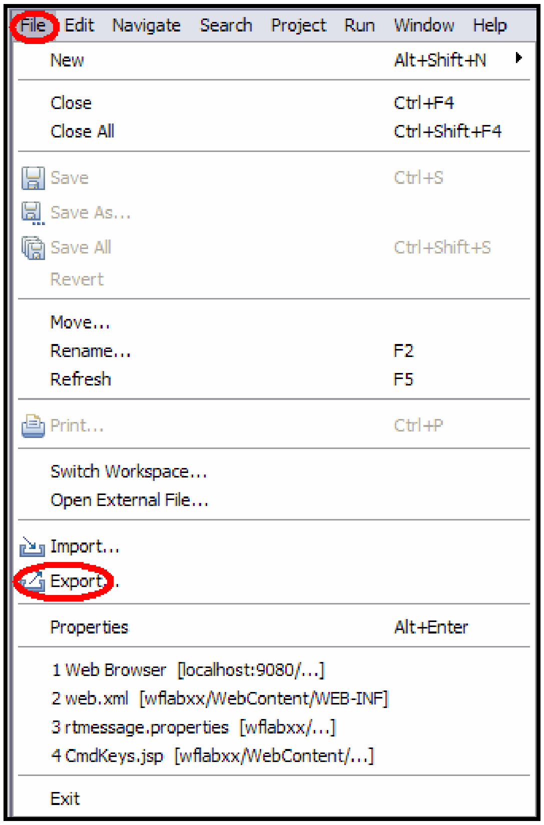
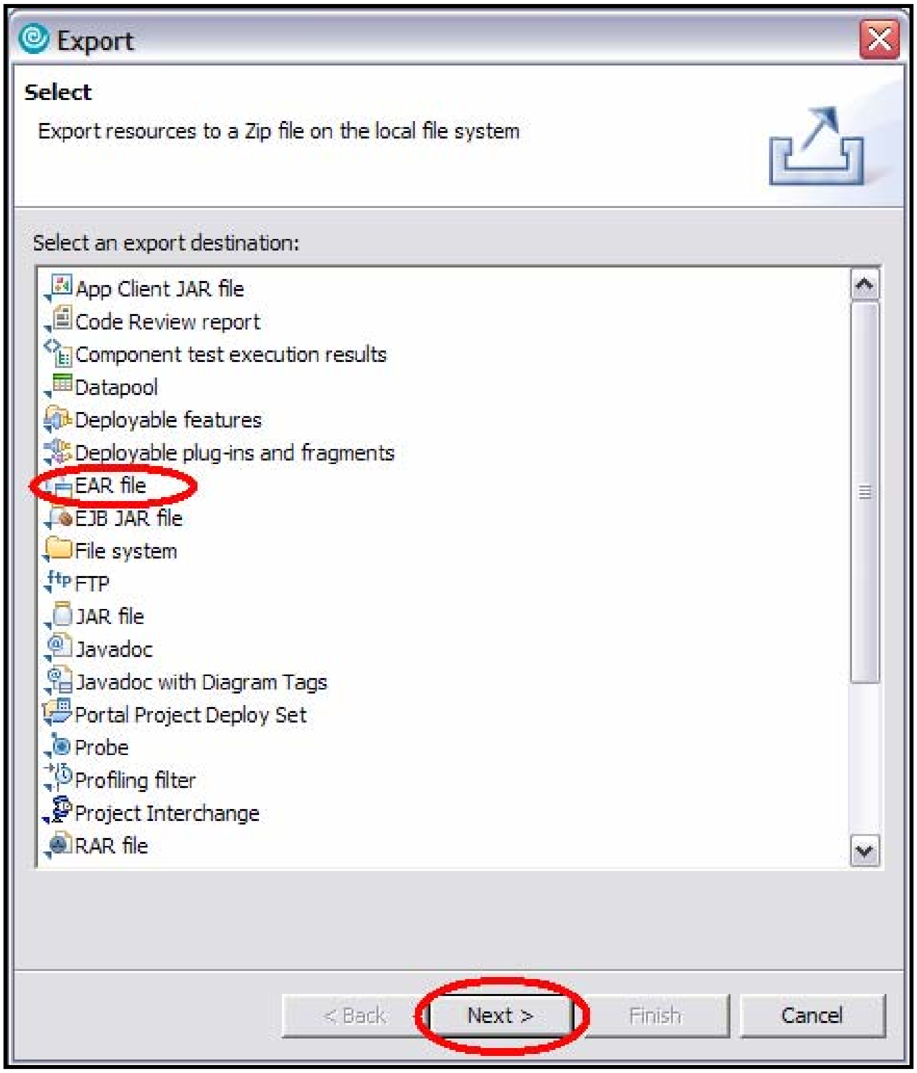


*Figure 198. Exporting your **EAR** file*

The Select page of the Export wizard opens (see Figure 199).



Select

Export resources to a Zip file on the local file system

Select an export destination:

- App Client JAR file
- Code Review report
- Component test execution results
- Datapool
- Deployable features
- Deployable plug-ins and fragments
- EAR file
- EJB JAR file
- File system
- FTP
- JAR file
- Javadoc
- Javadoc with Diagram Tags
- Portal Project Deploy Set
- Probe
- Profiling filter
- Project Interchange
- RAR file

Figure 199. The Select page of the Export wizard

3. Select the **EAR file** icon. An EAR file is a compressed Enterprise Application Archive, and it contains all the files needed for the Web application.
4. Click **Next**.

The **EAR Export** page opens (see Figure 200).

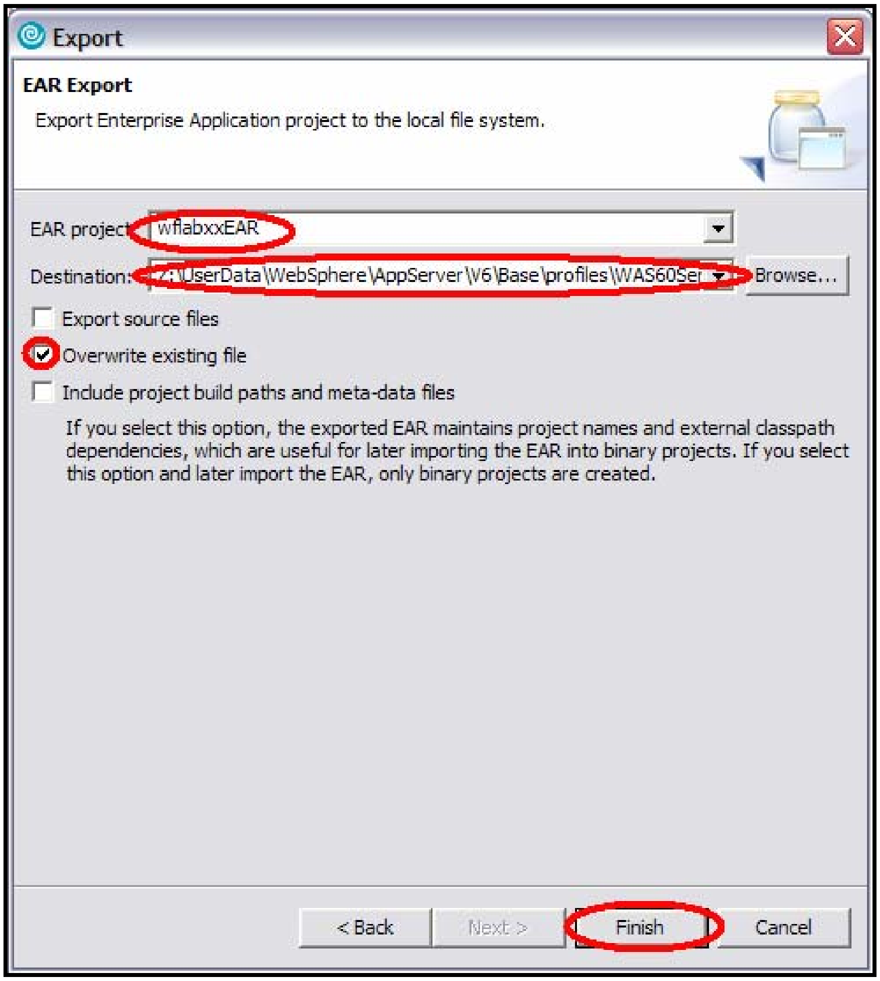

Export
EAR Export
Export Enterprise Application project to the local file system.
EAR project: wflabxxEAR
Destination: Z:\UserData\WebSphere\AppServer\V6\Base\profiles\WAS60Ser
Browse...
□ Export source files
☑ Overwrite existing file
□ Include project build paths and meta-data files
If you select this option, the exported EAR maintains project names and external classpath dependencies, which are useful for later importing the EAR into binary projects. If you select this option and later import the EAR, only binary projects are created.
< Back
Next >
Finish
Cancel

Figure 200. The EAR Export page

5. Select the name of the resources you used when creating your WebFacing project. This must be wflabxxEAR from the **Enterprise Application project name** list.

In the **Destination** field, specify where you want to put the EAR file by entering the drive and directory structure of the IFS drive you mapped in the steps before. For example, use the default directory structure for installable applications for WebSphere Application Server Express. This path is dependant on the version and naming of your WebSphere Application Server and instance naming. It is not so important where you put it, but that you can find it when you try to install it. In our example, this path is Z:\UserData\WebSphere\AppServer\V6\Base\profiles\WAS60Server\installableApps, where Z: was the drive letter that was mapped in the previous steps.

6. Select the **Overwrite existing files without warning** check box.
7. Click **Finish**.

If a message displays, asking you to create a directory or delete a file, Click **OK** or **Yes** and continue. Now you can install the application in WebSphere Application Server.

Exercise 12.3: Installing the application

At this point, you should already have WebSphere Application Server and the associated HTTP server on your System i model set up and started. In these instructions, the port number of the HTTP Servers Administrative Console is referred to as Port #a, and the port number to run your application is referred to as Port #b. Port #b is the port number of your HTTP server that is set up to work with your application server. These values were defined during the WebSphere Application server configuration.

To install the application:

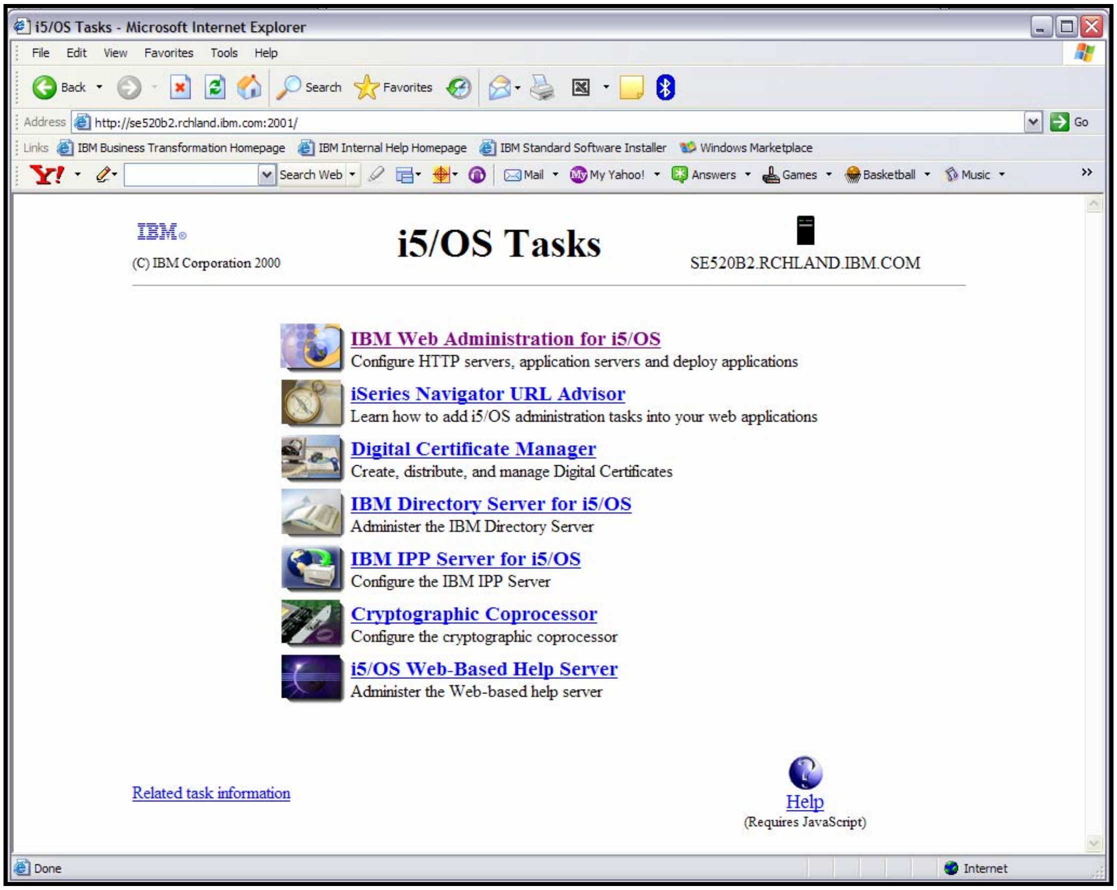
1. Open any Web browser and type: http://servername:port#a/ in the **Address** field.

   Note: The default administrative port number for the HTTP server is 2001.

   The Login dialog appears.

2. Type your i5/OS user ID in the **User ID** field.
3. Type your System i password in the **Password** field.
4. Click **OK**.

The browser opens (see Figure 201).



i5/OS Tasks - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://se520b2.rchland.ibm.com:2001/
Go
Links
IBM Business Transformation Homepage
IBM Internal Help Homepage
IBM Standard Software Installer
Windows Marketplace
Search Web
Mail
My Yahoo!
Answers
Games
Basketball
Music
Done
Internet

IBM
(C) IBM Corporation 2000

i5/OS Tasks

SE520B2.RCHLAND.IBM.COM

IBM Web Administration for i5/OS
Configure HTTP servers, application servers and deploy applications

iSeries Navigator URL Advisor
Learn how to add i5/OS administration tasks into your web applications

Digital Certificate Manager
Create, distribute, and manage Digital Certificates

IBM Directory Server for i5/OS
Administer the IBM Directory Server

IBM IPP Server for i5/OS
Configure the IBM IPP Server

Cryptographic Coprocessor
Configure the cryptographic coprocessor

i5/OS Web-Based Help Server
Administer the Web-based help server

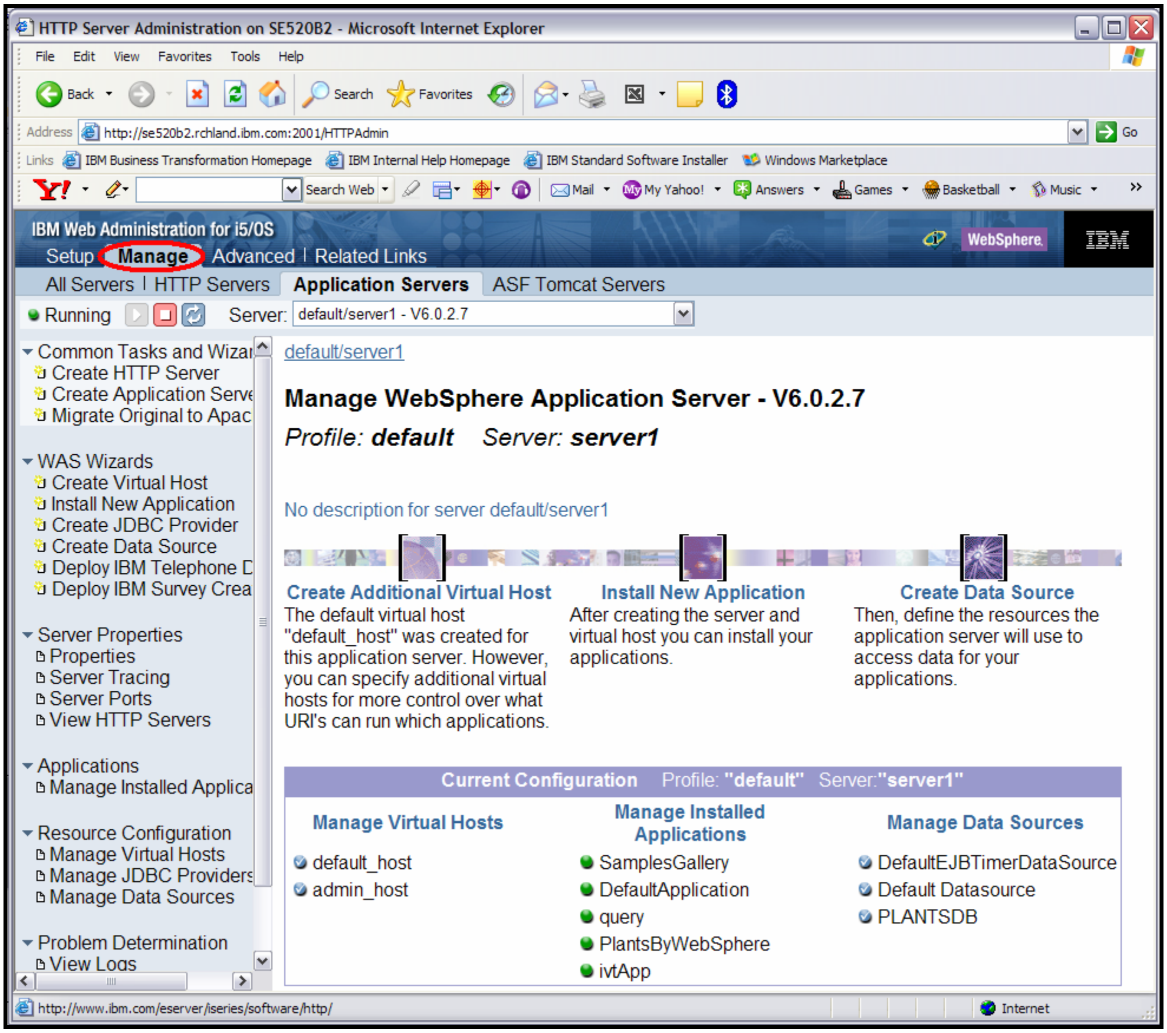
Related task information

Help
(Requires JavaScript)

Figure 201. Installing the application

5. Click **IBM Web Administration for i5/OS** from the i5/OS Tasks menu.

6. The **IBM Web Administration for i5/OS** pages are displayed. Select the **Manage** Tab if not already shown (see Figure 202).


HTTP Server Administration on SE520B2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://se520b2.rchland.ibm.com:2001/HTTPAdmin
Go
Links IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer Windows Marketplace
Y! Search Web Mail My Yahoo! Answers Games Basketball Music
IBM Web Administration for i5/OS
WebSphere
IBM
Setup Manage Advanced Related Links
All Servers HTTP Servers Application Servers ASF Tomcat Servers
Running
Server: default/server1 - V6.0.2.7
Common Tasks and Wizar
Create HTTP Server
Create Application Serve
Migrate Original to Apac
WAS Wizards
Create Virtual Host
Install New Application
Create JDBC Provider
Create Data Source
Deploy IBM Telephone D
Deploy IBM Survey Crea
Server Properties
Properties
Server Tracing
Server Ports
View HTTP Servers
Applications
Manage Installed Applica
Resource Configuration
Manage Virtual Hosts
Manage JDBC Providers
Manage Data Sources
Problem Determination
View Logs
default/server1
http://www.ibm.com/eserver/iseries/software/http/
Internet

Manage WebSphere Application Server - V6.0.2.7

Profile: ***default*** *Server:* ***server1***

No description for server default/server1

Create Additional Virtual Host
The default virtual host "default_host" was created for this application server. However, you can specify additional virtual hosts for more control over what URI's can run which applications.

Install New Application
After creating the server and virtual host you can install your applications.

Create Data Source
Then, define the resources the application server will use to access data for your applications.

Current Configuration Profile: **"default"** Server:**"server1"**

Manage Virtual Hosts	Manage Installed Applications	Manage Data Sources
default_host	SamplesGallery	DefaultEJBTimerDataSource
admin_host	DefaultApplication	Default Datasource
	query	PLANTSDB
	PlantsByWebSphere	
	ivtApp	

Figure 202. ***IBM Web Administration for i5/OS*** *pages*

7. Select your WebSphere Application Server instance from the **Server** dropdown list (see Figure 203).

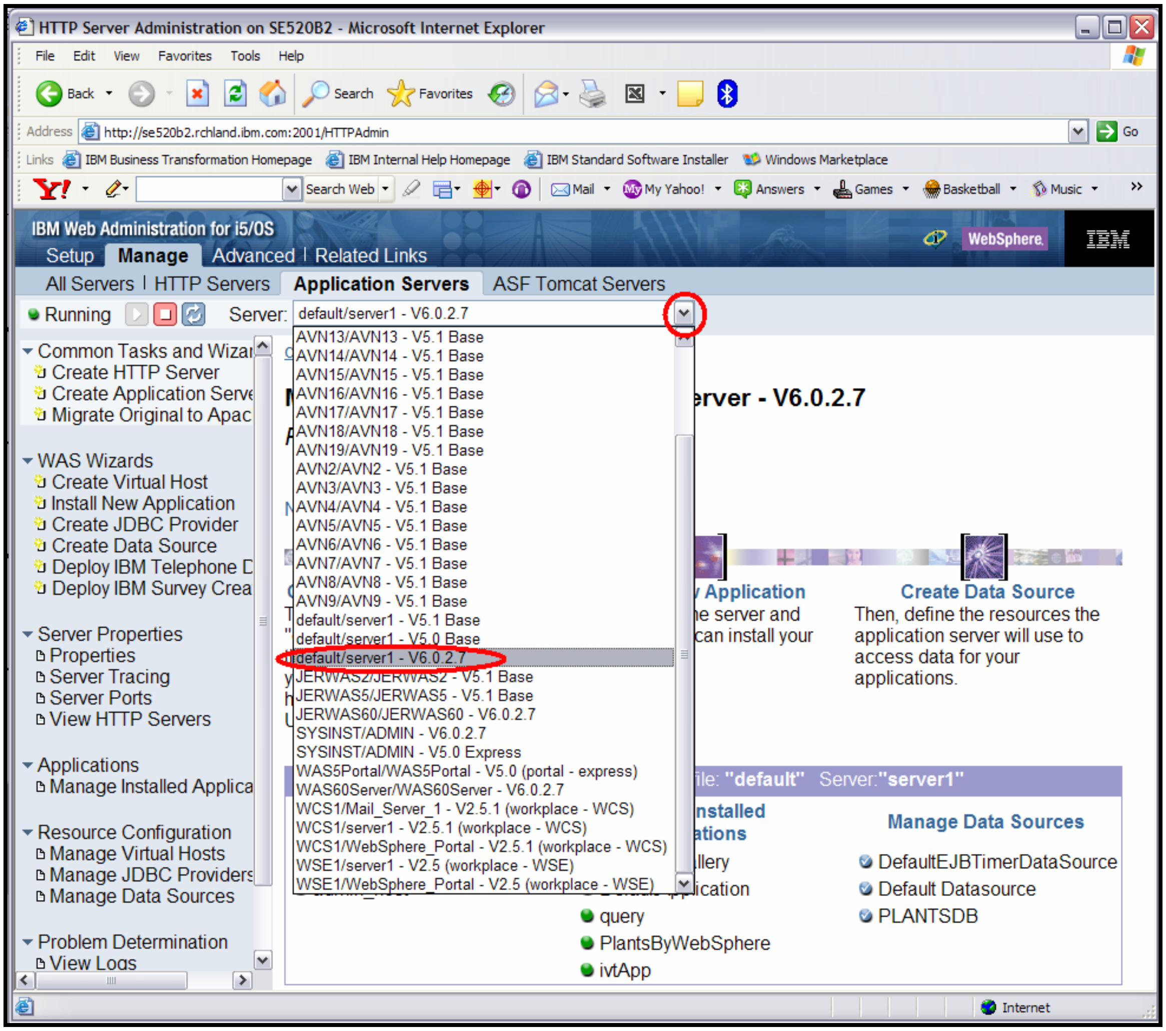
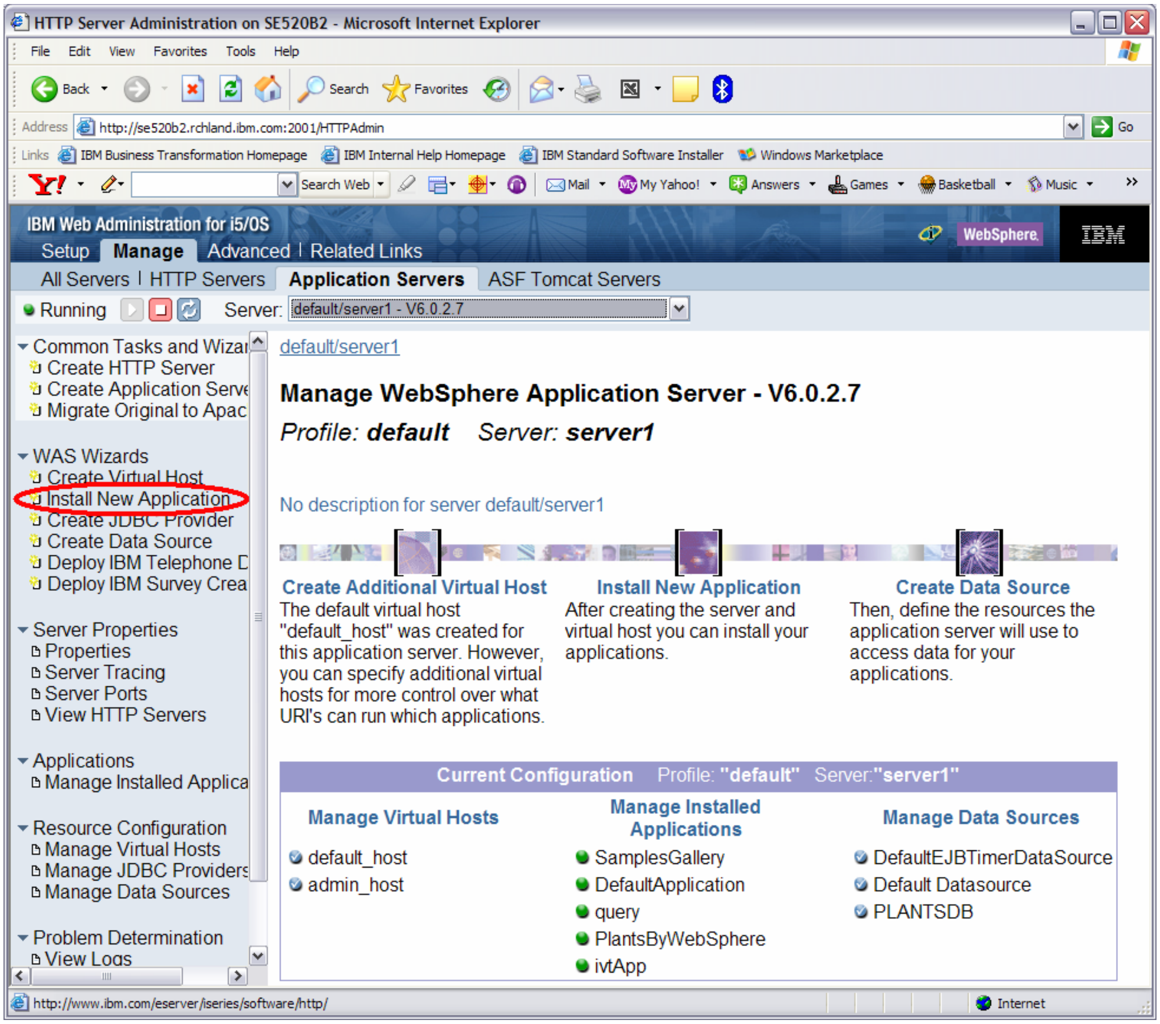

HTTP Server Administration on SE520B2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://se520b2.rchland.ibm.com:2001/HTTPAdmin
Go
Links IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer Windows Marketplace
Search Web Mail My Yahoo! Answers Games Basketball Music
IBM Web Administration for i5/OS
Setup Manage Advanced Related Links
WebSphere
IBM
All Servers HTTP Servers Application Servers ASF Tomcat Servers
Running
Server: default/server1 - V6.0.2.7
AVN13/AVN13 - V5.1 Base
AVN14/AVN14 - V5.1 Base
AVN15/AVN15 - V5.1 Base
AVN16/AVN16 - V5.1 Base
AVN17/AVN17 - V5.1 Base
AVN18/AVN18 - V5.1 Base
AVN19/AVN19 - V5.1 Base
AVN2/AVN2 - V5.1 Base
AVN3/AVN3 - V5.1 Base
AVN4/AVN4 - V5.1 Base
AVN5/AVN5 - V5.1 Base
AVN6/AVN6 - V5.1 Base
AVN7/AVN7 - V5.1 Base
AVN8/AVN8 - V5.1 Base
AVN9/AVN9 - V5.1 Base
default/server1 - V5.1 Base
default/server1 - V5.0 Base
default/server1 - V6.0.2.7
JERWAS2/JERWAS2 - V5.1 Base
JERWAS5/JERWAS5 - V5.1 Base
JERWAS60/JERWAS60 - V6.0.2.7
SYSINST/ADMIN - V6.0.2.7
SYSINST/ADMIN - V5.0 Express
WAS5Portal/WAS5Portal - V5.0 (portal - express)
WAS60Server/WAS60Server - V6.0.2.7
WCS1/Mail_Server_1 - V2.5.1 (workplace - WCS)
WCS1/server1 - V2.5.1 (workplace - WCS)
WCS1/WebSphere_Portal - V2.5.1 (workplace - WCS)
WSE1/server1 - V2.5 (workplace - WSE)
WSE1/WebSphere_Portal - V2.5 (workplace - WSE)
Common Tasks and Wizar
Create HTTP Server
Create Application Serve
Migrate Original to Apac
WAS Wizards
Create Virtual Host
Install New Application
Create JDBC Provider
Create Data Source
Deploy IBM Telephone D
Deploy IBM Survey Crea
Server Properties
Properties
Server Tracing
Server Ports
View HTTP Servers
Applications
Manage Installed Applica
Resource Configuration
Manage Virtual Hosts
Manage JDBC Providers
Manage Data Sources
Problem Determination
View Logs
erver - V6.0.2.7
Application
e server and
can install your
Create Data Source
Then, define the resources the application server will use to access data for your applications.
ile: "default" Server:"server1"
nstalled
ations
llery
ication
query
PlantsByWebSphere
ivtApp
Manage Data Sources
DefaultEJBTimerDataSource
Default Datasource
PLANTSDB
Internet

Figure 203. Select your WebSphere Application Server instance.

8. Click **Install New Application** under the **WebSphere Application Server Wizards** list (see Figure 204).



default/server1

Manage WebSphere Application Server - V6.0.2.7

Profile: ***default*** *Server:* ***server1***

No description for server default/server1

Create Additional Virtual Host
The default virtual host "default_host" was created for this application server. However, you can specify additional virtual hosts for more control over what URI's can run which applications.

Install New Application
After creating the server and virtual host you can install your applications.

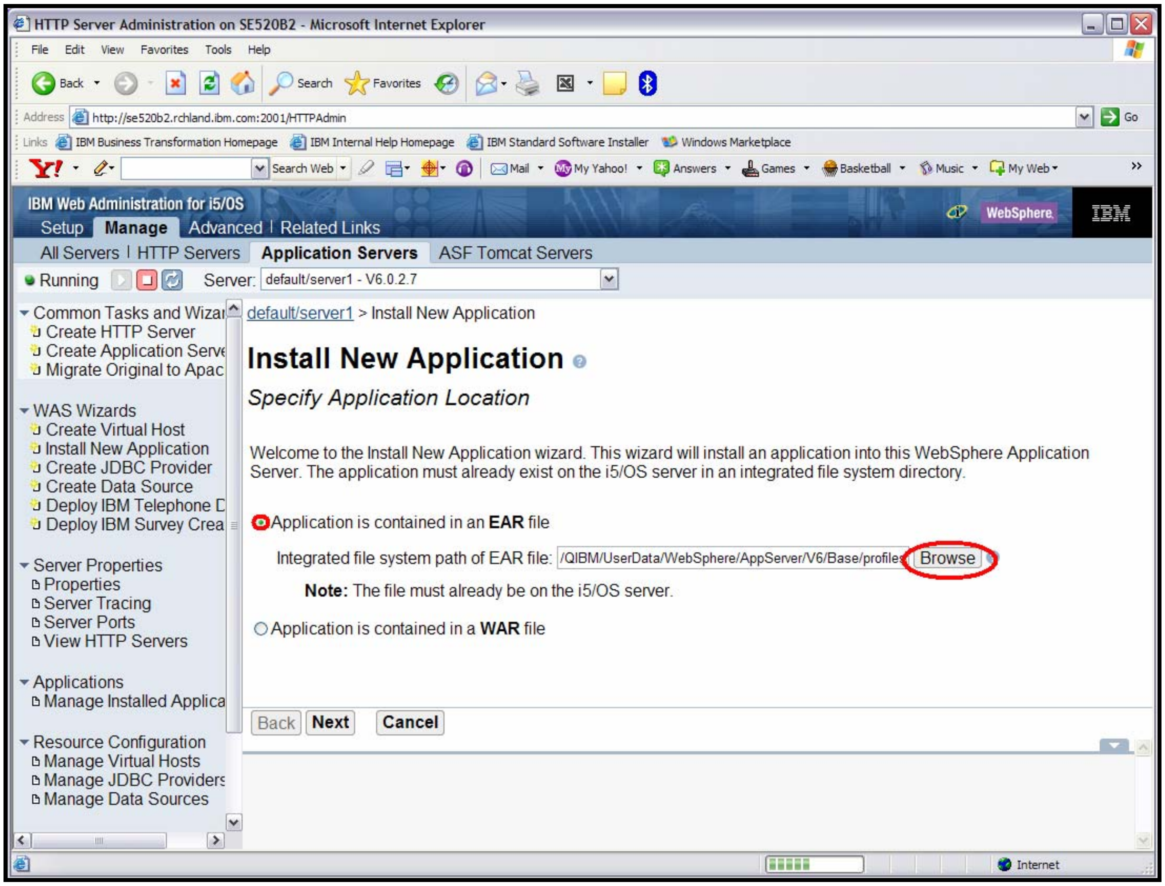
Create Data Source
Then, define the resources the application server will use to access data for your applications.

Current Configuration Profile: **"default"** Server:**"server1"**

Manage Virtual Hosts	Manage Installed Applications	Manage Data Sources
default_host	SamplesGallery	DefaultEJBTimerDataSource
admin_host	DefaultApplication	Default Datasource
	query	PLANTSDB
	PlantsByWebSphere	
	ivtApp	

*Figure 204. Click **Install New Application***

The Install New Application – Specify Application Location panel is displayed (see Figure 205).


HTTP Server Administration on SE520B2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://se520b2.rchland.ibm.com:2001/HTTPAdmin
IBM Web Administration for i5/OS
Setup Manage Advanced Related Links
All Servers HTTP Servers Application Servers ASF Tomcat Servers
Running Server: default/server1 - V6.0.2.7
Common Tasks and Wizards
Create HTTP Server
Create Application Server
Migrate Original to Apache
WAS Wizards
Create Virtual Host
Install New Application
Create JDBC Provider
Create Data Source
Deploy IBM Telephone D
Deploy IBM Survey Crea
Server Properties
Properties
Server Tracing
Server Ports
View HTTP Servers
Applications
Manage Installed Applica
Resource Configuration
Manage Virtual Hosts
Manage JDBC Providers
Manage Data Sources
default/server1 > Install New Application

Install New Application

Specify Application Location

Welcome to the Install New Application wizard. This wizard will install an application into this WebSphere Application Server. The application must already exist on the i5/OS server in an integrated file system directory.

● Application is contained in an **EAR** file

Integrated file system path of EAR file: /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles

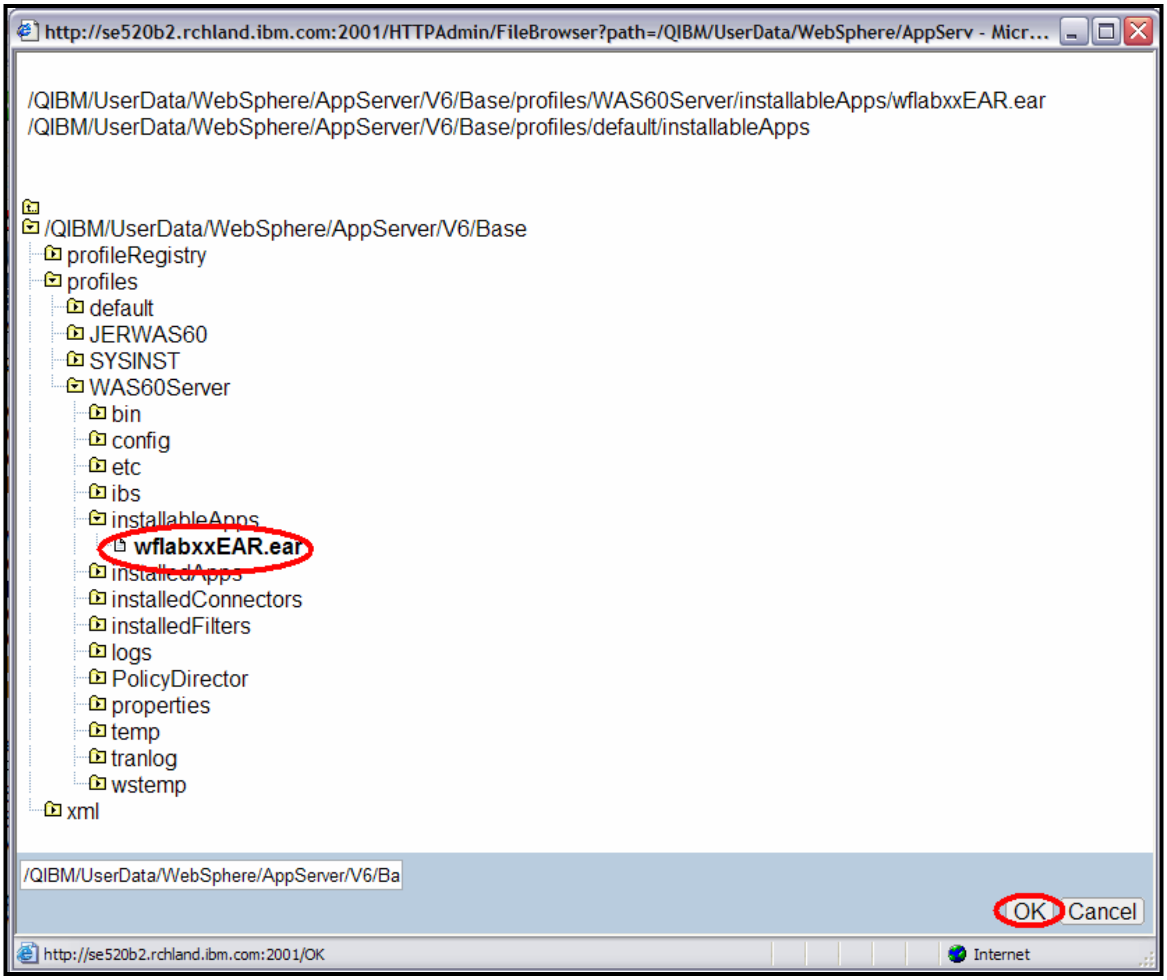
Note: The file must already be on the i5/OS server.

○ Application is contained in a **WAR** file

*Figure 205. The **Install New Application – Specify Application Location** panel*

9. Click the **Application is contained in an EAR file** radio button.
10. Specify the name and path location of your EAR file (the file that you exported in the previous step). Be sure you include the name of the EAR file in this input field.

You can use the **Browse** button to locate the file. The **file browser** panel is displayed. You might need to click the **Up Directory** icon 📁 to navigate to the correct directory (see Figure 206).



http://se520b2.rchland.ibm.com:2001/HTTPAdmin/FileBrowser?path=/QIBM/UserData/WebSphere/AppServ - Micr...

/QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/WAS60Server/installableApps/wflabxxEAR.ear
/QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/default/installableApps

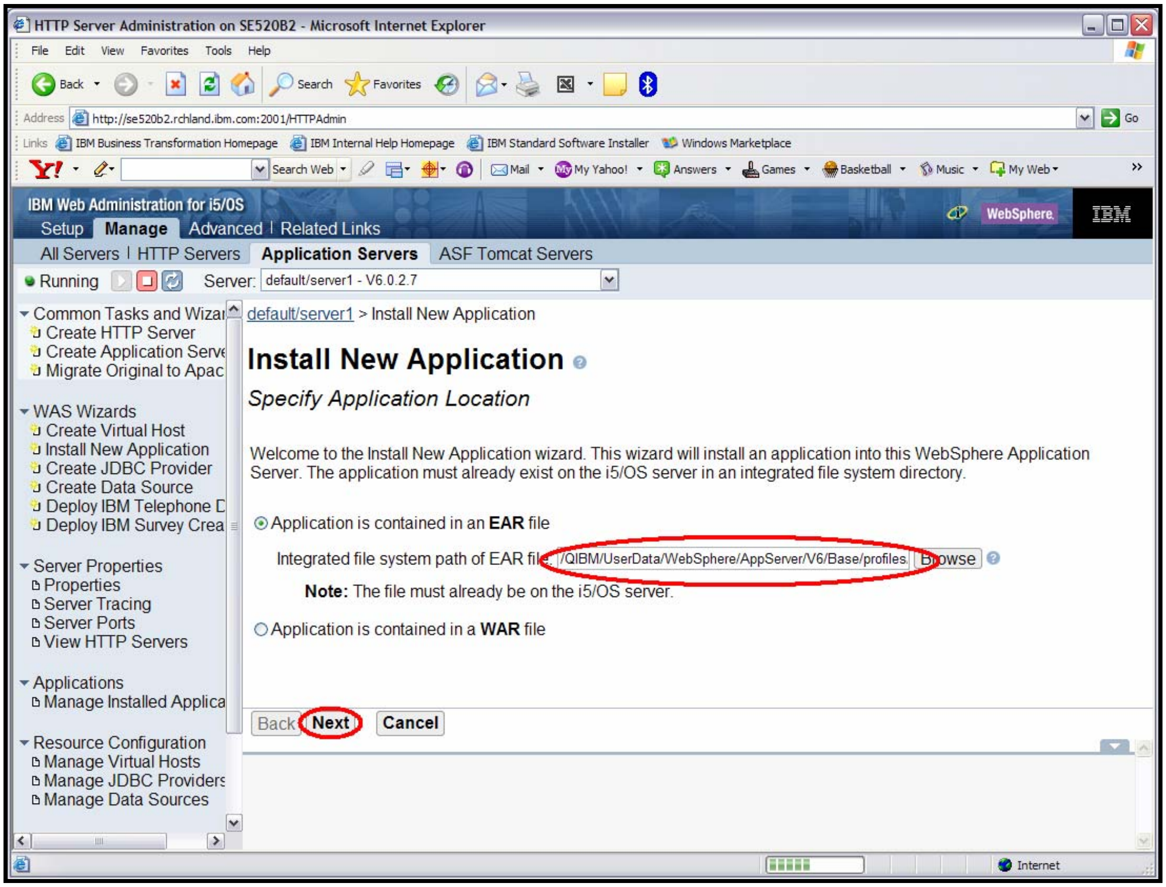
```
/QIBM/UserData/WebSphere/AppServer/V6/Base
  profileRegistry
  profiles
    default
    JERWAS60
    SYSINST
    WAS60Server
      bin
      config
      etc
      ibs
      installableApps
        wflabxxEAR.ear
      installedApps
      installedConnectors
      installedFilters
      logs
      PolicyDirector
      properties
      temp
      tranlog
      wstemp
  xml
```

/QIBM/UserData/WebSphere/AppServer/V6/Ba

*Figure 206. Specify your **EAR** file*

In our example, the file used for this field is: /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/WAS60Server/installableApps /wflabxxEAR.ear.

11. Click **OK**. The Install New Application – Specify Application Location panel is again displayed with the correct directory path and EAR file name (see Figure 207).


HTTP Server Administration on SE520B2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://se520b2.rchland.ibm.com:2001/HTTPAdmin
Go
Links IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer Windows Marketplace
Search Web Mail My Yahoo! Answers Games Basketball Music My Web
IBM Web Administration for i5/OS
WebSphere
IBM
Setup Manage Advanced Related Links
All Servers HTTP Servers Application Servers ASF Tomcat Servers
Running
Common Tasks and Wizar
Create HTTP Server
Create Application Serve
Migrate Original to Apac
WAS Wizards
Create Virtual Host
Install New Application
Create JDBC Provider
Create Data Source
Deploy IBM Telephone D
Deploy IBM Survey Crea
Server Properties
Properties
Server Tracing
Server Ports
View HTTP Servers
Applications
Manage Installed Applica
Resource Configuration
Manage Virtual Hosts
Manage JDBC Providers
Manage Data Sources
Internet

Server: default/server1 - V6.0.2.7

default/server1 > Install New Application

Install New Application

Specify Application Location

Welcome to the Install New Application wizard. This wizard will install an application into this WebSphere Application Server. The application must already exist on the i5/OS server in an integrated file system directory.

◉ Application is contained in an **EAR** file

Integrated file system path of EAR file: /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles Browse

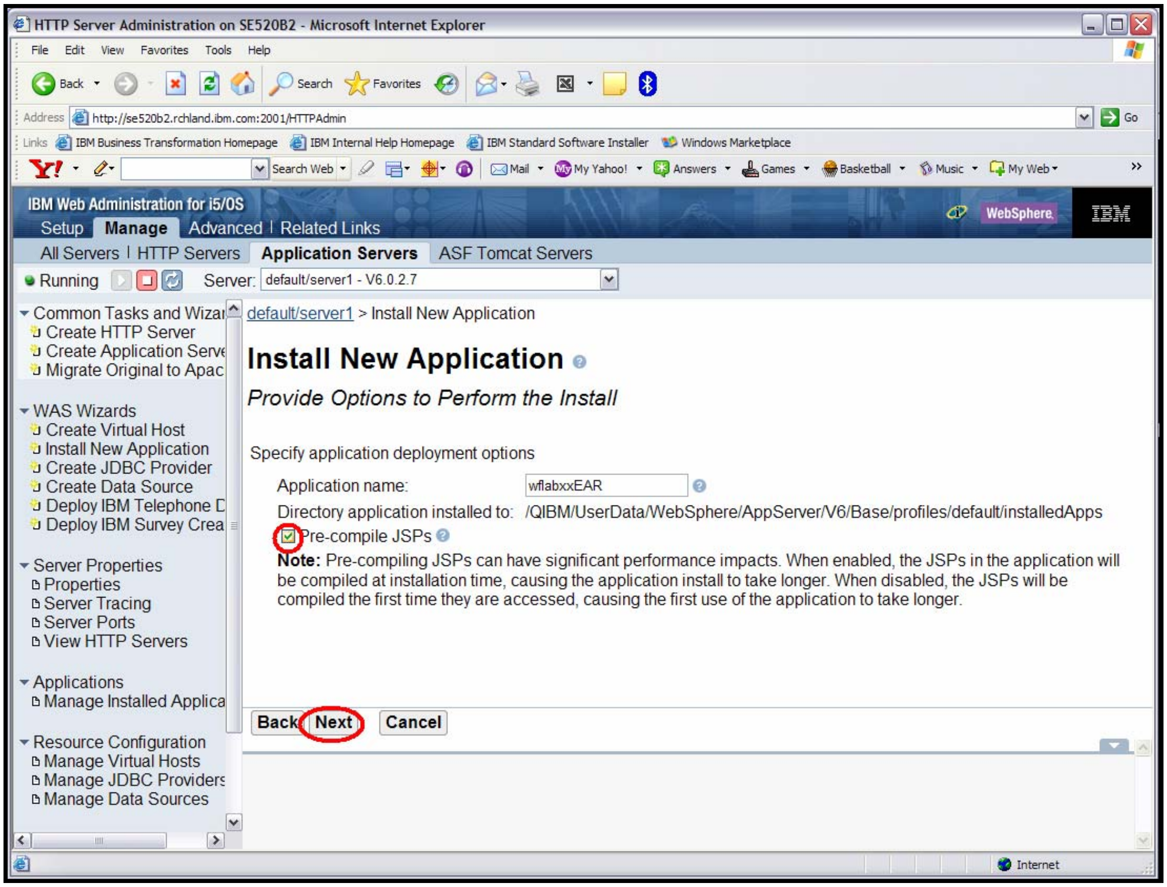
Note: The file must already be on the i5/OS server.

○ Application is contained in a **WAR** file

Back **Next** **Cancel**

*Figure 207. Correct directory path and **EAR** file displayed*

12. Click **Next**. The Install New Application – Provide Options to Perform the Install panel is displayed (see Figure 208).



Install New Application

Provide Options to Perform the Install

Specify application deployment options

Application name: wflabxxEAR

Directory application installed to: /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/default/installedApps

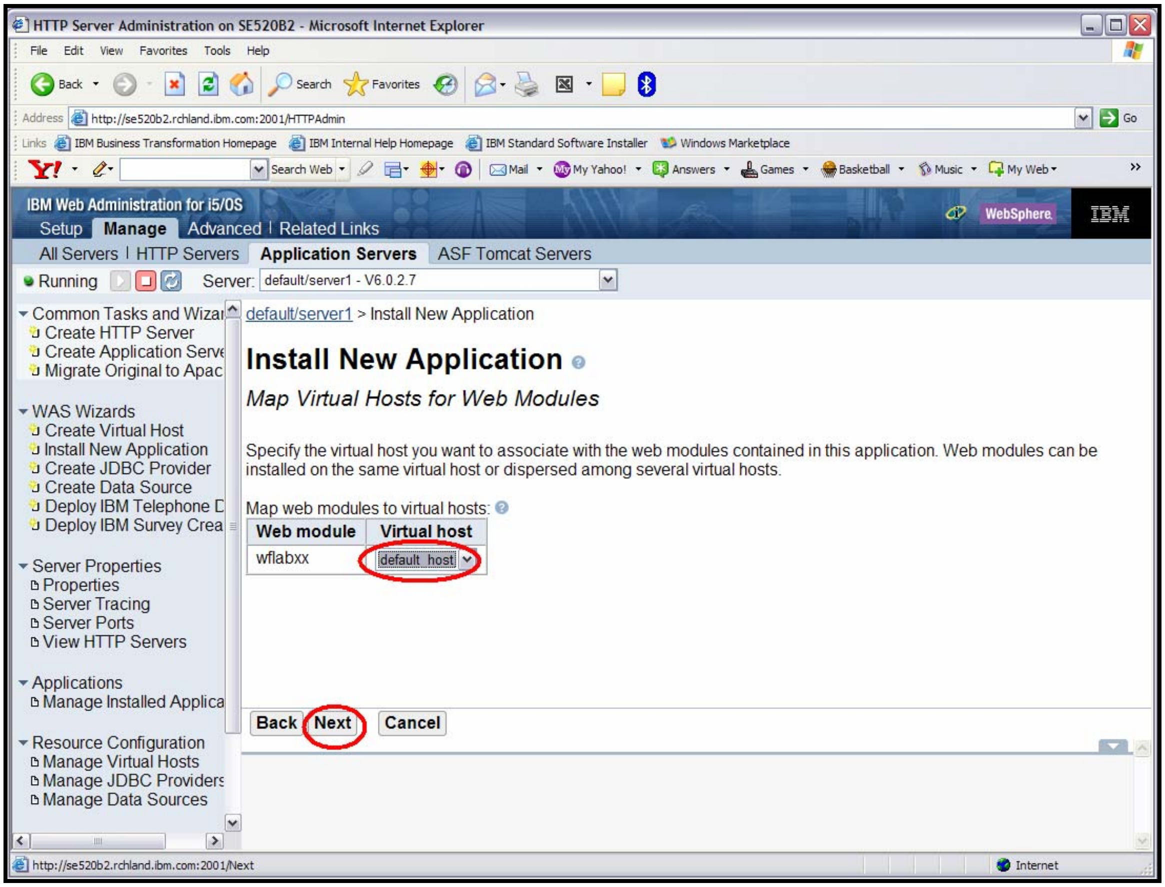
☑ Pre-compile JSPs

Note: Pre-compiling JSPs can have significant performance impacts. When enabled, the JSPs in the application will be compiled at installation time, causing the application install to take longer. When disabled, the JSPs will be compiled the first time they are accessed, causing the first use of the application to take longer.

Figure 208. The Install New Application – Provide Options to Perform the Install panel

13. In the **Application name** field, type `wflabxxEAR` if it not already displayed.
14. You can select the **Pre-compile JSPs** check box to speed up the process of showing the page the first time. (JSPs are compiled into servlets when first generated.)

15. Click **Next**. The Install New Application – Map Virtual Hosts for Web Modules panel is displayed (see Figure 209).



default/server1 > Install New Application

Install New Application

Map Virtual Hosts for Web Modules

Specify the virtual host you want to associate with the web modules contained in this application. Web modules can be installed on the same virtual host or dispersed among several virtual hosts.

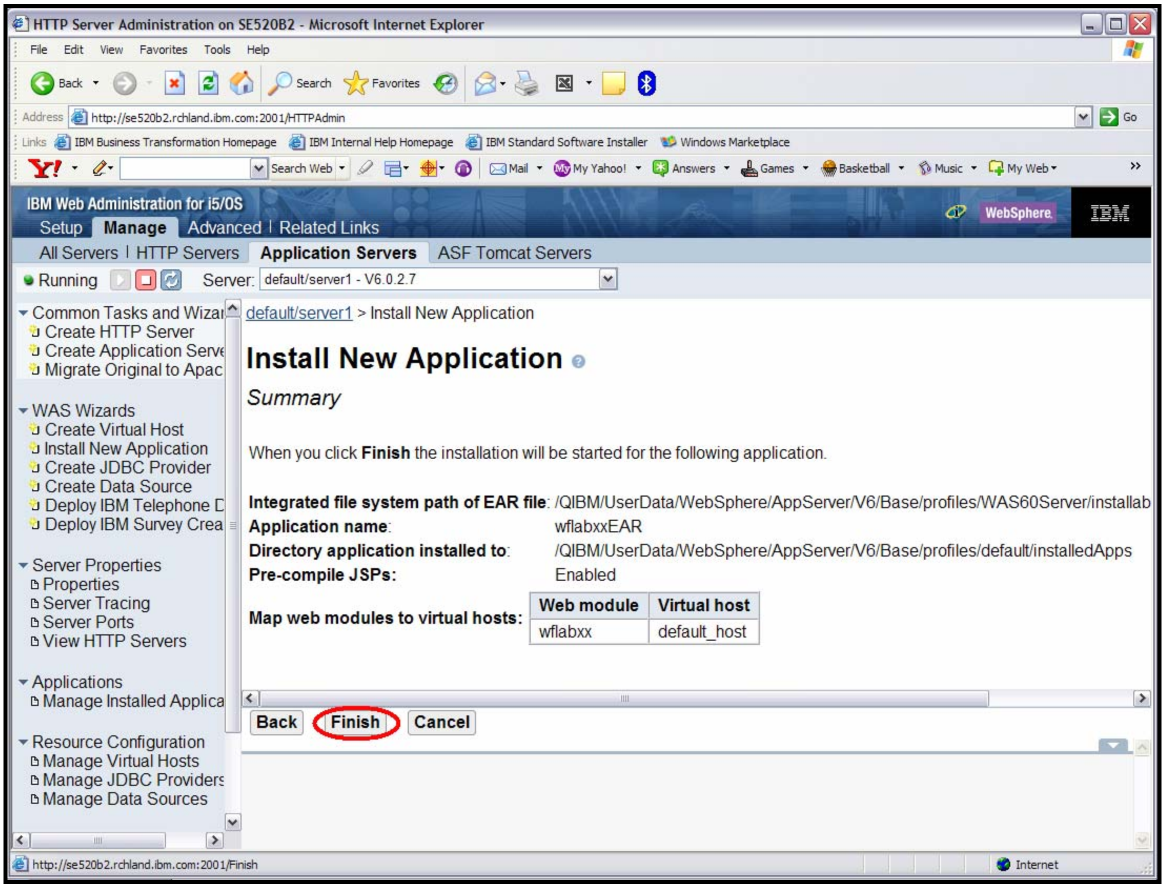
Map web modules to virtual hosts:

Web module	Virtual host
wflabxx	default_host

Figure 209. The Install New Application – Map Virtual Hosts for Web Modules panel

16. Leave the **Virtual host** field set to **default host**.

17. Click **Next**. The Install New Application – Summary panel is displayed (see Figure 210).



HTTP Server Administration on SE520B2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://se520b2.rchland.ibm.com:2001/HTTPAdmin
IBM Web Administration for i5/OS
Setup Manage Advanced Related Links
All Servers HTTP Servers Application Servers ASF Tomcat Servers
Running
Server: default/server1 - V6.0.2.7
Common Tasks and Wizards
Create HTTP Server
Create Application Server
Migrate Original to Apache
WAS Wizards
Create Virtual Host
Install New Application
Create JDBC Provider
Create Data Source
Deploy IBM Telephone D
Deploy IBM Survey Crea
Server Properties
Properties
Server Tracing
Server Ports
View HTTP Servers
Applications
Manage Installed Applica
Resource Configuration
Manage Virtual Hosts
Manage JDBC Providers
Manage Data Sources
default/server1 > Install New Application
Back Finish Cancel
http://se520b2.rchland.ibm.com:2001/Finish
Internet

Install New Application

Summary

When you click **Finish** the installation will be started for the following application.

Integrated file system path of EAR file: /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/WAS60Server/installab
Application name: wflabxxEAR
Directory application installed to: /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/default/installedApps
Pre-compile JSPs: Enabled

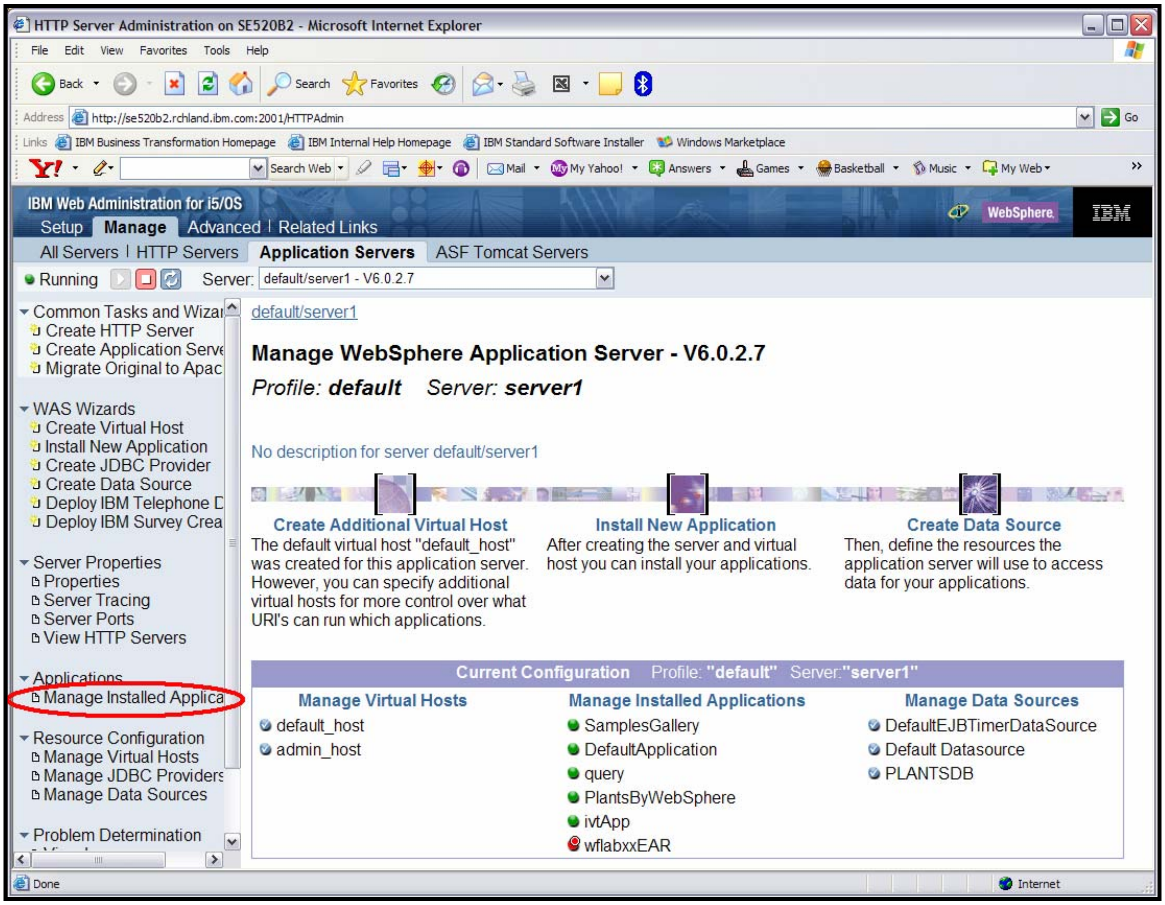
Map web modules to virtual hosts:

Web module	Virtual host
wflabxx	default_host

Figure 210. The Install New Application – Summary panel

18. Click **Finish**.

After the successful installation, you again see the page where you can manage the installed applications (see Figure 211).



HTTP Server Administration on SE520B2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://se520b2.rchland.ibm.com:2001/HTTPAdmin Go
Links IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer Windows Marketplace
Y! Search Web Mail My Yahoo! Answers Games Basketball Music My Web
IBM Web Administration for i5/OS
Setup Manage Advanced Related Links
WebSphere IBM
All Servers HTTP Servers Application Servers ASF Tomcat Servers
Running
Server: default/server1 - V6.0.2.7
Common Tasks and Wizar
Create HTTP Server
Create Application Serve
Migrate Original to Apac
WAS Wizards
Create Virtual Host
Install New Application
Create JDBC Provider
Create Data Source
Deploy IBM Telephone D
Deploy IBM Survey Crea
Server Properties
Properties
Server Tracing
Server Ports
View HTTP Servers
Applications
Manage Installed Applica
Resource Configuration
Manage Virtual Hosts
Manage JDBC Providers
Manage Data Sources
Problem Determination
Done
Internet

default/server1

Manage WebSphere Application Server - V6.0.2.7

Profile: ***default*** *Server:* ***server1***

No description for server default/server1

Create Additional Virtual Host
The default virtual host "default_host" was created for this application server. However, you can specify additional virtual hosts for more control over what URI's can run which applications.

Install New Application
After creating the server and virtual host you can install your applications.

Create Data Source
Then, define the resources the application server will use to access data for your applications.

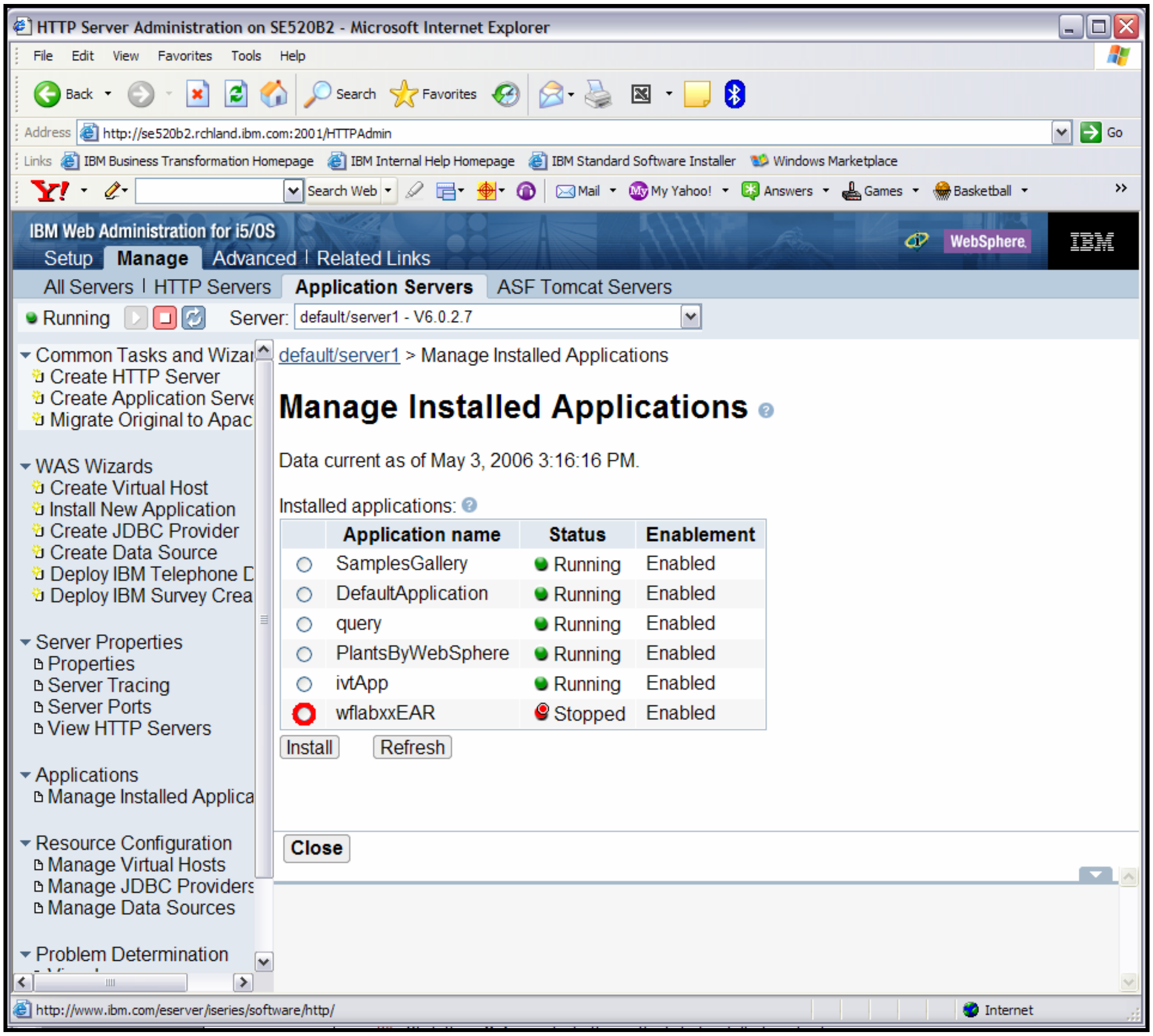
Current Configuration Profile: **"default"** Server:**"server1"**

Manage Virtual Hosts	Manage Installed Applications	Manage Data Sources
default_host	● SamplesGallery	DefaultEJBTimerDataSource
admin_host	● DefaultApplication	Default Datasource
	● query	PLANTSDB
	● PlantsByWebSphere	
	● ivtApp	
	● wflabxxEAR	

Figure 211. Successful installation

19. Click **Manage Installed Applications** under **Applications**.

    The **Manage Installed Applications** panel displays a list of installed applications (see Figure 212).



HTTP Server Administration on SE520B2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://se520b2.rchland.ibm.com:2001/HTTPAdmin Go
Links IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer Windows Marketplace
Search Web Mail My Yahoo! Answers Games Basketball
IBM Web Administration for i5/OS
Setup Manage Advanced Related Links
WebSphere IBM
All Servers HTTP Servers Application Servers ASF Tomcat Servers
Running Server: default/server1 - V6.0.2.7
Common Tasks and Wizar
Create HTTP Server
Create Application Serve
Migrate Original to Apac
WAS Wizards
Create Virtual Host
Install New Application
Create JDBC Provider
Create Data Source
Deploy IBM Telephone D
Deploy IBM Survey Crea
Server Properties
Properties
Server Tracing
Server Ports
View HTTP Servers
Applications
Manage Installed Applica
Resource Configuration
Manage Virtual Hosts
Manage JDBC Providers
Manage Data Sources
Problem Determination
http://www.ibm.com/eserver/iseries/software/http/
Internet

default/server1 > Manage Installed Applications

Manage Installed Applications

Data current as of May 3, 2006 3:16:16 PM.

Installed applications:

	Application name	Status	Enablement
○	SamplesGallery	● Running	Enabled
○	DefaultApplication	● Running	Enabled
○	query	● Running	Enabled
○	PlantsByWebSphere	● Running	Enabled
○	ivtApp	● Running	Enabled
●	wflabxxEAR	● Stopped	Enabled

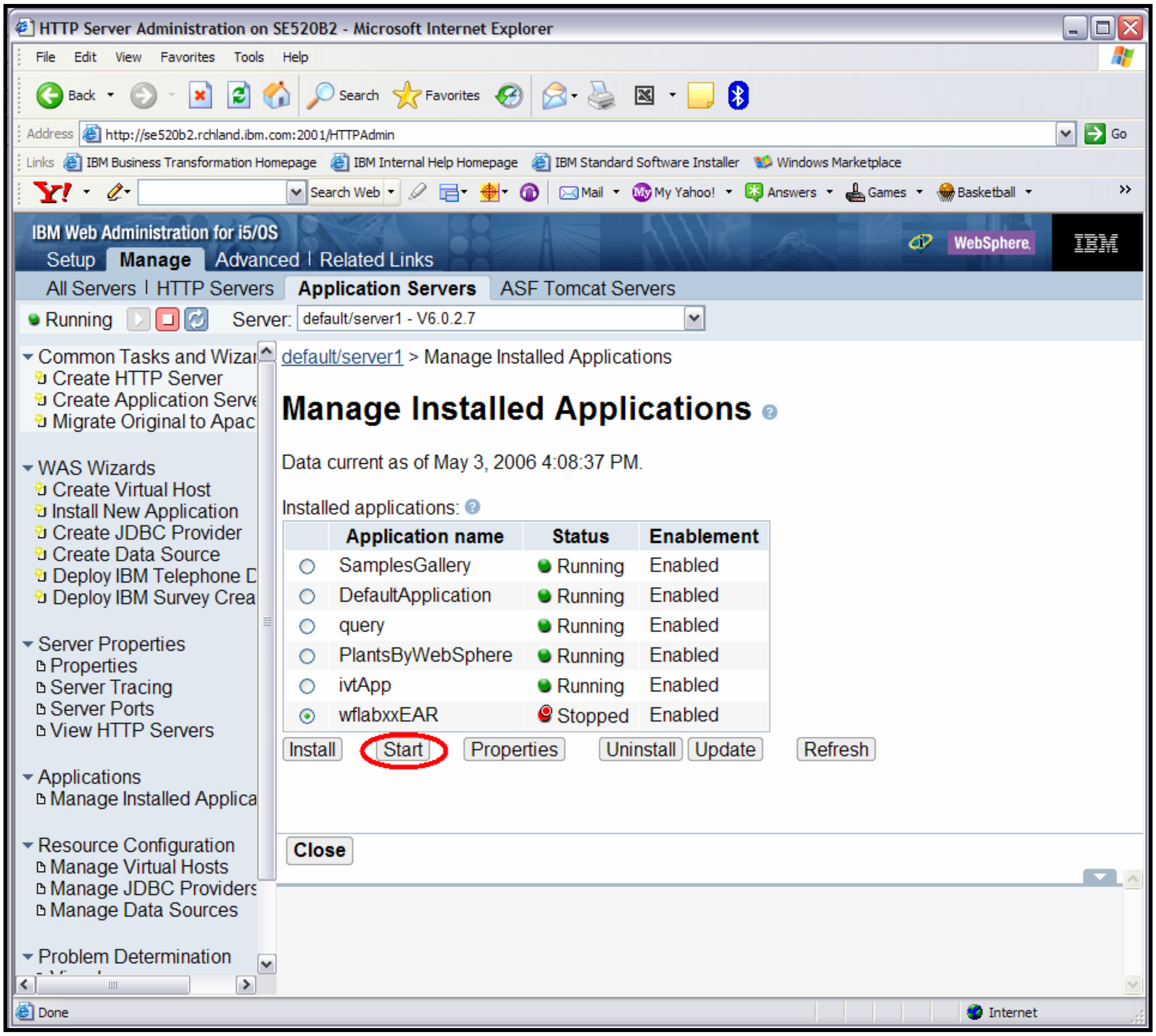
Install Refresh

Close

Figure 212. The Manage Installed Applications panel

20. Click the **wflabxx** radio button in the list of installed applications. The screen refreshes and additional button options are shown.

    Start the application with wflabxx in the enabled state (see Figure 213).


HTTP Server Administration on SE520B2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://se520b2.rchland.ibm.com:2001/HTTPAdmin
Go
Links IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer Windows Marketplace
Search Web Mail My Yahoo! Answers Games Basketball
IBM Web Administration for i5/OS
Setup Manage Advanced Related Links
WebSphere
IBM
All Servers HTTP Servers Application Servers ASF Tomcat Servers
Running
Server: default/server1 - V6.0.2.7
Common Tasks and Wizar
Create HTTP Server
Create Application Serve
Migrate Original to Apac
WAS Wizards
Create Virtual Host
Install New Application
Create JDBC Provider
Create Data Source
Deploy IBM Telephone D
Deploy IBM Survey Crea
Server Properties
Properties
Server Tracing
Server Ports
View HTTP Servers
Applications
Manage Installed Applica
Resource Configuration
Manage Virtual Hosts
Manage JDBC Providers
Manage Data Sources
Problem Determination
default/server1 > Manage Installed Applications
Install Start Properties Uninstall Update Refresh
Close
Done
Internet

Manage Installed Applications

Data current as of May 3, 2006 4:08:37 PM.

Installed applications:

	Application name	Status	Enablement
○	SamplesGallery	● Running	Enabled
○	DefaultApplication	● Running	Enabled
○	query	● Running	Enabled
○	PlantsByWebSphere	● Running	Enabled
○	ivtApp	● Running	Enabled
●	wflabxxEAR	● Stopped	Enabled

Figure 213. Start the application with wflabxx in the enabled state.

21. Click **Start**.

    Now you are ready to test your WebFacing application.

Exercise 12.4: Testing the WebFacing application

To test the Web application:

1. Open Internet Explorer and type `http://servername:port#b/wflabxx` in the **Address** field, where **servername** is the name of your server, and **port#b** is the port number identified earlier.

   Your application index.jsp page opens. If it does not, double-check the URL and the port number (see Figure 214).

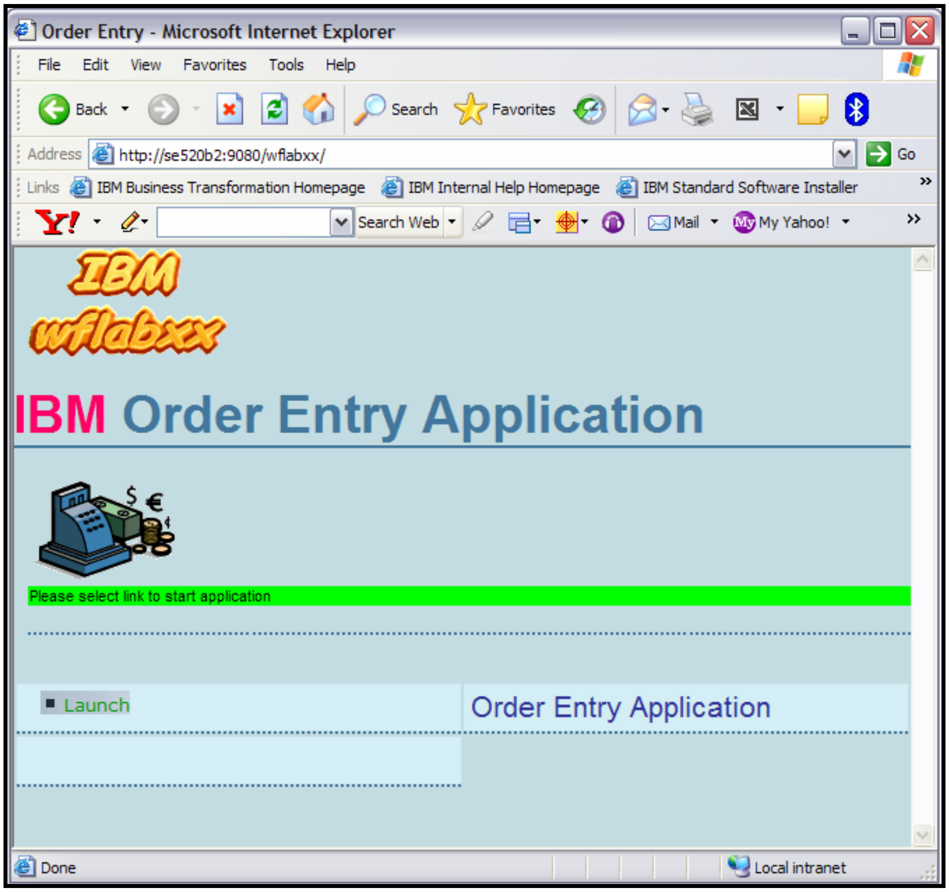

Order Entry - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://se520b2:9080/wflabxx/ Go
Links IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer
Search Web Mail My Yahoo!
IBM wflabxx
IBM Order Entry Application
Please select link to start application
Launch
Order Entry Application
Done
Local intranet

Figure 214. Your application index.jsp page opens.

Recap

You have completed “Exporting to WebSphere Application Server Express for iSeries V6.0.2.7.” You now have the information to understand how to:

- Export files created by the IBM WebFacing Tool to a remote WebSphere Application Server.
- Publish and deploy your application in a real production environment.

IBM WebFacing Tool deployment options

The goal of this section is for you to learn and understand different options available to you in the deployment of your IBM WebFacing Tool application. In customer situations, these options become valuable for performance, uptime, and other business reasons.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 13.1: Learn how to accomplish Partial Deployment of an updated IBM WebFacing Tool application
- Exercise 13.2: Understand JSP Pre-Touch and JSP Compilation
- Exercise 13.3: Configuring the Pre-Touch Attributes

Length of time

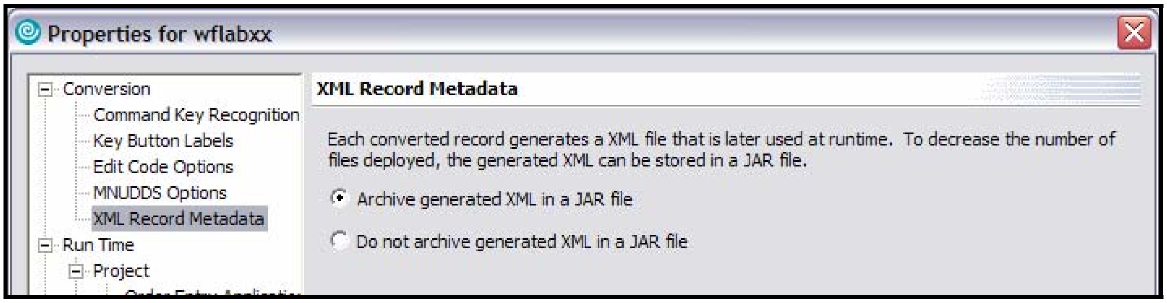
This chapter takes approximately 45 minutes to complete.

Exercise 13.1 Partial deployment

There might be occasions where after successfully deploying an IBM WebFacing Tool application to a customer site, it might be necessary to update the deployed WebFaced application. In a business setting, most etnerprises do not want to uninstall an entire WebFaced application and reinstall a completely new application.

It is possible to deploy only the parts of an IBM WebFacing Tool application that have changed. This can mean deploying any DDS sources, the reconverted JSP files, and the associated XML files to their proper location within the integrated file system or IFS.

1. Recall that the IBM WebFacing Tool generates two JSP files and an XML file for every DDS source file record. By default, all the XML files generated are placed in a single .jar file to decrease the number of generated files for deployment. The file name where all of these XML files are stored is called DDSGeneratedData.jar (see Figure 215).



XML Record Metadata

Each converted record generates a XML file that is later used at runtime. To decrease the number of files deployed, the generated XML can be stored in a JAR file.

● Archive generated XML in a JAR file

○ Do not archive generated XML in a JAR file

Figure 215. Partial deployment of parts of an IBM WebFacing Tool application

A screen shot of an expanded wflabxx Application in the WebFacing Perspective is shown in Figure 216.

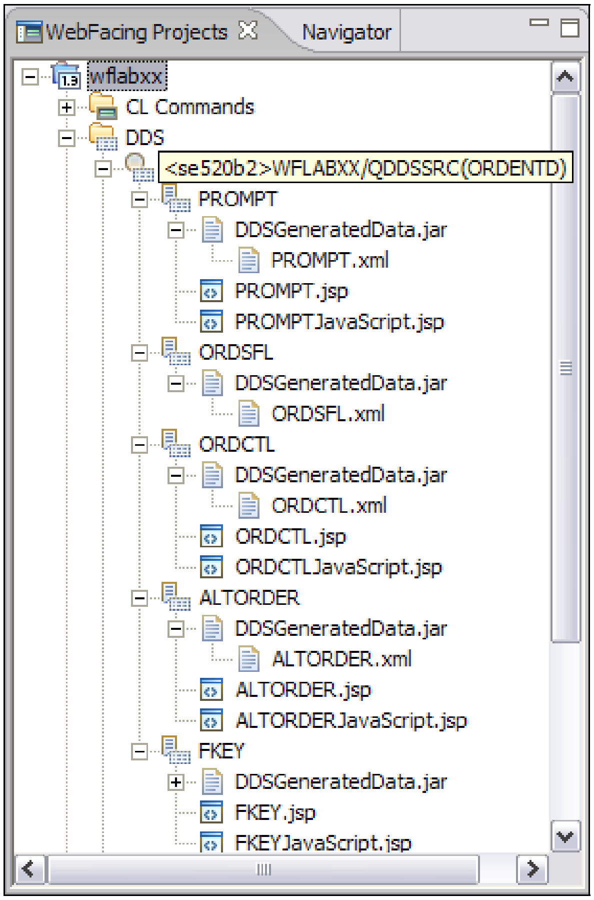
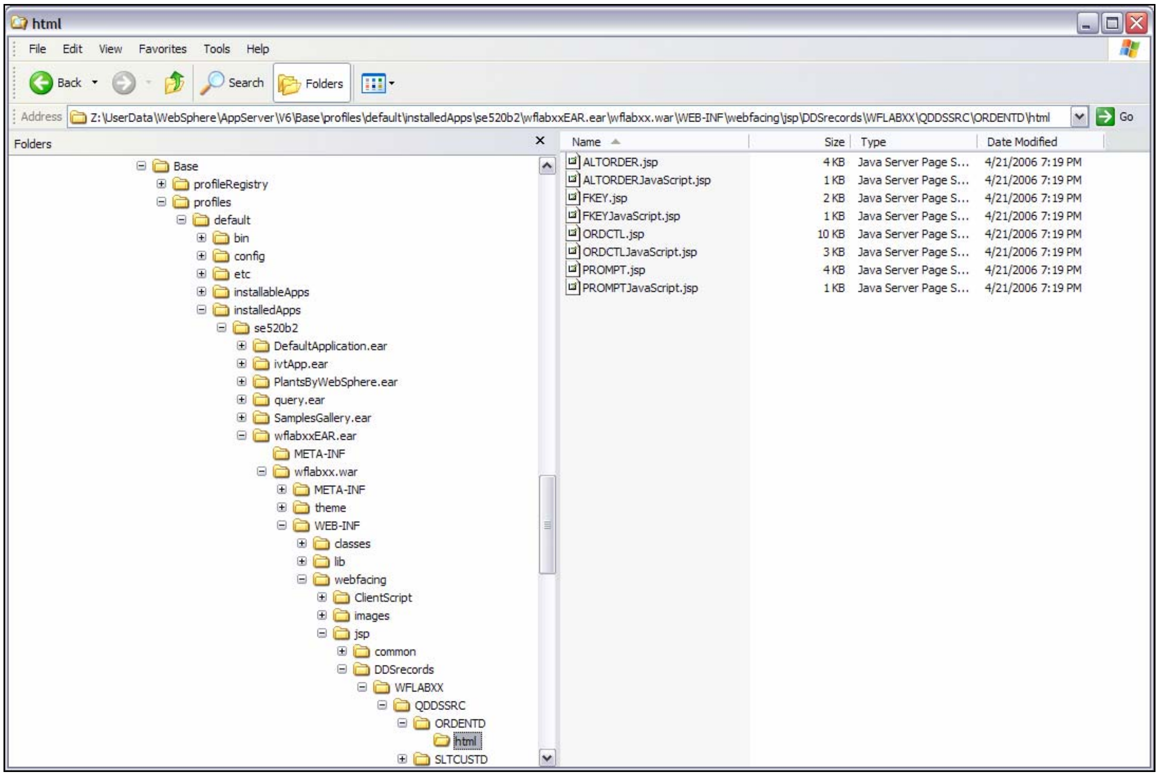


Figure 216. Screen shot of an expanded wflabxx Application in the WebFacing Perspective

2. Notice that for the ORDENTD DDS source for the Prompt record there is a Prompt.JSP, a PromptJavaScript.jsp, and a PROMPT.xml file. This can mean that a change to the PROMPT record of the ORDENTD DDS source file requires deployment of the PROMPT.xml, PROMPT.jsp, and PROMPTJavaScript.JSP files.

3. Looking at the deployed application using Internet Explorer, you can see the Prompt.jsp and PromptJavaScript.jsp files (see Figure 217).



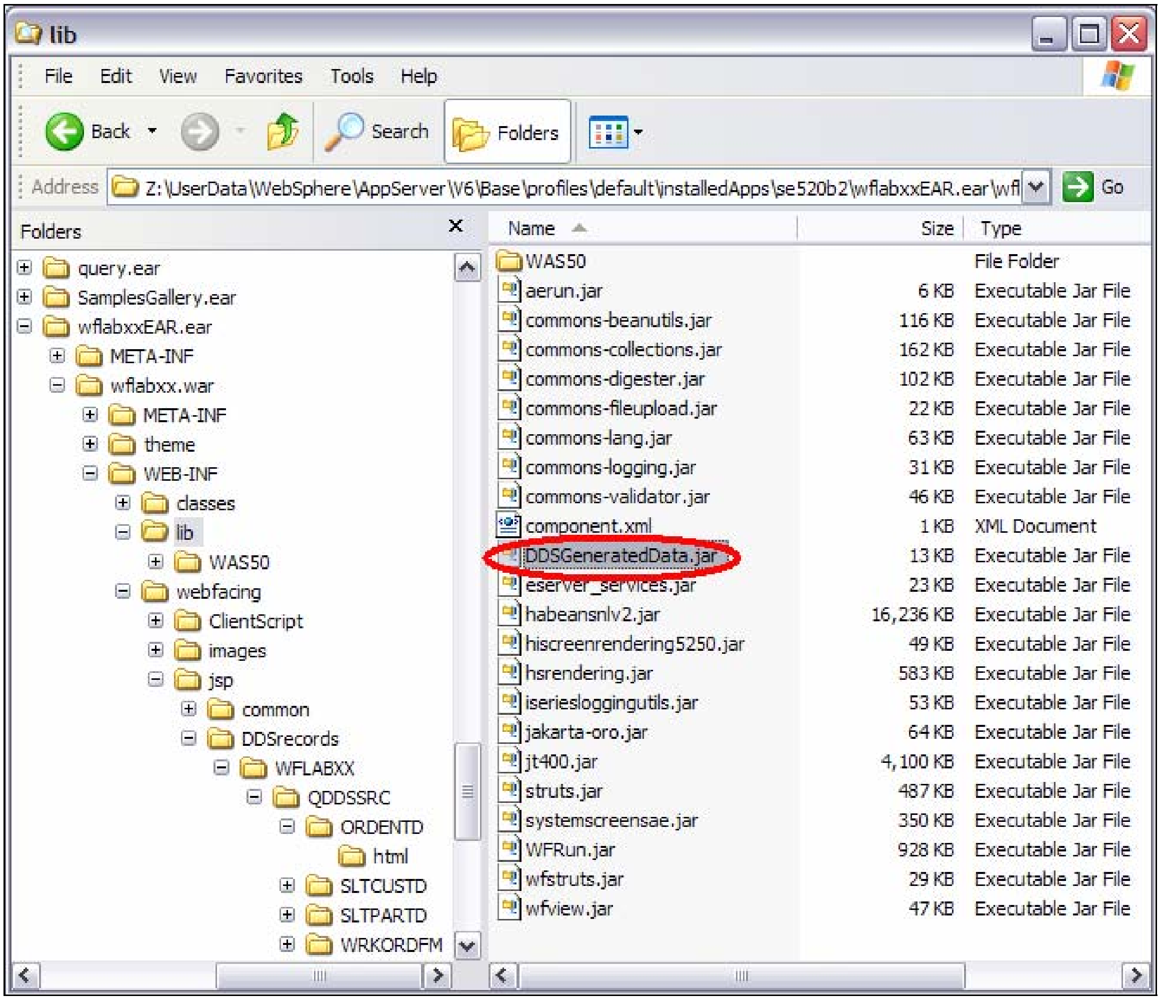
Z:\UserData\WebSphere\AppServer\V6\Base\profiles\default\installedApps\se520b2\wflabxxEAR.ear\wflabxx.war\WEB-INF\webfacing\jsp\DDSrecords\WFLABXX\QDDSSRC\ORDENTD\html

```
⊟ Base
  ⊞ profileRegistry
  ⊟ profiles
    ⊟ default
      ⊞ bin
      ⊞ config
      ⊞ etc
      ⊞ installableApps
      ⊟ installedApps
        ⊟ se520b2
          ⊞ DefaultApplication.ear
          ⊞ ivtApp.ear
          ⊞ PlantsByWebSphere.ear
          ⊞ query.ear
          ⊞ SamplesGallery.ear
          ⊟ wflabxxEAR.ear
              META-INF
            ⊟ wflabxx.war
              ⊞ META-INF
              ⊞ theme
              ⊟ WEB-INF
                ⊞ classes
                ⊞ lib
                ⊟ webfacing
                  ⊞ ClientScript
                  ⊞ images
                  ⊟ jsp
                    ⊞ common
                    ⊟ DDSrecords
                      ⊟ WFLABXX
                        ⊟ QDDSSRC
                          ⊟ ORDENTD
                              html
                          ⊞ SLTCUSTD
```

Name	Size	Type	Date Modified
ALTORDER.jsp	4 KB	Java Server Page S...	4/21/2006 7:19 PM
ALTORDERJavaScript.jsp	1 KB	Java Server Page S...	4/21/2006 7:19 PM
FKEY.jsp	2 KB	Java Server Page S...	4/21/2006 7:19 PM
FKEYJavaScript.jsp	1 KB	Java Server Page S...	4/21/2006 7:19 PM
ORDCTL.jsp	10 KB	Java Server Page S...	4/21/2006 7:19 PM
ORDCTLJavaScript.jsp	3 KB	Java Server Page S...	4/21/2006 7:19 PM
PROMPT.jsp	4 KB	Java Server Page S...	4/21/2006 7:19 PM
PROMPTJavaScript.jsp	1 KB	Java Server Page S...	4/21/2006 7:19 PM

*Figure 217. The **Prompt.jsp** and **PromptJavaScript.jsp** files*

4. The DDSGeneratedData.jar file is located in the WEB-INF/Lib directory (see Figure 218).

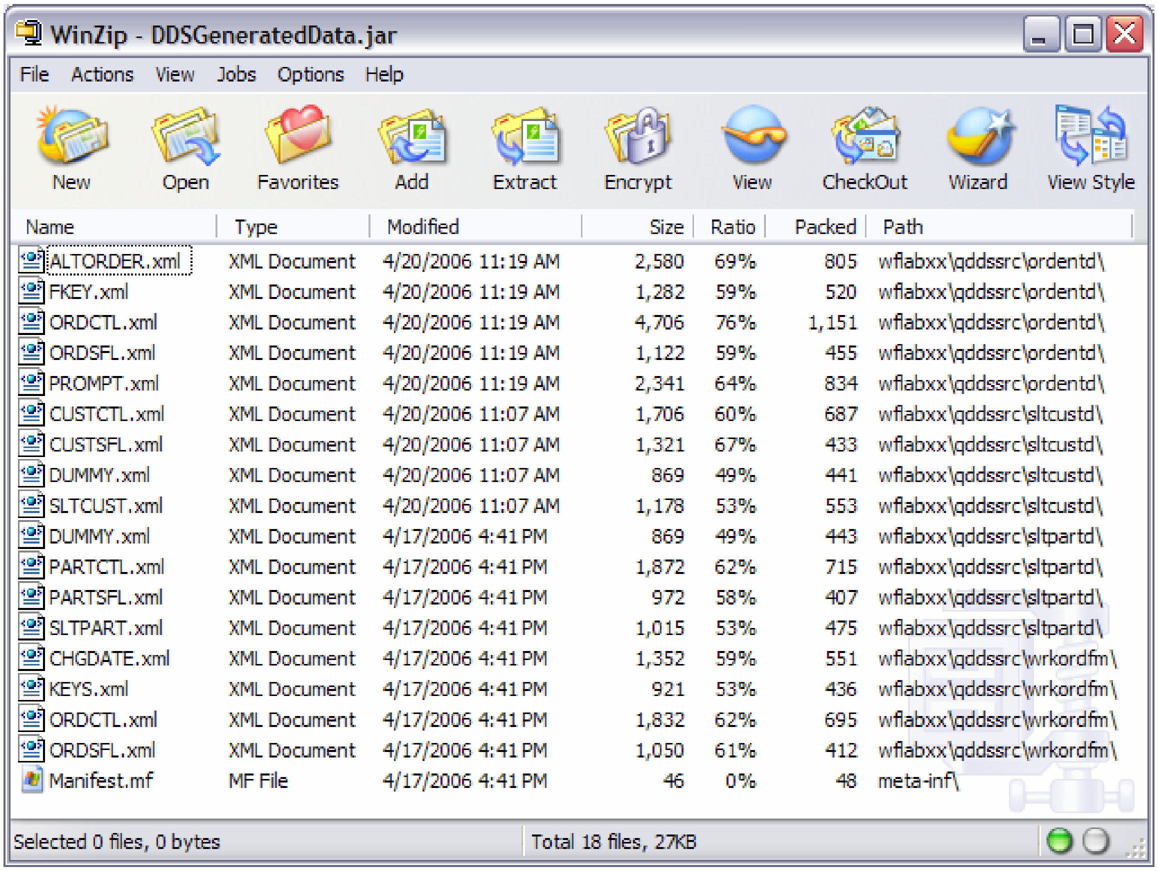


Address Z:\UserData\WebSphere\AppServer\V6\Base\profiles\default\installedApps\se520b2\wflabxxEAR.ear\wfl

Name	Size	Type
WAS50		File Folder
aerun.jar	6 KB	Executable Jar File
commons-beanutils.jar	116 KB	Executable Jar File
commons-collections.jar	162 KB	Executable Jar File
commons-digester.jar	102 KB	Executable Jar File
commons-fileupload.jar	22 KB	Executable Jar File
commons-lang.jar	63 KB	Executable Jar File
commons-logging.jar	31 KB	Executable Jar File
commons-validator.jar	46 KB	Executable Jar File
component.xml	1 KB	XML Document
DDSGeneratedData.jar	13 KB	Executable Jar File
eserver_services.jar	23 KB	Executable Jar File
habeansnlv2.jar	16,236 KB	Executable Jar File
hiscreenrendering5250.jar	49 KB	Executable Jar File
hsrendering.jar	583 KB	Executable Jar File
iseriesloggingutils.jar	53 KB	Executable Jar File
jakarta-oro.jar	64 KB	Executable Jar File
jt400.jar	4,100 KB	Executable Jar File
struts.jar	487 KB	Executable Jar File
systemscreensae.jar	350 KB	Executable Jar File
WFRun.jar	928 KB	Executable Jar File
wfstruts.jar	29 KB	Executable Jar File
wfview.jar	47 KB	Executable Jar File

Figure 218. The ***DDSGeneratedData.jar*** *file is located in the* ***WEB-INF/Lib*** *directory.*

5. Open the File using a .zip utility. Notice that all of the XML files for all of the DDS records are stored in this file. Notice the path associated with each XML file. In the example below in Figure 219, the path to PROMPT.xml is wflabxx\qddssrc\ordentd. This is important to know for the later partial deployment.



Name	Type	Modified	Size	Ratio	Packed	Path
ALTORDER.xml	XML Document	4/20/2006 11:19 AM	2,580	69%	805	wflabxx\qddssrc\ordentd\
FKEY.xml	XML Document	4/20/2006 11:19 AM	1,282	59%	520	wflabxx\qddssrc\ordentd\
ORDCTL.xml	XML Document	4/20/2006 11:19 AM	4,706	76%	1,151	wflabxx\qddssrc\ordentd\
ORDSFL.xml	XML Document	4/20/2006 11:19 AM	1,122	59%	455	wflabxx\qddssrc\ordentd\
PROMPT.xml	XML Document	4/20/2006 11:19 AM	2,341	64%	834	wflabxx\qddssrc\ordentd\
CUSTCTL.xml	XML Document	4/20/2006 11:07 AM	1,706	60%	687	wflabxx\qddssrc\sltcustd\
CUSTSFL.xml	XML Document	4/20/2006 11:07 AM	1,321	67%	433	wflabxx\qddssrc\sltcustd\
DUMMY.xml	XML Document	4/20/2006 11:07 AM	869	49%	441	wflabxx\qddssrc\sltcustd\
SLTCUST.xml	XML Document	4/20/2006 11:07 AM	1,178	53%	553	wflabxx\qddssrc\sltcustd\
DUMMY.xml	XML Document	4/17/2006 4:41 PM	869	49%	443	wflabxx\qddssrc\sltpartd\
PARTCTL.xml	XML Document	4/17/2006 4:41 PM	1,872	62%	715	wflabxx\qddssrc\sltpartd\
PARTSFL.xml	XML Document	4/17/2006 4:41 PM	972	58%	407	wflabxx\qddssrc\sltpartd\
SLTPART.xml	XML Document	4/17/2006 4:41 PM	1,015	53%	475	wflabxx\qddssrc\sltpartd\
CHGDATE.xml	XML Document	4/17/2006 4:41 PM	1,352	59%	551	wflabxx\qddssrc\wrkordfm\
KEYS.xml	XML Document	4/17/2006 4:41 PM	921	53%	436	wflabxx\qddssrc\wrkordfm\
ORDCTL.xml	XML Document	4/17/2006 4:41 PM	1,832	62%	695	wflabxx\qddssrc\wrkordfm\
ORDSFL.xml	XML Document	4/17/2006 4:41 PM	1,050	61%	412	wflabxx\qddssrc\wrkordfm\
Manifest.mf	MF File	4/17/2006 4:41 PM	46	0%	48	meta-inf\

Selected 0 files, 0 bytes

Total 18 files, 27KB

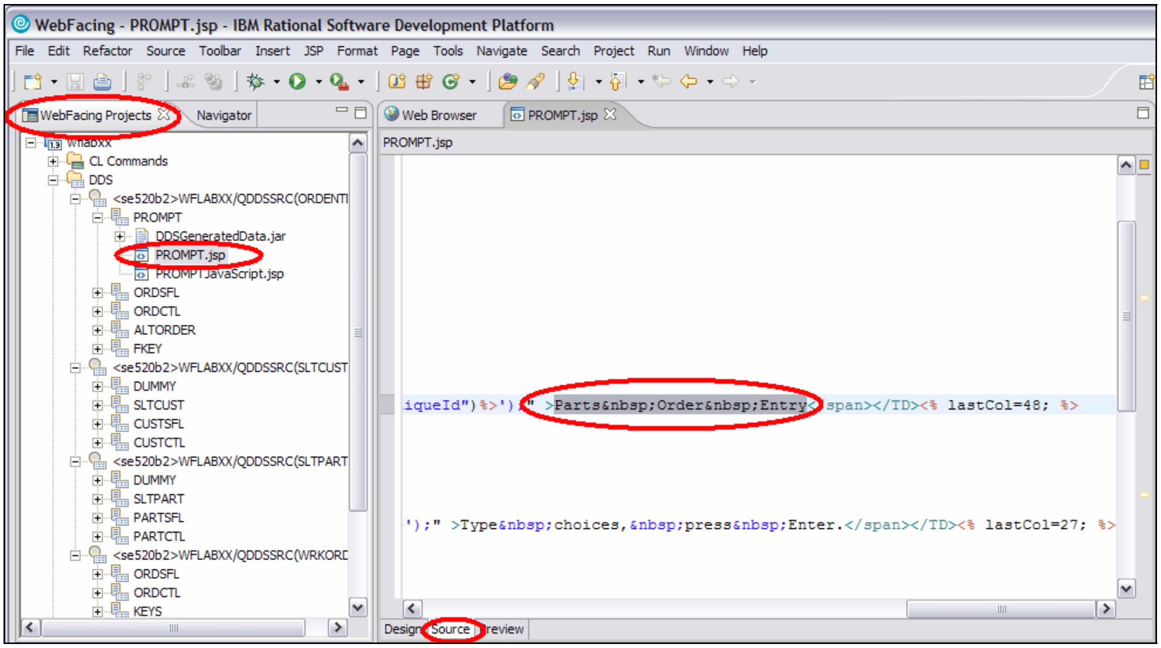
Figure 219. All of the XML files for all of the DDS records are stored in this file.

IBM WebFacing Tool partial deployment

The IBM WebFacing tool does not provided any tools for partially deploying application code changes. Partial deployment is really a manual process. The process might be made easier by developing a tool to perform the steps. For the purposes of this example, you perform these steps manually.

For this example, instead of changing DDS source files and reconverting, you manually make a change to the Prompt JSP file to simulate a DDS Source file change. Remember, when the DDS Source is changed for this file three files need to be copied: PROMPT.jsp, PROMPTJavaScript.jsp, and PROMPT.XML.

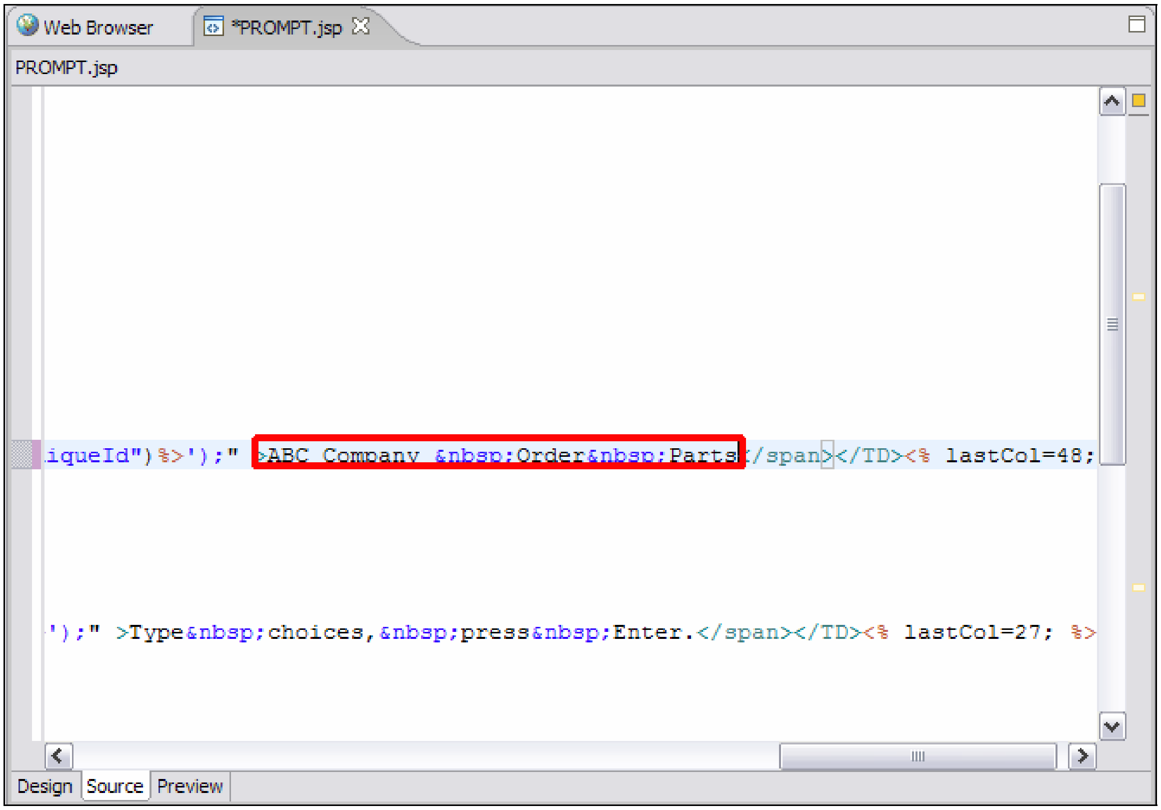
1. Switch to the Webfacing Projects perspective. Double-click the **ORDENT PROMPT.jsp** to open it and switch to the Source view by clicking the **Source** tab. Search for the Parts Order Entry text **Parts&nbsp;Order&nbsp;Entry** as shown in Figure 220.


WebFacing - PROMPT.jsp - IBM Rational Software Development Platform
File Edit Refactor Source Toolbar Insert JSP Format Page Tools Navigate Search Project Run Window Help
WebFacing Projects
Navigator
Web Browser
PROMPT.jsp
PROMPT.jsp
WhaBXX
CL Commands
DDS
<se520b2>WFLABXX/QDDSSRC(ORDENT
PROMPT
DDSGeneratedData.jar
PROMPT.jsp
PROMPTJavaScript.jsp
ORDSFL
ORDCTL
ALTORDER
FKEY
<se520b2>WFLABXX/QDDSSRC(SLTCUST
DUMMY
SLTCUST
CUSTSFL
CUSTCTL
<se520b2>WFLABXX/QDDSSRC(SLTPART
DUMMY
SLTPART
PARTSFL
PARTCTL
<se520b2>WFLABXX/QDDSSRC(WRKORD
ORDSFL
ORDCTL
KEYS
Design
Source
Preview

```
iqueId")%>');" >Parts&nbsp;Order&nbsp;Entry</span></TD><% lastCol=48; %>

');" >Type&nbsp;choices,&nbsp;press&nbsp;Enter.</span></TD><% lastCol=27; %>
```

Figure 220. IBM WebFacing Tool Partial Deployment

2. Change the Parts Order Entry Text to `ABC Company &nbsp;Order&nbsp;Parts` as shown in Figure 221.



```
iqueId")%>');" >ABC Company &nbsp;Order&nbsp;Parts</span></TD><% lastCol=48;

');" >Type&nbsp;choices,&nbsp;press&nbsp;Enter.</span></TD><% lastCol=27; %>
```

Figure 221. Change the Parts Order Entry Text

3. Save your changes and test the changes from within WebSphere Development Studio Client. After launching and signing on to the application, the changes are visible on the first Order Entry Application screen (see Figure 222).

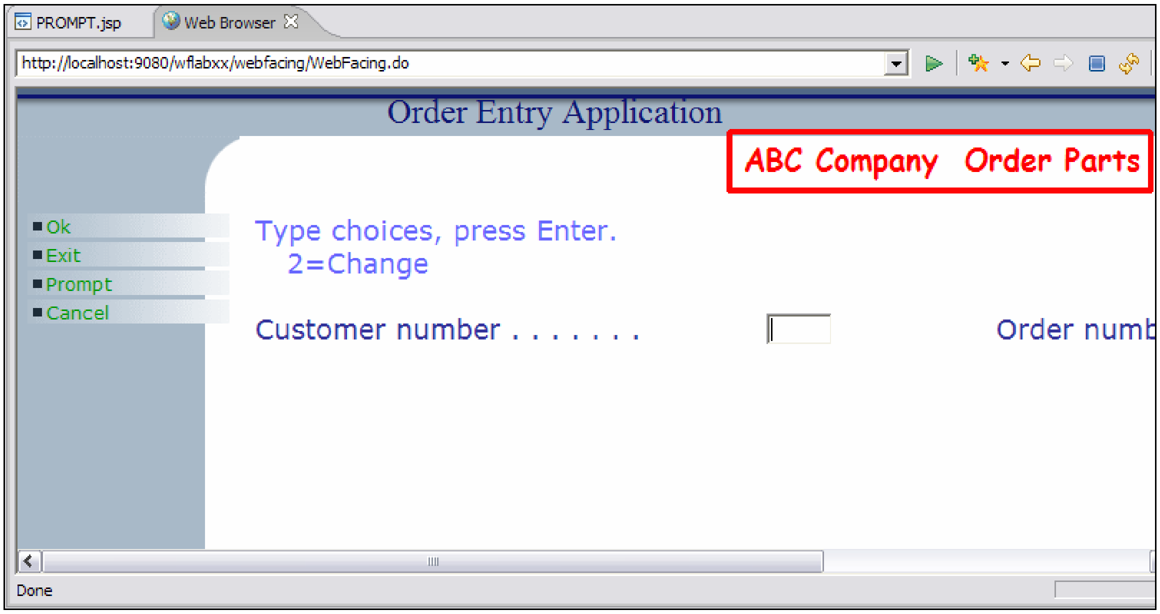
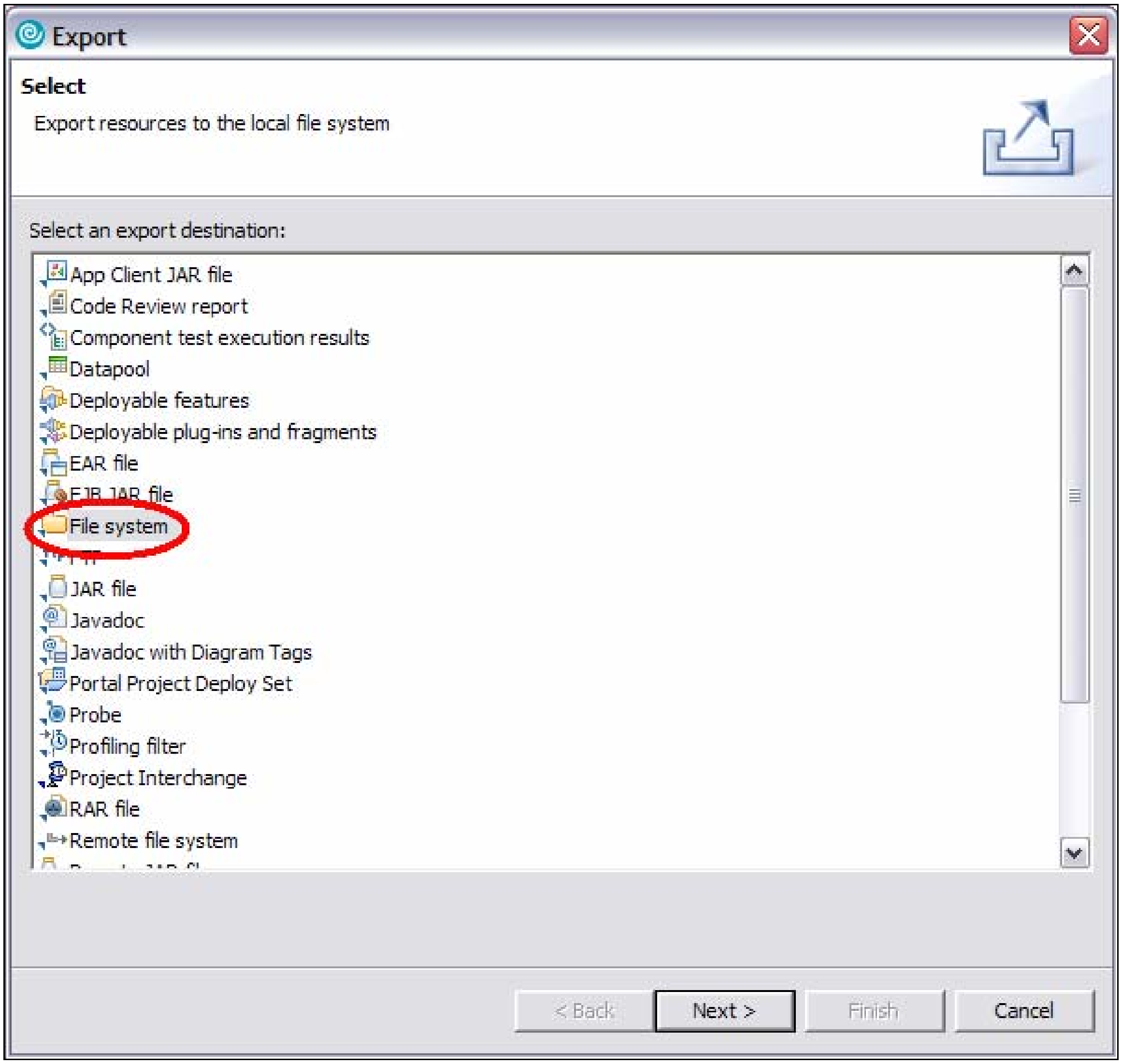

PROMPT.jsp
Web Browser
http://localhost:9080/wflabxx/webfacing/WebFacing.do
Order Entry Application
ABC Company Order Parts
Ok
Exit
Prompt
Cancel
Type choices, press Enter.
2=Change
Customer number . . . .
Order numb
Done

Figure 222. Changes seen on Order Entry Application screen

4. Before you can deploy the sources to the running System i model, you have to get the sources necessary from WebSphere Development Studio Client. The three files to deploy are: PROMPT.jsp, PROMPTJavaScript.jsp, and PROMPT.XML. Because PROMPT.XML in the DDSGenerated.jar file is archived, you need to export this file to the file system to be able to extract PROMPT.XML. These are the three files that might have changed if the DDS source had changed.

5. You need to locate these three files and export them. Use **File > Export** and select **File System** (see Figure 223).



Export

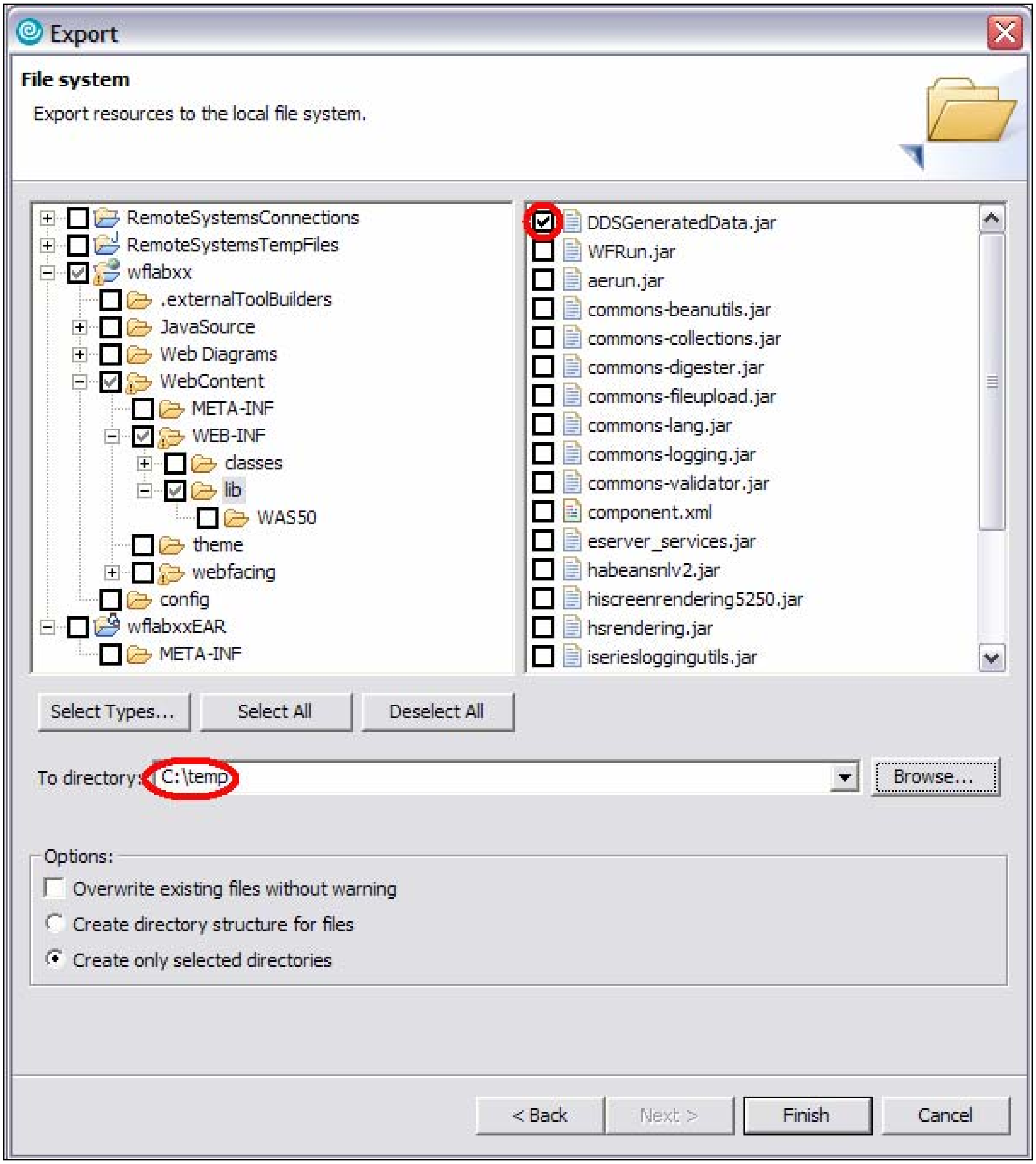
Select

Export resources to the local file system

Select an export destination:

Figure 223. Locate files in File system to export them

6. There are two screen shots to illustrate where these three files are located. Select each of the files to be exported as shown. Select **c:\temp** as the **To directory:** target for the export, and click **Finish** to export the three files to c:\temp (see Figure 224 and Figure 225).



Export

File system

Export resources to the local file system.

- □ RemoteSystemsConnections
- □ RemoteSystemsTempFiles
- ☑ wflabxx
  - □ .externalToolBuilders
  - □ JavaSource
  - □ Web Diagrams
  - ☑ WebContent
    - □ META-INF
    - ☑ WEB-INF
      - □ classes
      - ☑ lib
        - □ WAS50
    - □ theme
    - □ webfacing
  - □ config
- □ wflabxxEAR
  - □ META-INF

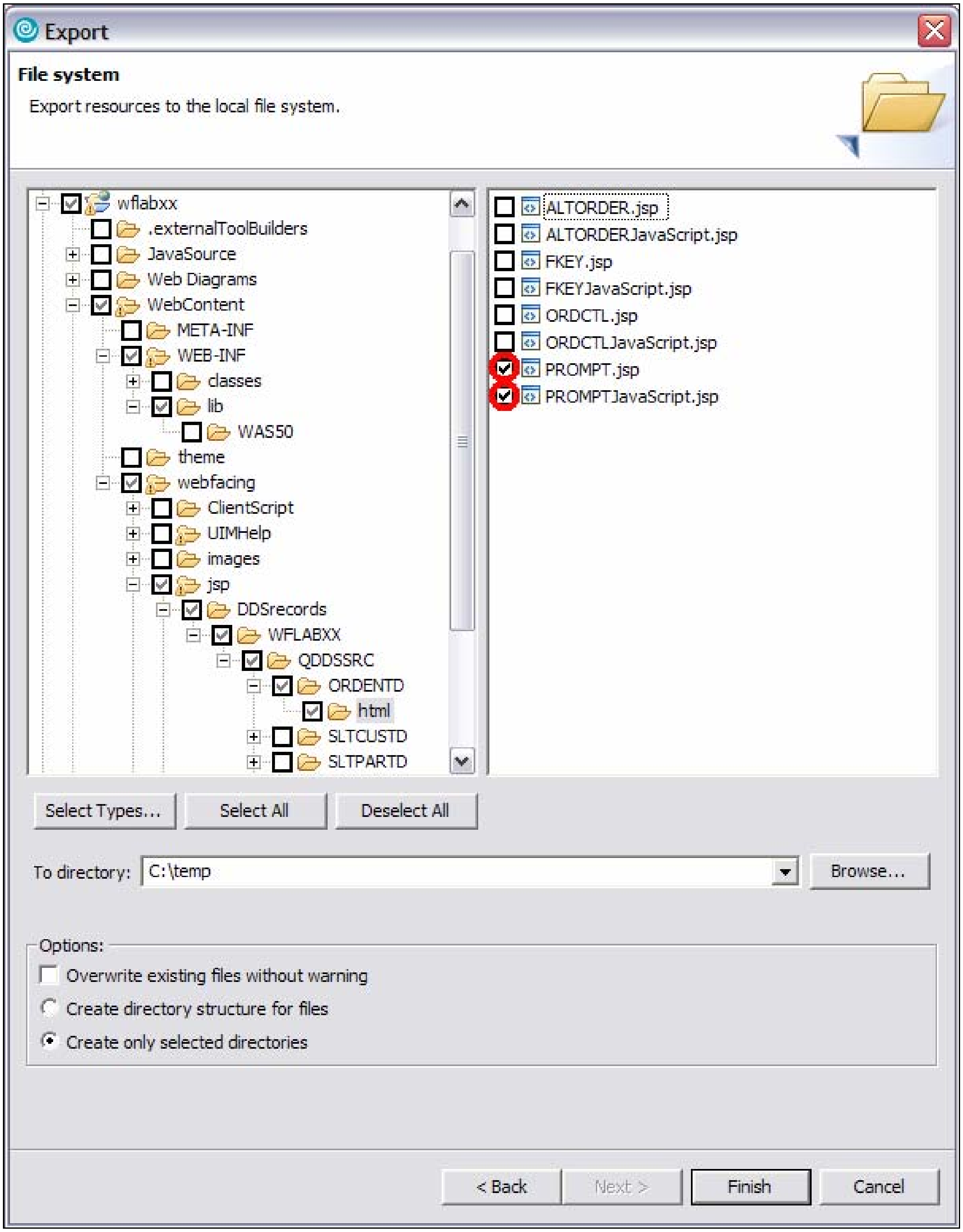
☑ DDSGeneratedData.jar
□ WFRun.jar
□ aerun.jar
□ commons-beanutils.jar
□ commons-collections.jar
□ commons-digester.jar
□ commons-fileupload.jar
□ commons-lang.jar
□ commons-logging.jar
□ commons-validator.jar
□ component.xml
□ eserver_services.jar
□ habeansnlv2.jar
□ hiscreenrendering5250.jar
□ hsrendering.jar
□ iseriesloggingutils.jar

To directory: C:\temp

Options:

□ Overwrite existing files without warning
○ Create directory structure for files
● Create only selected directories

*Figure 224. Locating files in **File system***



Export

File system

Export resources to the local file system.

- ⊟ ☑ wflabxx
  - □ .externalToolBuilders
  - ⊞ □ JavaSource
  - ⊞ □ Web Diagrams
  - ⊟ ☑ WebContent
    - □ META-INF
    - ⊟ ☑ WEB-INF
      - ⊞ □ classes
      - ⊟ ☑ lib
        - □ WAS50
    - □ theme
    - ⊟ ☑ webfacing
      - ⊞ □ ClientScript
      - ⊞ □ UIMHelp
      - ⊞ □ images
      - ⊟ ☑ jsp
        - ⊟ ☑ DDSrecords
          - ⊟ ☑ WFLABXX
            - ⊟ ☑ QDDSSRC
              - ⊟ ☑ ORDENTD
                - ☑ html
              - ⊞ □ SLTCUSTD
              - ⊞ □ SLTPARTD

- □ ALTORDER.jsp
- □ ALTORDERJavaScript.jsp
- □ FKEY.jsp
- □ FKEYJavaScript.jsp
- □ ORDCTL.jsp
- □ ORDCTLJavaScript.jsp
- ☑ PROMPT.jsp
- ☑ PROMPTJavaScript.jsp

To directory: C:\temp

Options:

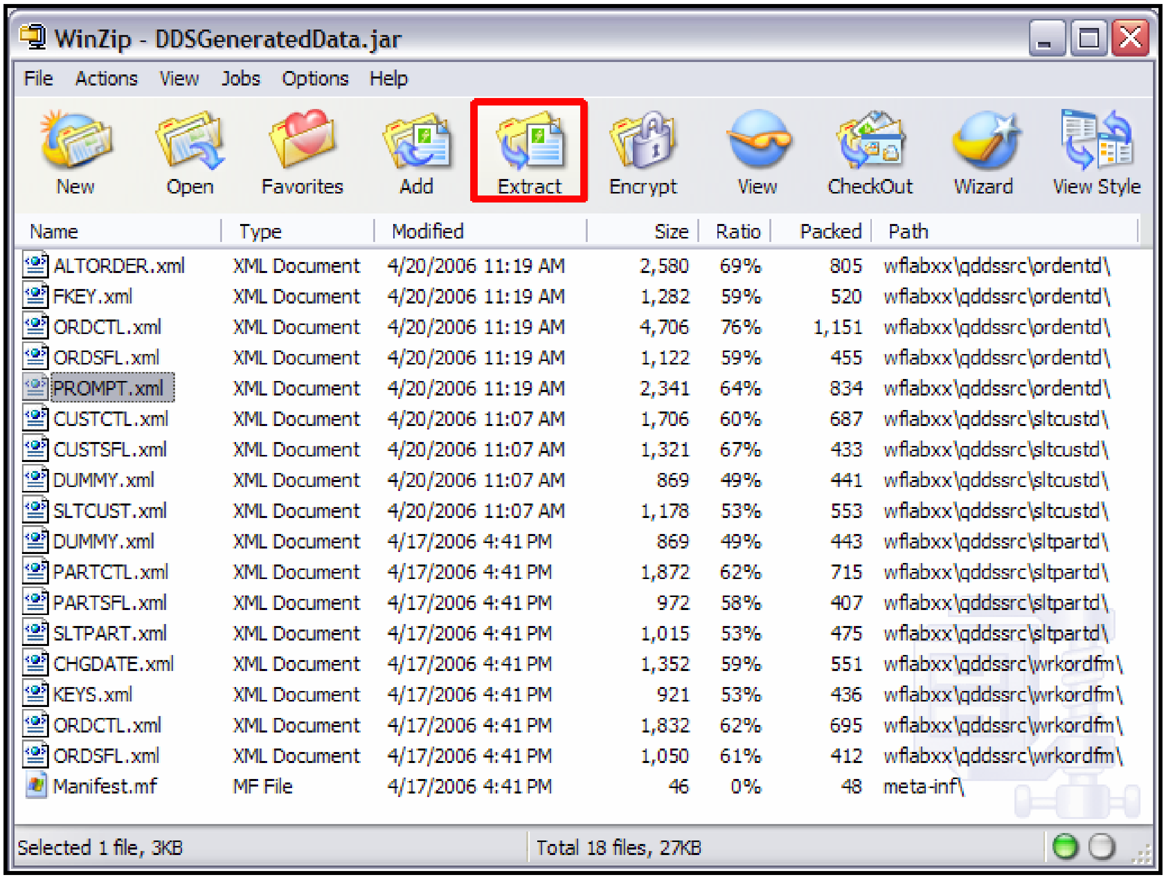
□ Overwrite existing files without warning

○ Create directory structure for files

● Create only selected directories

Figure 225. Locating files in ***File system***

7. Open the DDSGeneratedData.jar file with your .zip utility and extract the PROMPT.XML File (see Figure 226).

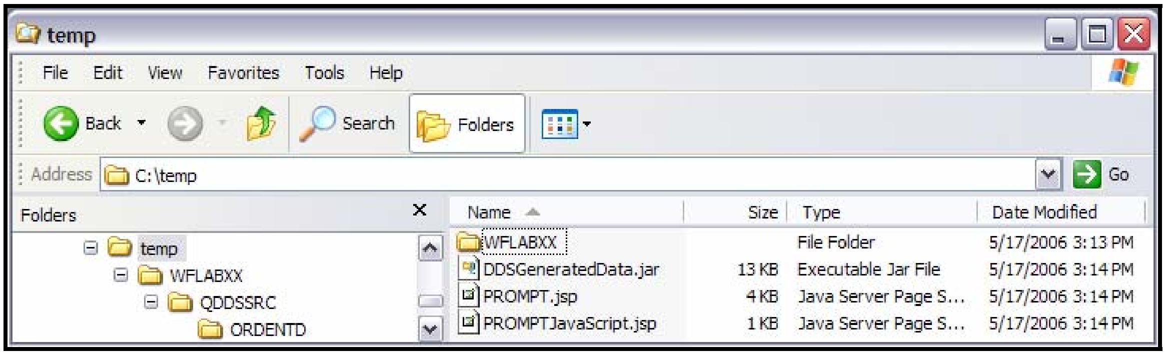

WinZip - DDSGeneratedData.jar
File Actions View Jobs Options Help
New
Open
Favorites
Add
Extract
Encrypt
View
CheckOut
Wizard
View Style

Name	Type	Modified	Size	Ratio	Packed	Path
ALTORDER.xml	XML Document	4/20/2006 11:19 AM	2,580	69%	805	wflabxx\qddssrc\ordentd\
FKEY.xml	XML Document	4/20/2006 11:19 AM	1,282	59%	520	wflabxx\qddssrc\ordentd\
ORDCTL.xml	XML Document	4/20/2006 11:19 AM	4,706	76%	1,151	wflabxx\qddssrc\ordentd\
ORDSFL.xml	XML Document	4/20/2006 11:19 AM	1,122	59%	455	wflabxx\qddssrc\ordentd\
PROMPT.xml	XML Document	4/20/2006 11:19 AM	2,341	64%	834	wflabxx\qddssrc\ordentd\
CUSTCTL.xml	XML Document	4/20/2006 11:07 AM	1,706	60%	687	wflabxx\qddssrc\sltcustd\
CUSTSFL.xml	XML Document	4/20/2006 11:07 AM	1,321	67%	433	wflabxx\qddssrc\sltcustd\
DUMMY.xml	XML Document	4/20/2006 11:07 AM	869	49%	441	wflabxx\qddssrc\sltcustd\
SLTCUST.xml	XML Document	4/20/2006 11:07 AM	1,178	53%	553	wflabxx\qddssrc\sltcustd\
DUMMY.xml	XML Document	4/17/2006 4:41 PM	869	49%	443	wflabxx\qddssrc\sltpartd\
PARTCTL.xml	XML Document	4/17/2006 4:41 PM	1,872	62%	715	wflabxx\qddssrc\sltpartd\
PARTSFL.xml	XML Document	4/17/2006 4:41 PM	972	58%	407	wflabxx\qddssrc\sltpartd\
SLTPART.xml	XML Document	4/17/2006 4:41 PM	1,015	53%	475	wflabxx\qddssrc\sltpartd\
CHGDATE.xml	XML Document	4/17/2006 4:41 PM	1,352	59%	551	wflabxx\qddssrc\wrkordfm\
KEYS.xml	XML Document	4/17/2006 4:41 PM	921	53%	436	wflabxx\qddssrc\wrkordfm\
ORDCTL.xml	XML Document	4/17/2006 4:41 PM	1,832	62%	695	wflabxx\qddssrc\wrkordfm\
ORDSFL.xml	XML Document	4/17/2006 4:41 PM	1,050	61%	412	wflabxx\qddssrc\wrkordfm\
Manifest.mf	MF File	4/17/2006 4:41 PM	46	0%	48	meta-inf\

Selected 1 file, 3KB — Total 18 files, 27KB

*Figure 226. Open the **DDSGeneratedData.jar** file and extract the **PROMPT.XML File**.*

8. When you are finished, the **c:temp** file system looks similar to Figure 227.



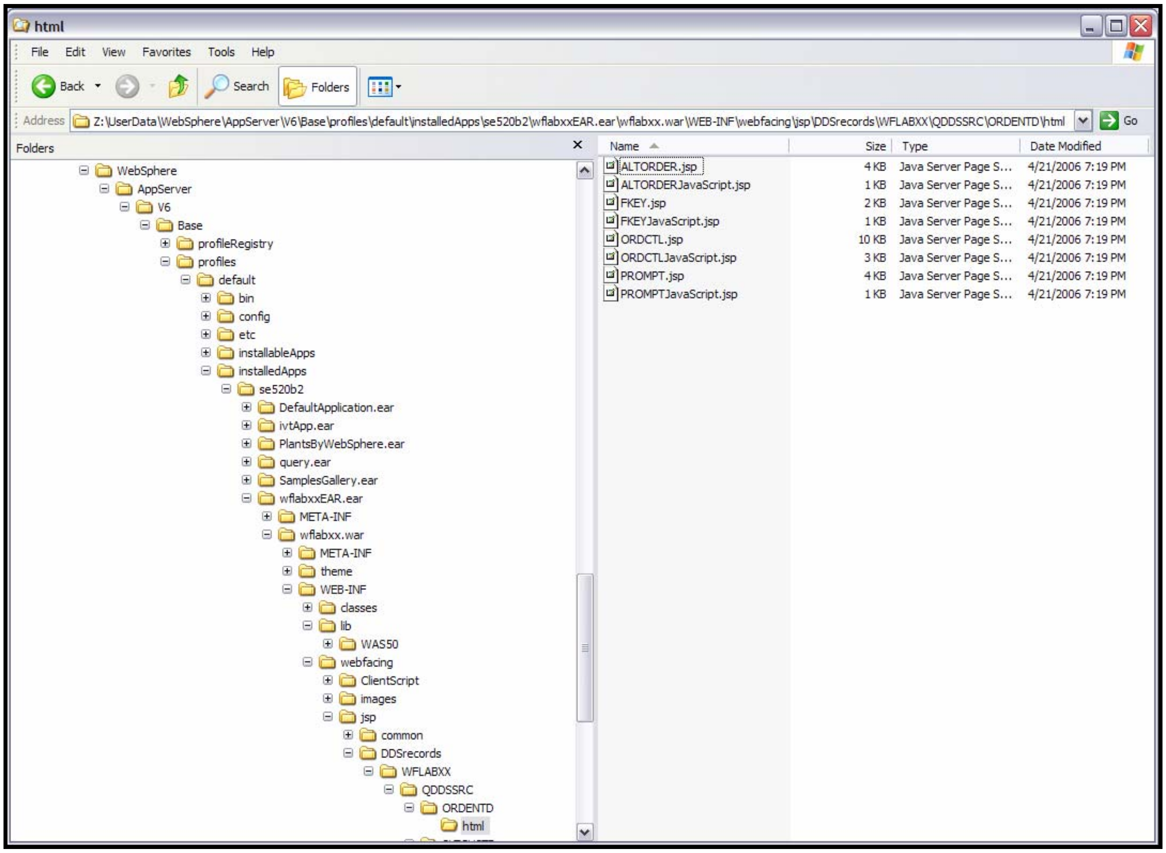
Name	Size	Type	Date Modified
WFLABXX		File Folder	5/17/2006 3:13 PM
DDSGeneratedData.jar	13 KB	Executable Jar File	5/17/2006 3:14 PM
PROMPT.jsp	4 KB	Java Server Page S...	5/17/2006 3:14 PM
PROMPTJavaScript.jsp	1 KB	Java Server Page S...	5/17/2006 3:14 PM

*Figure 227. The **c:temp** File system*

Notice that when you extracted PROMPT.XML you left the directory structure intact.

9. Now copy the PROMPT.jsp, PROMPTJavaScript.jsp, and the WFLABXX directory structure containing PROMPT.XML to the System i model.

10. The directory for the PROMPT.jsp and PROMPTJavaScript.jsp files is Z:\UserData\WebSphere\AppServer\V6\Base\profiles\default\installedApps\se520b2\wflabxxEAR.ear\wflabxx.war\WEB-INF\webfacing\jsp\DDSrecords\WFLABXX\QDDSSRC\ORDENTD\html, and is shown in the image below in Figure 228. The path can vary based on what version of WebSphere Application Server you are working with.



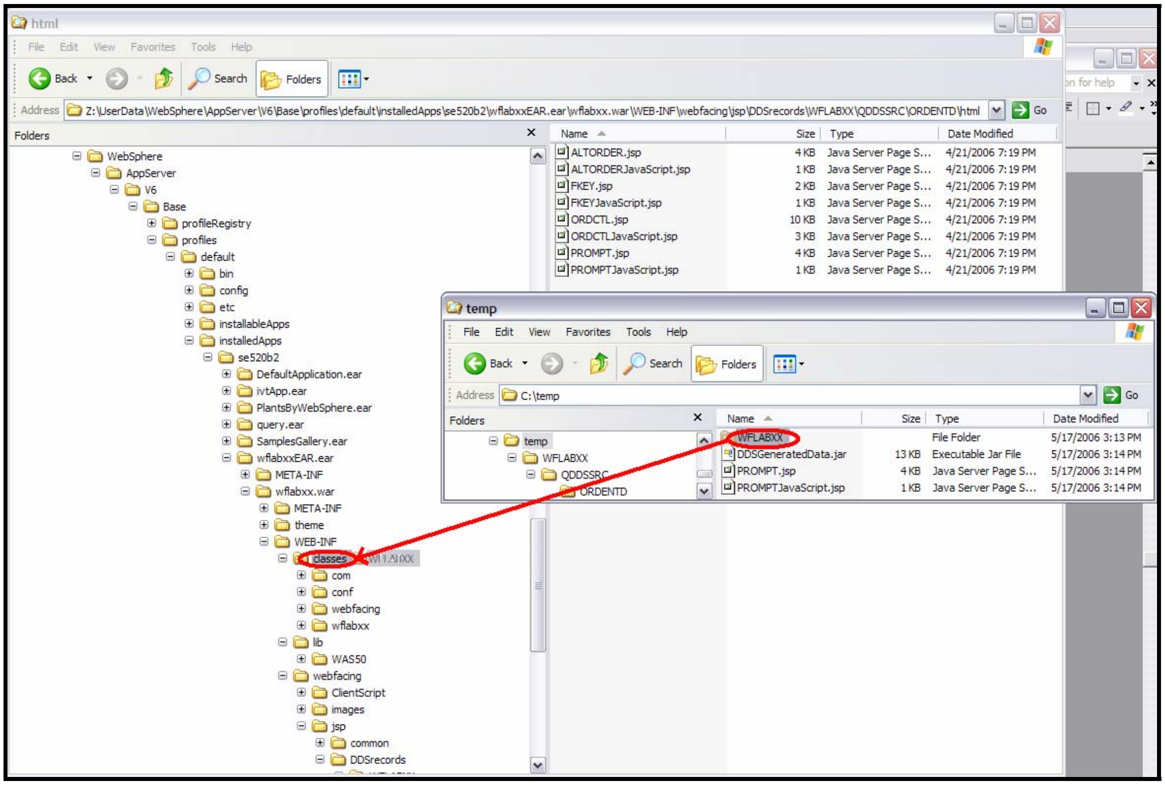
Address: Z:\UserData\WebSphere\AppServer\V6\Base\profiles\default\installedApps\se520b2\wflabxxEAR.ear\wflabxx.war\WEB-INF\webfacing\jsp\DDSrecords\WFLABXX\QDDSSRC\ORDENTD\html

```
⊟ WebSphere
  ⊟ AppServer
    ⊟ V6
      ⊟ Base
        ⊞ profileRegistry
        ⊟ profiles
          ⊟ default
            ⊞ bin
            ⊞ config
            ⊞ etc
            ⊞ installableApps
            ⊟ installedApps
              ⊟ se520b2
                ⊞ DefaultApplication.ear
                ⊞ ivtApp.ear
                ⊞ PlantsByWebSphere.ear
                ⊞ query.ear
                ⊞ SamplesGallery.ear
                ⊟ wflabxxEAR.ear
                  ⊞ META-INF
                  ⊟ wflabxx.war
                    ⊞ META-INF
                    ⊞ theme
                    ⊟ WEB-INF
                      ⊞ classes
                      ⊟ lib
                        ⊞ WAS50
                      ⊟ webfacing
                        ⊞ ClientScript
                        ⊞ images
                        ⊟ jsp
                          ⊞ common
                          ⊟ DDSrecords
                            ⊟ WFLABXX
                              ⊟ QDDSSRC
                                ⊟ ORDENTD
                                    html
```

Name	Size	Type	Date Modified
ALTORDER.jsp	4 KB	Java Server Page S...	4/21/2006 7:19 PM
ALTORDERJavaScript.jsp	1 KB	Java Server Page S...	4/21/2006 7:19 PM
FKEY.jsp	2 KB	Java Server Page S...	4/21/2006 7:19 PM
FKEYJavaScript.jsp	1 KB	Java Server Page S...	4/21/2006 7:19 PM
ORDCTL.jsp	10 KB	Java Server Page S...	4/21/2006 7:19 PM
ORDCTLJavaScript.jsp	3 KB	Java Server Page S...	4/21/2006 7:19 PM
PROMPT.jsp	4 KB	Java Server Page S...	4/21/2006 7:19 PM
PROMPTJavaScript.jsp	1 KB	Java Server Page S...	4/21/2006 7:19 PM

*Figure 228. Directory for the **PROMPT.jsp** and **PROMPTJavaScript.jsp** files*

11. Copy the workstation’s WFLABXX directory to the server’s **classes** directory as shown below in
Figure ***229***. Copying the XML file to the **classes** directory causes the WebSphere Application Server to pick up the XML files from this directory structure before it picks up files in the lib directory.



Address Z:\UserData\WebSphere\AppServer\V6\Base\profiles\default\installedApps\se520b2\wflabxxEAR.ear\wflabxx.war\WEB-INF\webfacing\jsp\DDSrecords\WFLABXX\QDDSSRC\ORDENTD\html

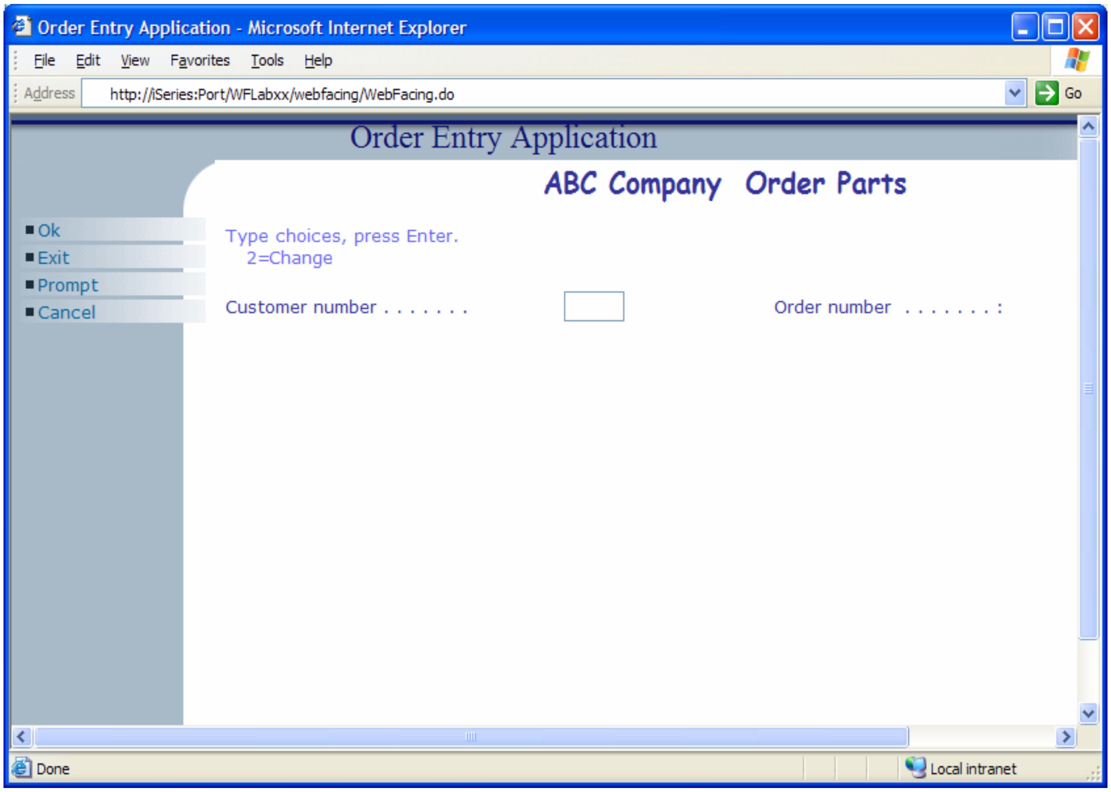
Name	Size	Type	Date Modified
ALTORDER.jsp	4 KB	Java Server Page S...	4/21/2006 7:19 PM
ALTORDERJavaScript.jsp	1 KB	Java Server Page S...	4/21/2006 7:19 PM
FKEY.jsp	2 KB	Java Server Page S...	4/21/2006 7:19 PM
FKEYJavaScript.jsp	1 KB	Java Server Page S...	4/21/2006 7:19 PM
ORDCTL.jsp	10 KB	Java Server Page S...	4/21/2006 7:19 PM
ORDCTLJavaScript.jsp	3 KB	Java Server Page S...	4/21/2006 7:19 PM
PROMPT.jsp	4 KB	Java Server Page S...	4/21/2006 7:19 PM
PROMPTJavaScript.jsp	1 KB	Java Server Page S...	4/21/2006 7:19 PM

Address C:\temp

Name	Size	Type	Date Modified
WFLABXX		File Folder	5/17/2006 3:13 PM
DDSGeneratedData.jar	13 KB	Executable Jar File	5/17/2006 3:14 PM
PROMPT.jsp	4 KB	Java Server Page S...	5/17/2006 3:14 PM
PROMPTJavaScript.jsp	1 KB	Java Server Page S...	5/17/2006 3:14 PM

Figure 229. Copy the workstation’s ***WFLABXX*** *directory to the server’s* ***classes*** *directory*

12. Stop and restart your instance of WebSphere Application Server. When restarted, you see your changes (see
Figure ***230***).



Order Entry Application - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Address
http://iSeries:Port/WFLabxx/webfacing/WebFacing.do
Go
Order Entry Application
Ok
Exit
Prompt
Cancel
Done
Local intranet

ABC Company Order Parts

Type choices, press Enter.
  2=Change

Customer number . . . .   Order number . . . . :

Figure 230. Changes shown after restarting WebSphere Application Server

Exercise 13.2 JSP precompile options

WebSphere Application Server for iSeries has a unique feature known as JSP Pre-Touch for precompiling JSP files.

What is JSP Pre-Touch?

JSP Pre-Touch is a tool for compiling and class loading JSPs during application startup. It is configured with a series of initialization parameters that are added to a Web module's ibm-web-ext.xmi file.

JSP Pre-Touch is used to compile JSPs and reduce the amount of time it takes to invoke a JSP after an application starts (called *first touch*), even when the JSP has already been compiled.

When to use JSP Pre-Touch

Use the JSP Pre-Touch mechanism when all JSPs need to be compiled asynchronously.

A secondary use is to class load all JSPs for an application into WebSphere Application Server. This removes the class load penalty paid by the first user to access a JSP file.

When not to use JSP Pre-Touch

Avoid running the Pre-Touch tool in a large application startup in production environments. Loading large numbers of infrequently used JSPs up front has tradeoffs. It can significantly increase the JVM heap size, which can cause memory bottlenecks at startup and during garbage collection cycles.

JSP compilation

To enable the JSP Pre-Touch, three possible attributes can be configured. Each of the attribute settings reside in the ibm-web-ext.xmi file of a Web module (found in the WEB-INF directory). Setting the parameters in this file can be achieved using the WebSphere Development Studio Tools, the Application Assembly Tool, or any text editor.

The two most important attributes are prepareJSPs and prepareJSP. The third attribute, prepareJSPThreadCount, can be used on multi-processor systems.

The prepareJSP attribute

When the attribute, prepareJSP is present, all JSP files are compiled on WebSphere Application Server startup. This process happens in a separate thread, so the WebSphere Application Server might, depending on the quantity and size of JSP that the application has, finish starting up before the JSP files are finished compiling. If starting and running the Web-enabled application seems slow, it can be because the JSP files are still compiling. Check the server log files to know when all of the JSP files have been compiled.

The numeric attribute value represents the minimum size (in kilobytes) that a JSP must be for it to be class loaded and JIT-compiled. The default is 0, which causes all JSPs to be class loaded and JIT-compiled.

Set the prepareJSP attribute

Set the prepareJSP attribute to a value that is a request parameter composed of an alphanumeric that the JSP never expects to receive. This attribute is used to perform a quick exit from the service method of each JSP when they are prepared by this tool. This enables the tool to work much faster and prevent exceptions from showing in the WebSphere Application Server logs as a result of running the JSP at startup.

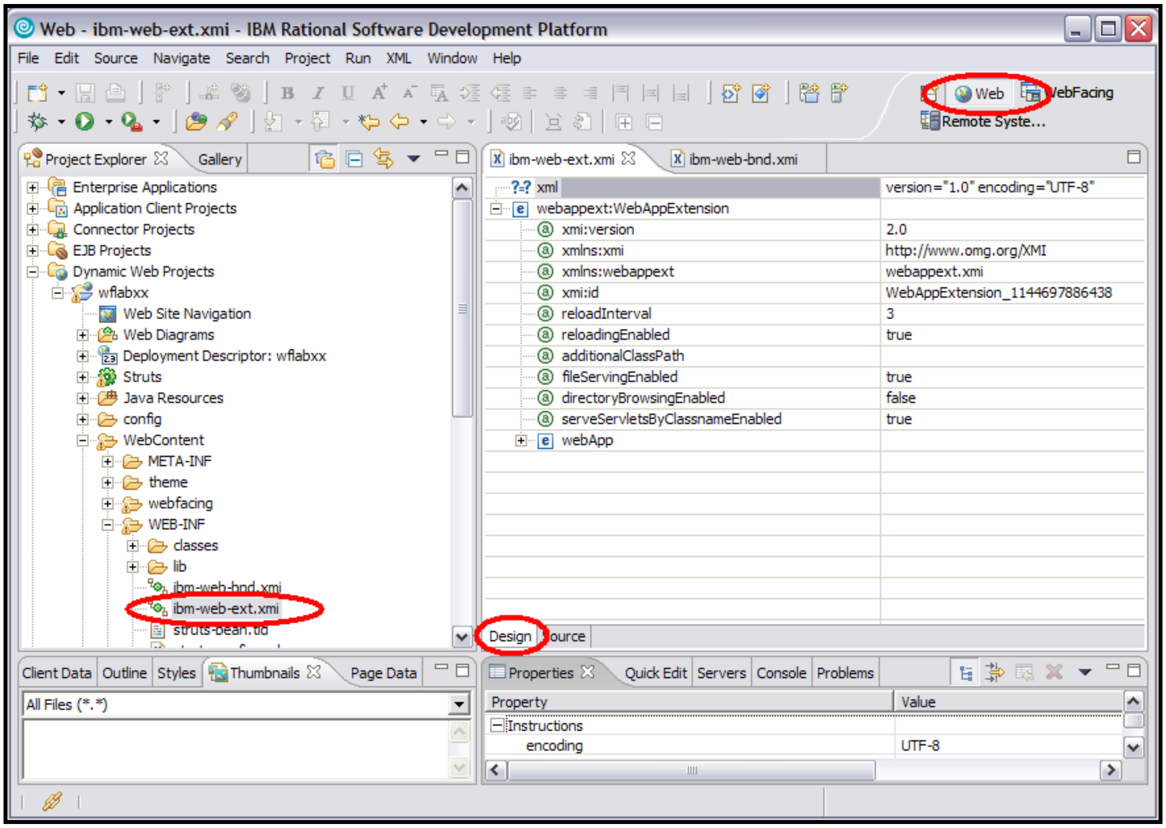
If this attribute is not set on a Web-enabled application but the prepareJSP attribute is set, exceptions are thrown by the application when the service method for the JSP is called because the WebFacing run time has not been initialized. This puts many error messages in the log, uses heap space, and causes the garbage collector to run.

Tip: If precompiling the JSP, use the prepareJSP attribute setting as well.

Exercise 13.3 Configuring the Pre-Touch attributes

The Pre-Touch attributes can be configured using the WebSphere Development Studio Client.

1. To set the attributes, open the Web Perspective, then open the ibm-web-ext.xmi file (see Figure ***231***).

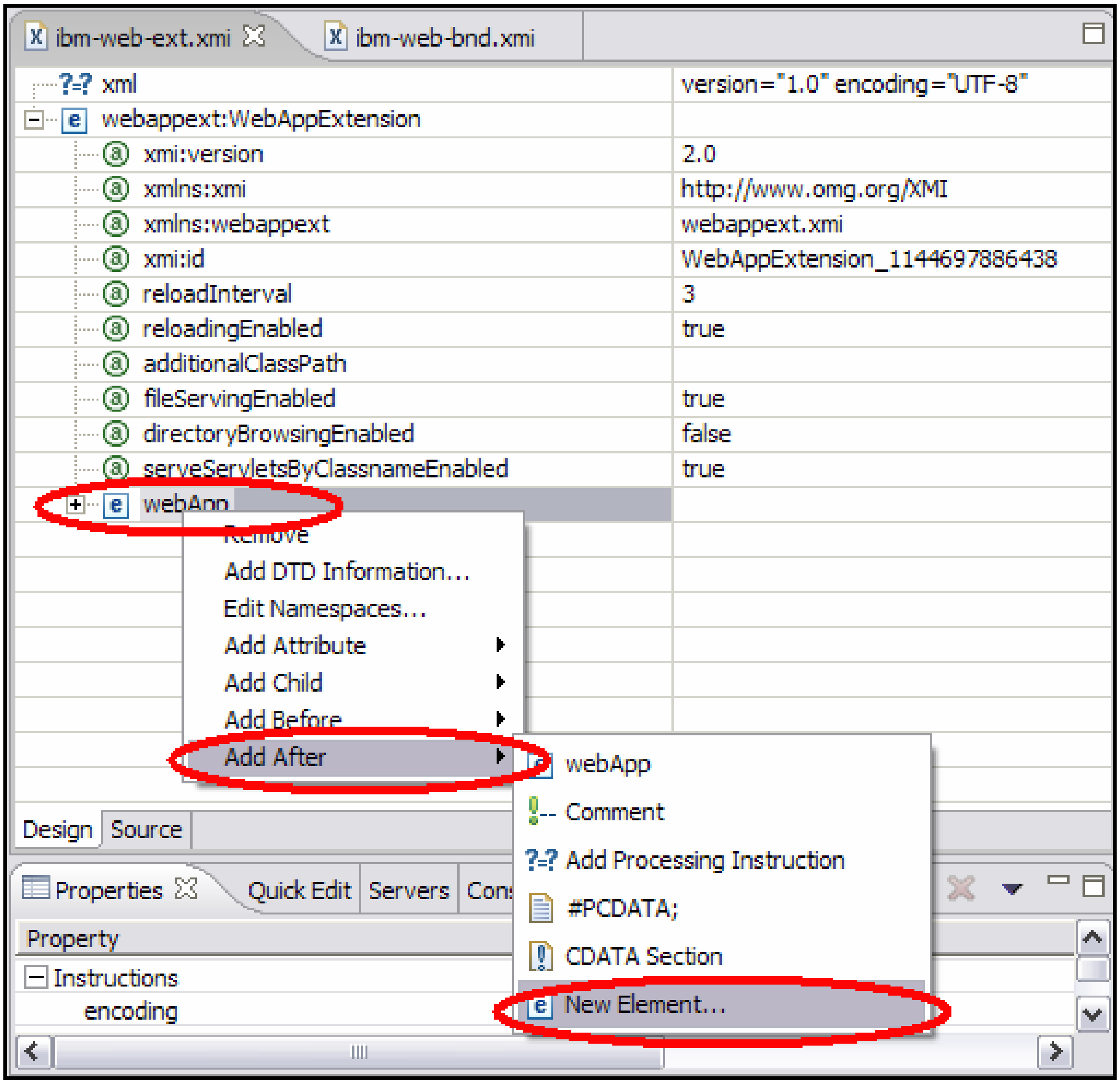


xml	version="1.0" encoding="UTF-8"
webappext:WebAppExtension	
xmi:version	2.0
xmlns:xmi	http://www.omg.org/XMI
xmlns:webappext	webappext.xmi
xmi:id	WebAppExtension_1144697886438
reloadInterval	3
reloadingEnabled	true
additionalClassPath	
fileServingEnabled	true
directoryBrowsingEnabled	false
serveServletsByClassnameEnabled	true
webApp	

Property	Value
Instructions	
encoding	UTF-8

Figure 231. Setting the attributes

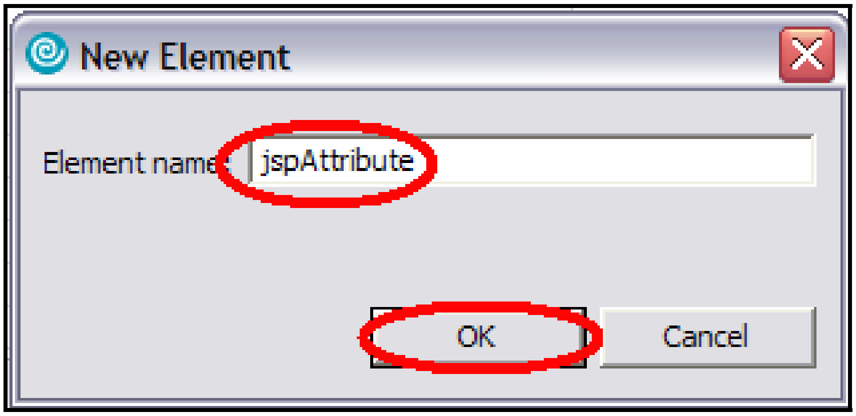
2. Open the webappext element. Right-click the **webApp** element and select **Add After** as shown in
Figure ***232***.



?-? xml	version="1.0" encoding="UTF-8"
e webappext:WebAppExtension	
xmi:version	2.0
xmlns:xmi	http://www.omg.org/XMI
xmlns:webappext	webappext.xmi
xmi:id	WebAppExtension_1144697886438
reloadInterval	3
reloadingEnabled	true
additionalClassPath	
fileServingEnabled	true
directoryBrowsingEnabled	false
serveServletsByClassnameEnabled	true
e webApp	

*Figure 232. Open the **webappext** element.*

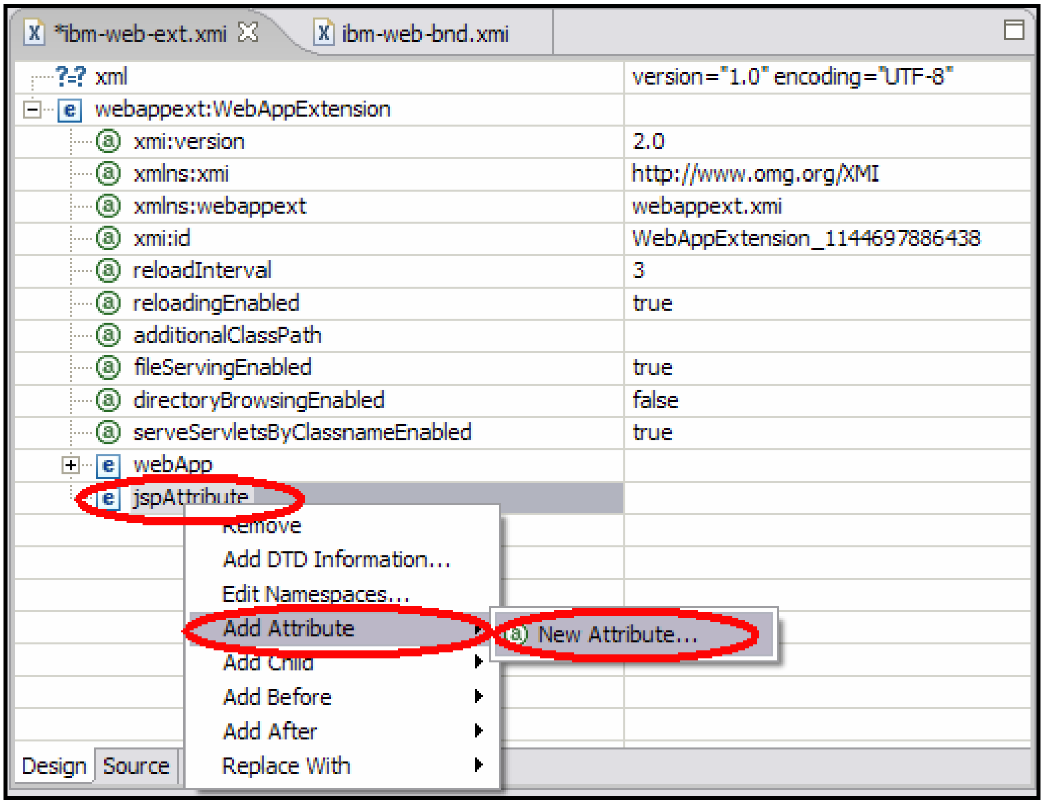
3. In the dialog that appears, type `jspAttribute` and press **OK** (see
Figure ***233***).



Element name: jspAttribute

*Figure 233. Type **jspAttribute** and press **OK**.*

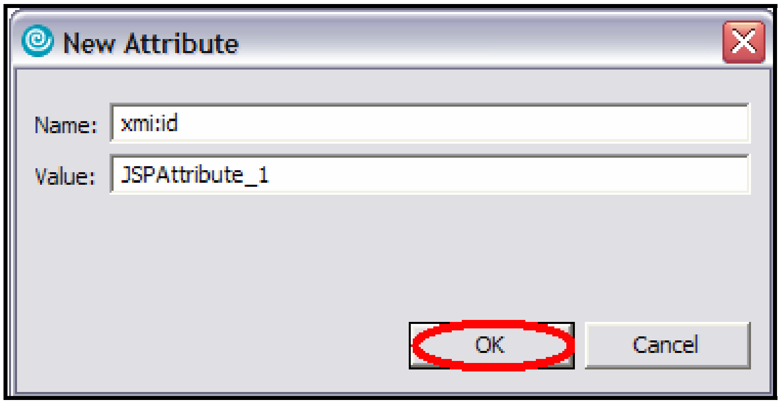
4. Highlight the newly created attribute, right-click, select **Add Attribute > New Attribute** as shown below in
Figure ***234***.



?=? xml	version="1.0" encoding="UTF-8"
webappext:WebAppExtension	
xmi:version	2.0
xmlns:xmi	http://www.omg.org/XMI
xmlns:webappext	webappext.xmi
xmi:id	WebAppExtension_1144697886438
reloadInterval	3
reloadingEnabled	true
additionalClassPath	
fileServingEnabled	true
directoryBrowsingEnabled	false
serveServletsByClassnameEnabled	true
webApp	
jspAttribute	

*Figure 234. Right click **jspAttribute** and select **Add Attribute > New Attribute**.*

5. In the dialog box that appears, enter **xmi:id** in the **Name** field, and **JSPAttribute_1** in the **Value** field. Your panel looks similar to
Figure ***235***. Press **OK**.



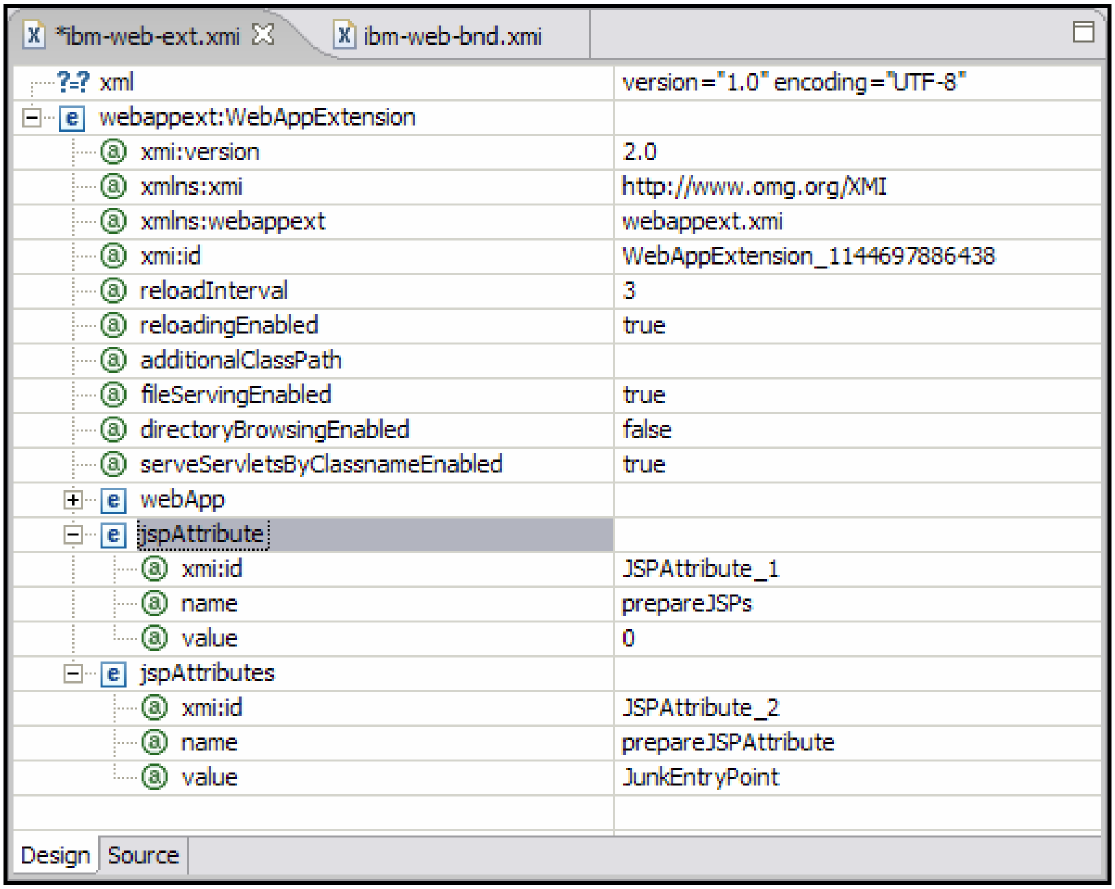
Name: xmi:id

Value: JSPAttribute_1

*Figure 235. Enter **xmi:id** in the **Name** field, and **JSPAttribute_1** in the **Value** field.*

6. Continue this process until the screen looks like the screen below in Figure ***236***. To summarize, there are two jspAttributes elements to be added. One jspAttribute has an attribute xmi:id with a value of JSPAttribute_1, a name attribute with a value of prepareJSPs, and a value attribute with a value of 0. The second jspAttributes entry has an attribute xmi:id with a value of JSPAttribute_2, a name attribute with a value of prepareJSPAttribute, and a value attribute with a value of JunkEntryPoint.

   Note: Everything is case sensitive so type things as defined in the screen shot.

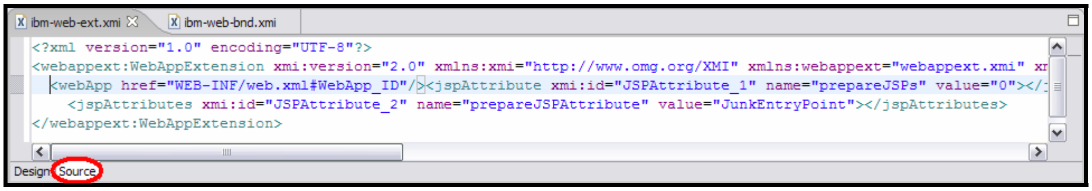


?=? xml	version="1.0" encoding="UTF-8"
webappext:WebAppExtension	
xmi:version	2.0
xmlns:xmi	http://www.omg.org/XMI
xmlns:webappext	webappext.xmi
xmi:id	WebAppExtension_1144697886438
reloadInterval	3
reloadingEnabled	true
additionalClassPath	
fileServingEnabled	true
directoryBrowsingEnabled	false
serveServletsByClassnameEnabled	true
webApp	
jspAttribute	
xmi:id	JSPAttribute_1
name	prepareJSPs
value	0
jspAttributes	
xmi:id	JSPAttribute_2
name	prepareJSPAttribute
value	JunkEntryPoint

Figure 236. Screen results of ***jspAttributes*** *elements*

7. Save the changes When completed. To see what the changes look like in raw xml format, click the **Source** tab from within the WebSphere Development Studio Client environment located near the bottom of the XML editor (see Figure ***237***).

   Note: The jspAttributes that have been added to the file. These entries can be made manually to a deployed WebFaced application by editing the ibm-web-ext.xmi file. After all JSPs have been compiled, it is a good idea to remove these entries, which can be done manually using a normal text editor. Failure to remove these entries causes all JSPs to be class loaded upon WebSphere Application Server startup. As mentioned earlier for large applications, this can introduce resource issues. For smaller applications, leaving these entries in an XML file work well.



```
<?xml version="1.0" encoding="UTF-8"?>
<webappext:WebAppExtension xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:webappext="webappext.xmi" xr
  <webApp href="WEB-INF/web.xml#WebApp_ID"/><jspAttribute xmi:id="JSPAttribute_1" name="prepareJSPs" value="0"></:
    <jspAttributes xmi:id="JSPAttribute_2" name="prepareJSPAttribute" value="JunkEntryPoint"></jspAttributes>
</webappext:WebAppExtension>
```

Figure 237. XML format

8. After you have added the jspAttributes settings, test the changes. This is done by deploying the application to WebSphere Application Server on i5/OS as done earlier. This lab does not walk you through those steps, so refer to how to deploy the application in exercise 12. You either have to stop your currently running WFLabxx application or deploy a WAR file with a different context root. When the application is installed, start the application server. When the application server is started, WebSphere Application Server will start to compile the JSP files. This can be verified by viewing the System.out log file.

   To begin, open the mapped drive to the System i model.

   The SystemOut.log file is located in the
   /<InstanceName>/logs/ directory of your WebSphere Application Server instance.

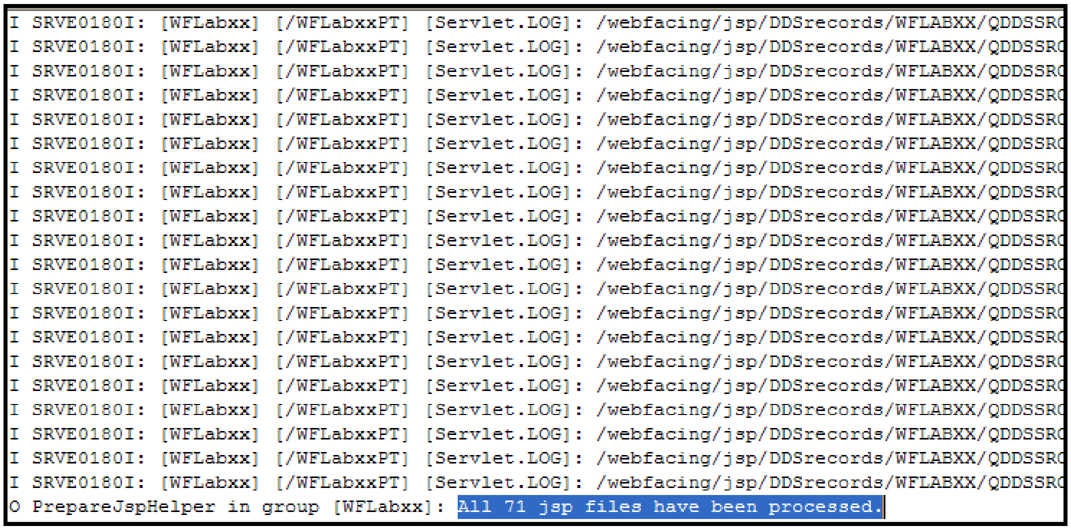
9. You can find the file by using Internet Explorer. Open the System.out log file using the editpad lite application installed on your PC. Both NotePad and WordPad are unacceptable for viewing the log file. Scroll down the log file to find a line similar to:

   [9/1/04 8:25:07:078 UTC] b29ffde8 SystemOut   O PrepareJspHelper in group [WFLabxx]: 25 jsp files have been processed.

   For every 25 JSP files, WebSphere Application Server writes this entry to the log file. You might need to refresh the editor until you see this.

   Eventually, you see an entry similar to:
   [9/1/04 8:25:09:935 UTC] b1af809f SystemOut   O PrepareJspHelper in group [WFLabxx]: All 71 jsp files have been processed.

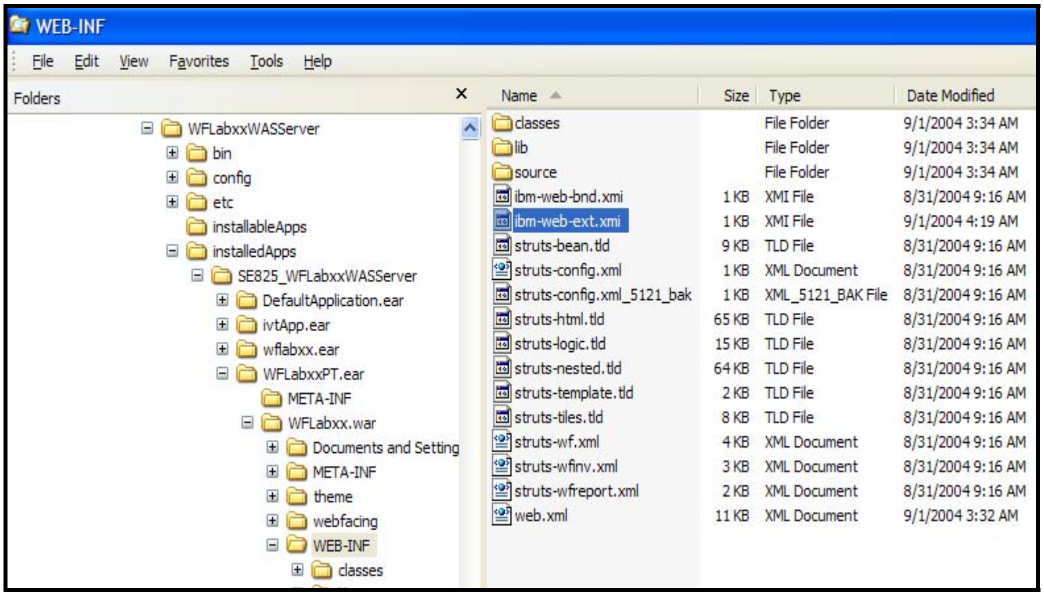
The important thing to look for is the "All." This means that all JSP's have been compiled and class loaded (see
Figure ***238***).



```
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
I SRVE0180I: [WFLabxx] [/WFLabxxPT] [Servlet.LOG]: /webfacing/jsp/DDSrecords/WFLABXX/QDDSSRC
O PrepareJspHelper in group [WFLabxx]: All 71 jsp files have been processed.
```

Figure 238. Processed jsp files

10. The next step in the process is to remove the jspAttributes setting. This might not be required for small applications, but typically for large applications, you do not want all JSPs class loaded. The entry in the jspAttributes settings in ibm-web-ext.xmi must be removed. Remember, when you deploy an application to WebSphere Application Server, the ibm-web-ext.xmi exists in two places. One is located the installed apps directory:

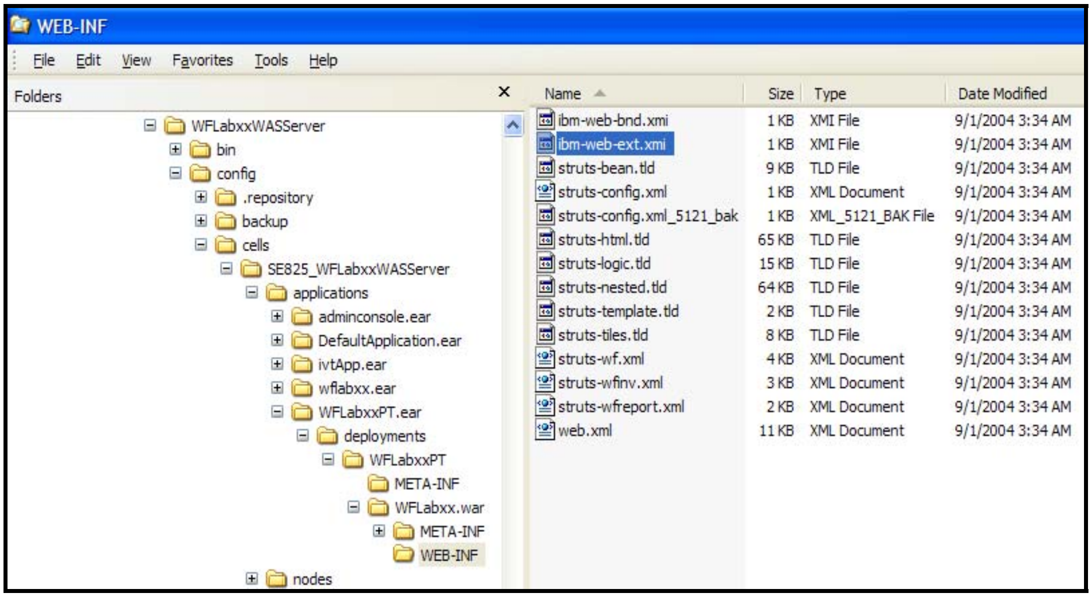


Name	Size	Type	Date Modified
classes		File Folder	9/1/2004 3:34 AM
lib		File Folder	9/1/2004 3:34 AM
source		File Folder	9/1/2004 3:34 AM
ibm-web-bnd.xmi	1 KB	XMI File	8/31/2004 9:16 AM
ibm-web-ext.xmi	1 KB	XMI File	9/1/2004 4:19 AM
struts-bean.tld	9 KB	TLD File	8/31/2004 9:16 AM
struts-config.xml	1 KB	XML Document	8/31/2004 9:16 AM
struts-config.xml_5121_bak	1 KB	XML_5121_BAK File	8/31/2004 9:16 AM
struts-html.tld	65 KB	TLD File	8/31/2004 9:16 AM
struts-logic.tld	15 KB	TLD File	8/31/2004 9:16 AM
struts-nested.tld	64 KB	TLD File	8/31/2004 9:16 AM
struts-template.tld	2 KB	TLD File	8/31/2004 9:16 AM
struts-tiles.tld	8 KB	TLD File	8/31/2004 9:16 AM
struts-wf.xml	4 KB	XML Document	8/31/2004 9:16 AM
struts-wfinv.xml	3 KB	XML Document	8/31/2004 9:16 AM
struts-wfreport.xml	2 KB	XML Document	8/31/2004 9:16 AM
web.xml	11 KB	XML Document	9/1/2004 3:32 AM

*Figure 239. The **ibm-web-ext.xmi** file located in the installed apps directory*

This is not the one to change. Think of it as the *Prod* data side of the application.

The other file cells directory as shown below in
Figure ***240***:

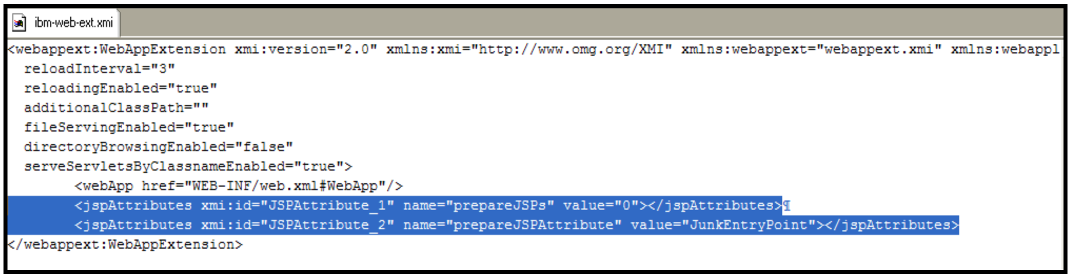


Name	Size	Type	Date Modified
ibm-web-bnd.xmi	1 KB	XMI File	9/1/2004 3:34 AM
ibm-web-ext.xmi	1 KB	XMI File	9/1/2004 3:34 AM
struts-bean.tld	9 KB	TLD File	9/1/2004 3:34 AM
struts-config.xml	1 KB	XML Document	9/1/2004 3:34 AM
struts-config.xml_5121_bak	1 KB	XML_5121_BAK File	9/1/2004 3:34 AM
struts-html.tld	65 KB	TLD File	9/1/2004 3:34 AM
struts-logic.tld	15 KB	TLD File	9/1/2004 3:34 AM
struts-nested.tld	64 KB	TLD File	9/1/2004 3:34 AM
struts-template.tld	2 KB	TLD File	9/1/2004 3:34 AM
struts-tiles.tld	8 KB	TLD File	9/1/2004 3:34 AM
struts-wf.xml	4 KB	XML Document	9/1/2004 3:34 AM
struts-wfinv.xml	3 KB	XML Document	9/1/2004 3:34 AM
struts-wfreport.xml	2 KB	XML Document	9/1/2004 3:34 AM
web.xml	11 KB	XML Document	9/1/2004 3:34 AM

*Figure 240. The **ibm-web-ext.xmi** file located in a different directory*

This is the ibm-web-ext.xmi file that is edited.

11. To remove these entries, open the file with editpad lite. Remove the two jspAttributes lines (see
Figure ***241***).



```
<webappext:WebAppExtension xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:webappext="webappext.xmi" xmlns:webappl
  reloadInterval="3"
  reloadingEnabled="true"
  additionalClassPath=""
  fileServingEnabled="true"
  directoryBrowsingEnabled="false"
  serveServletsByClassnameEnabled="true">
        <webApp href="WEB-INF/web.xml#WebApp"/>
        <jspAttributes xmi:id="JSPAttribute_1" name="prepareJSPs" value="0"></jspAttributes>¶
        <jspAttributes xmi:id="JSPAttribute_2" name="prepareJSPAttribute" value="JunkEntryPoint"></jspAttributes>
</webappext:WebAppExtension>
```

Figure 241. Remove the two jspAttributes lines.

12. Save the file.
13. Stop and restart your server.
14. Go back into the System.out log file. For the latest restart of the server, the JSP precompiler does not try compiling the JSPs.

Recap

You have completed the “IBM WebFacing Tool deployment options” section. You now understand how to:

- Partially deploy an IBM WebFacing Tool application
- Use Pre-Touch and JSP Precompile

Creating a WebFacing portlet project

In this chapter, you learn how to create a WebFacing project that runs as a portlet inside a Portal server. For this chapter, you convert the Order Entry application used previously.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 14.1: Starting the WebFacing Web Portlet Project wizard
- Exercise 14.2: Selecting display file members to convert
- Exercise 14.3: Specifying the CL command to launch the application
- Exercise 14.4: Selecting a Web style
- Exercise 14.5: Completing the WebFacing project information
- Exercise 14.6: Creating the Portal test server
- Exercise 14.7: Adding the portlet project to the portal server
- Exercise 14.8: Staring the portal server
- Exercise 14.9: Testing the WebFacing portlet application

The exercises in this chapter must be completed in order. Start with “Exercise 14.1: Starting the WebFacing Web Portlet Project wizard.”

Length of time

This chapter takes approximately 30 minutes to complete.

Exercise 14.1: Starting the WebFacing Web Portlet Project wizard

To start the WebFacing Portlet Project wizard:

1. In the WebFacing perspective, click **File > New > Project** from the workbench menu (see Figure 242).

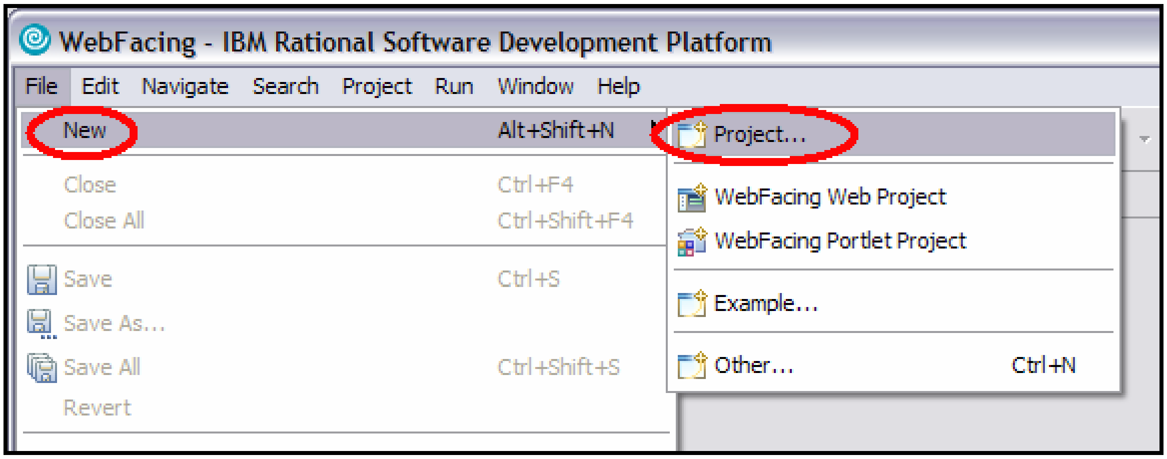
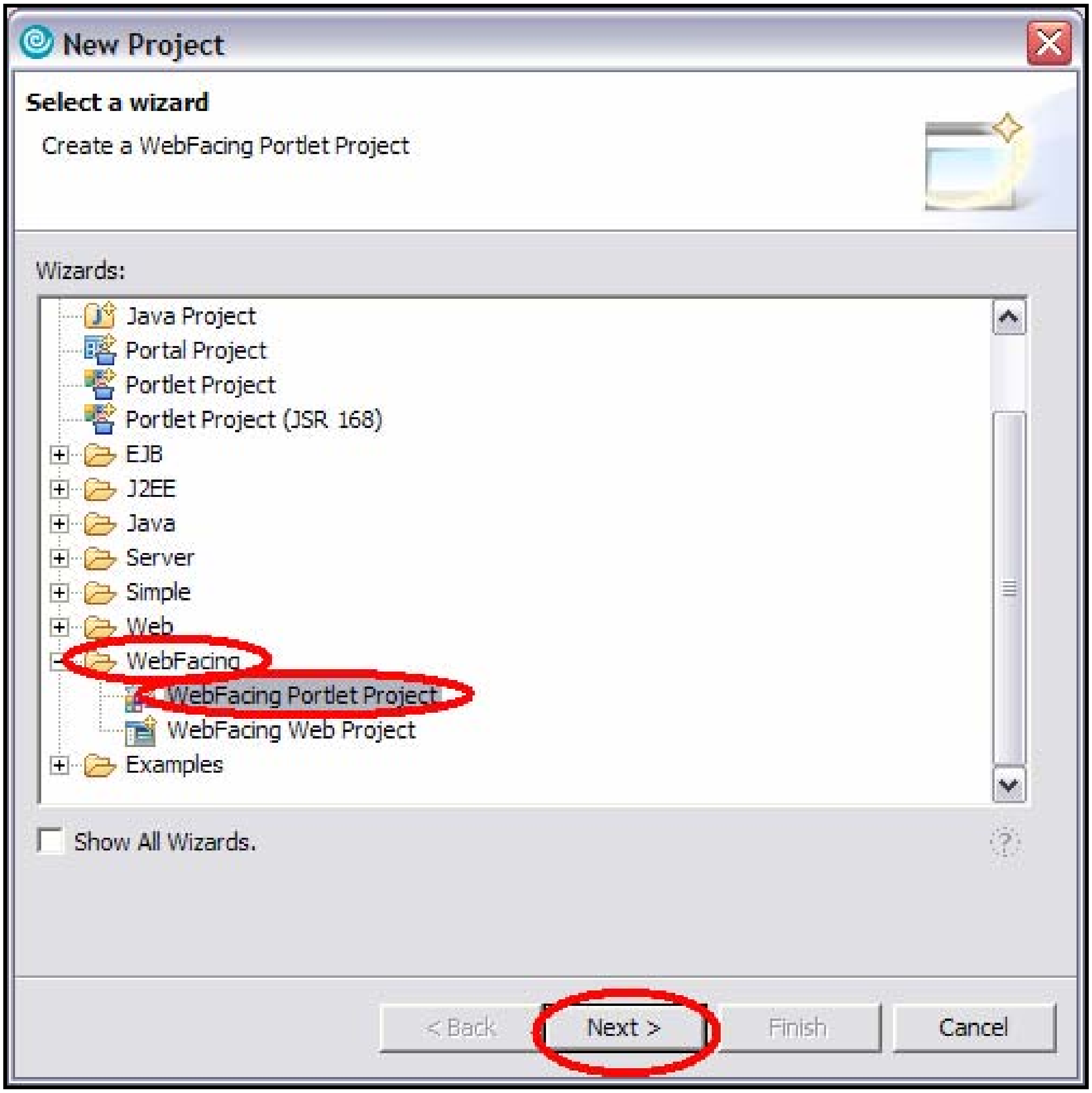


Figure 242. Starting the WebFacing Portlet Project wizard

The New Project dialog is displayed.

2. In the left pane, select **WebFacing** to expand its options (see Figure 243).
3. In the expanded list below WebFacing, select **WebFacing Portlet Project** (see Figure 243).



New Project

Select a wizard

Create a WebFacing Portlet Project

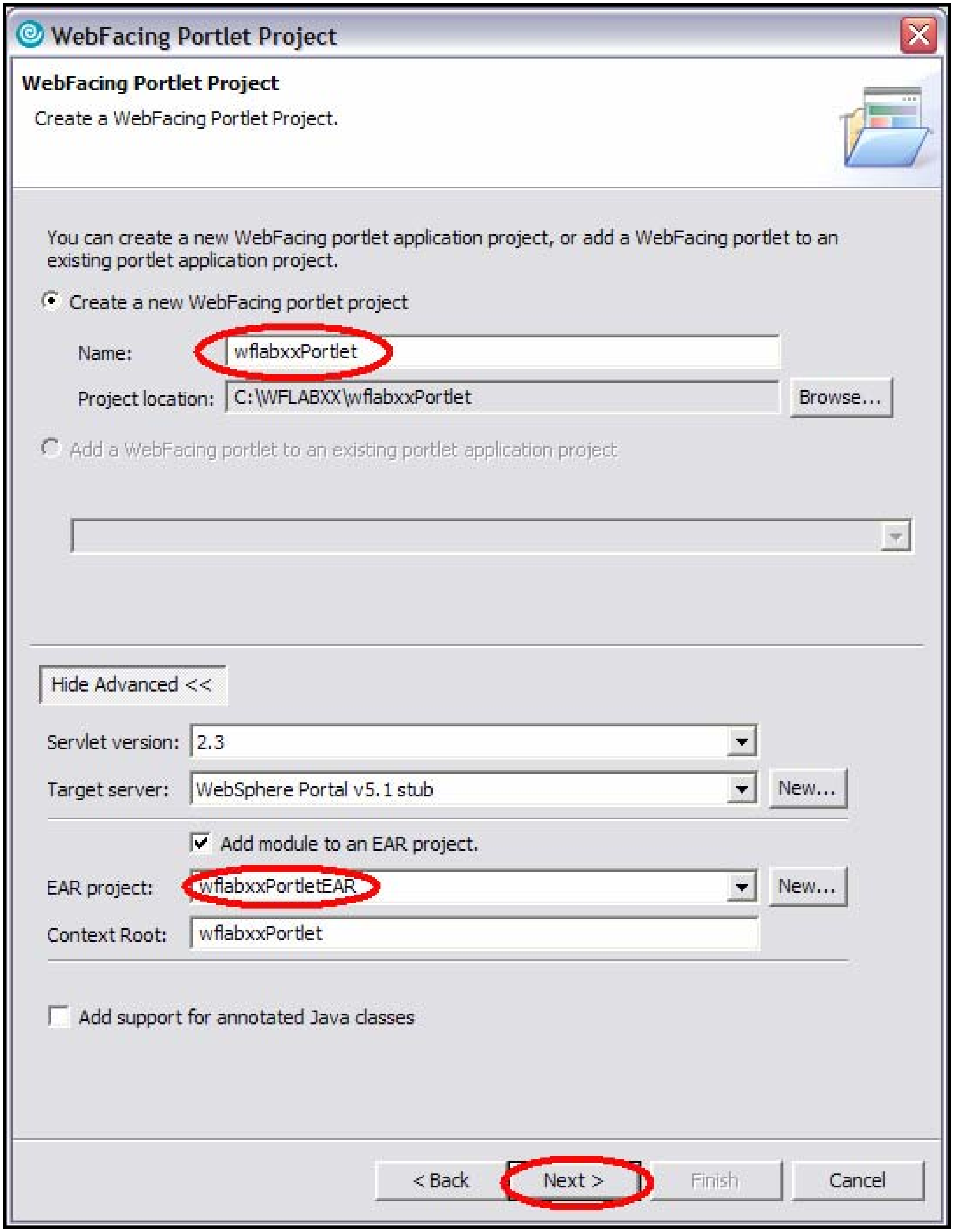
Wizards:

*Figure 243. Select **WebFacing Portlet Project**.*

4. Click **Next**.

This starts the **WebFacing Portlet Project** wizard.

5. In the WebFacing Portlet Project wizard, you are presented with the Create a WebFacing Portlet Project page (see Figure 244).



WebFacing Portlet Project

WebFacing Portlet Project

Create a WebFacing Portlet Project.

You can create a new WebFacing portlet application project, or add a WebFacing portlet to an existing portlet application project.

● Create a new WebFacing portlet project

Name: wflabxxPortlet

Project location: C:\WFLABXX\wflabxxPortlet

○ Add a WebFacing portlet to an existing portlet application project

Servlet version: 2.3

Target server: WebSphere Portal v5.1 stub

☑ Add module to an EAR project.

EAR project: wflabxxPortletEAR

Context Root: wflabxxPortlet

□ Add support for annotated Java classes

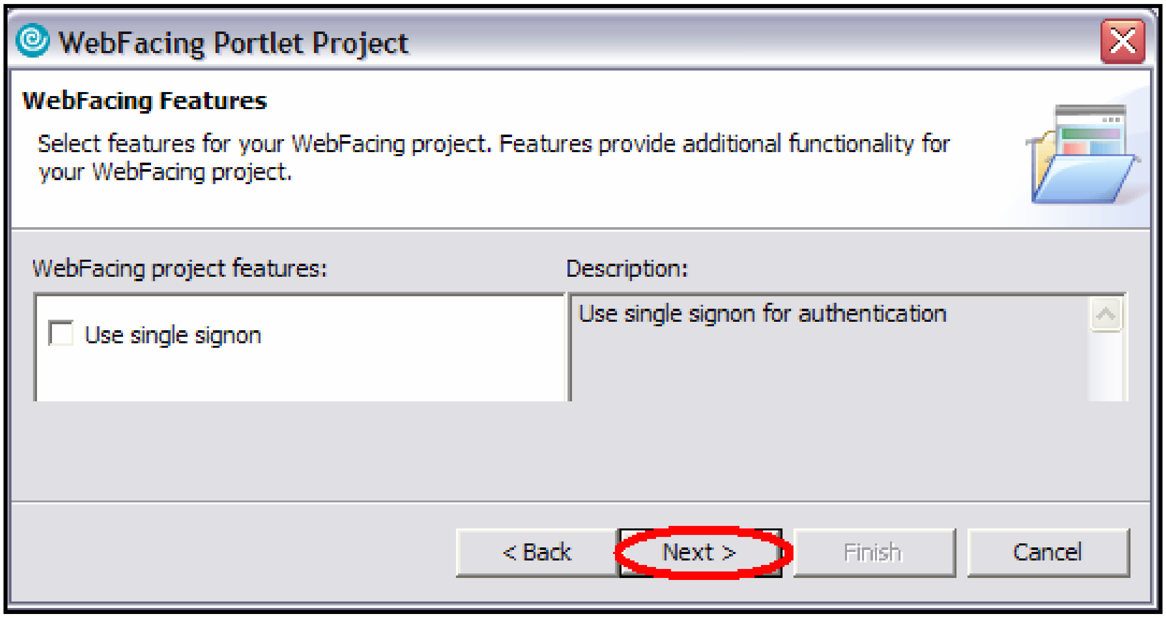
Figure 244. The Create a WebFacing Portlet Project page

6. Click the **Show Advanced >>** button to see additional panel fields.
7. In the N**ame** field, type the portlet project name **wflabxxPortlet**.

   The **EAR Project:** (Enterprise Archive) field is updated automatically.

   This creates a unique Application file for this WebFacing project.

8. Click **Next**.
9. The WebFacing Features panel opens (see Figure 245).



WebFacing Portlet Project

WebFacing Features

Select features for your WebFacing project. Features provide additional functionality for your WebFacing project.

WebFacing project features:

□ Use single signon

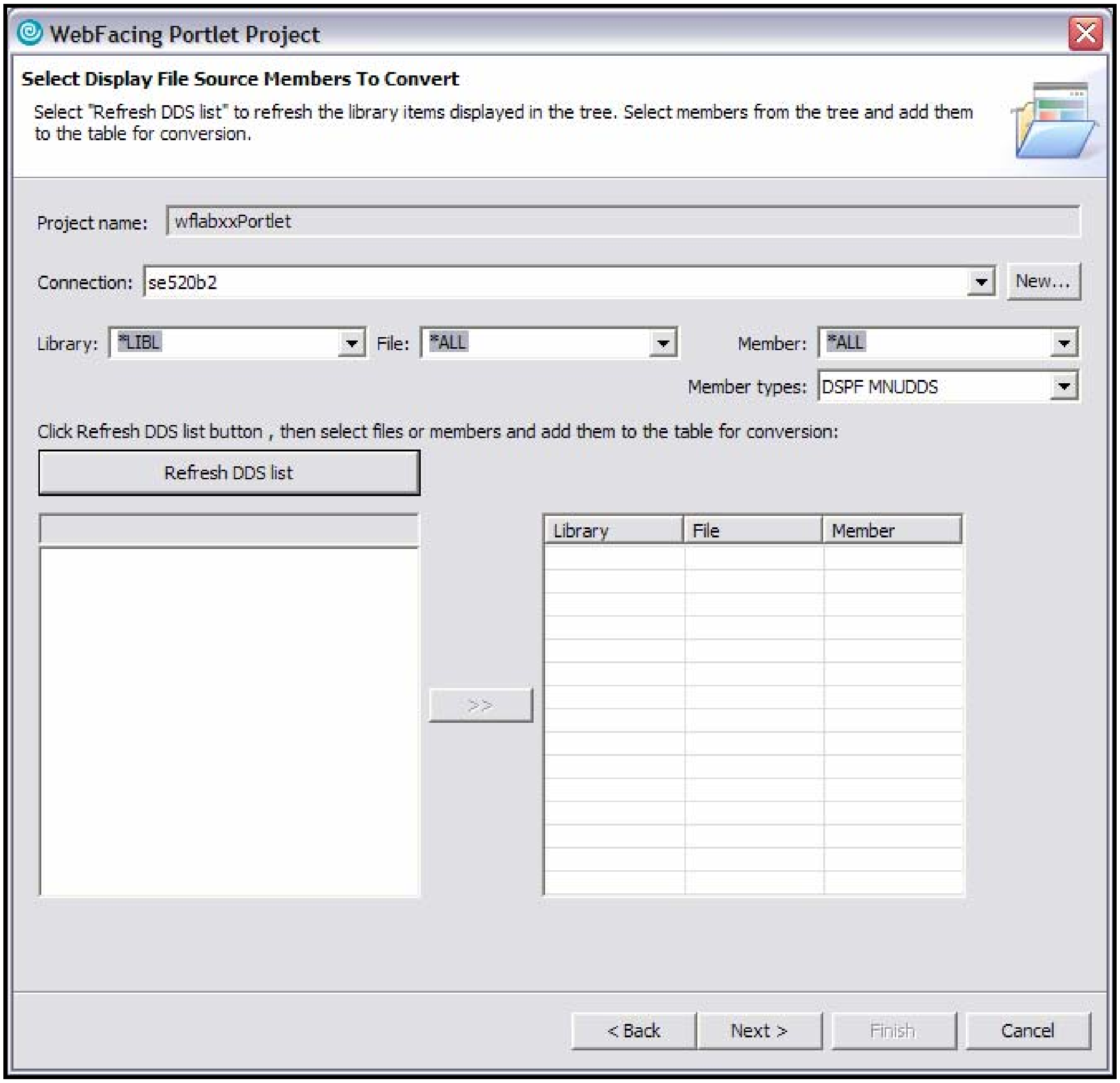
Description:

Use single signon for authentication

*Figure 245. The **WebFacing Features** panel*

10. Click **Next.**

11. The Select Display File Source Members to Convert page opens (see Figure 246).



WebFacing Portlet Project

Select Display File Source Members To Convert

Select "Refresh DDS list" to refresh the library items displayed in the tree. Select members from the tree and add them to the table for conversion.

Project name: wflabxxPortlet

Connection: se520b2

Library: *LIBL

File: *ALL

Member: *ALL

Member types: DSPF MNUDDS

Click Refresh DDS list button , then select files or members and add them to the table for conversion:

Library	File	Member

Figure 246. The Select Display File Source Members to Convert page

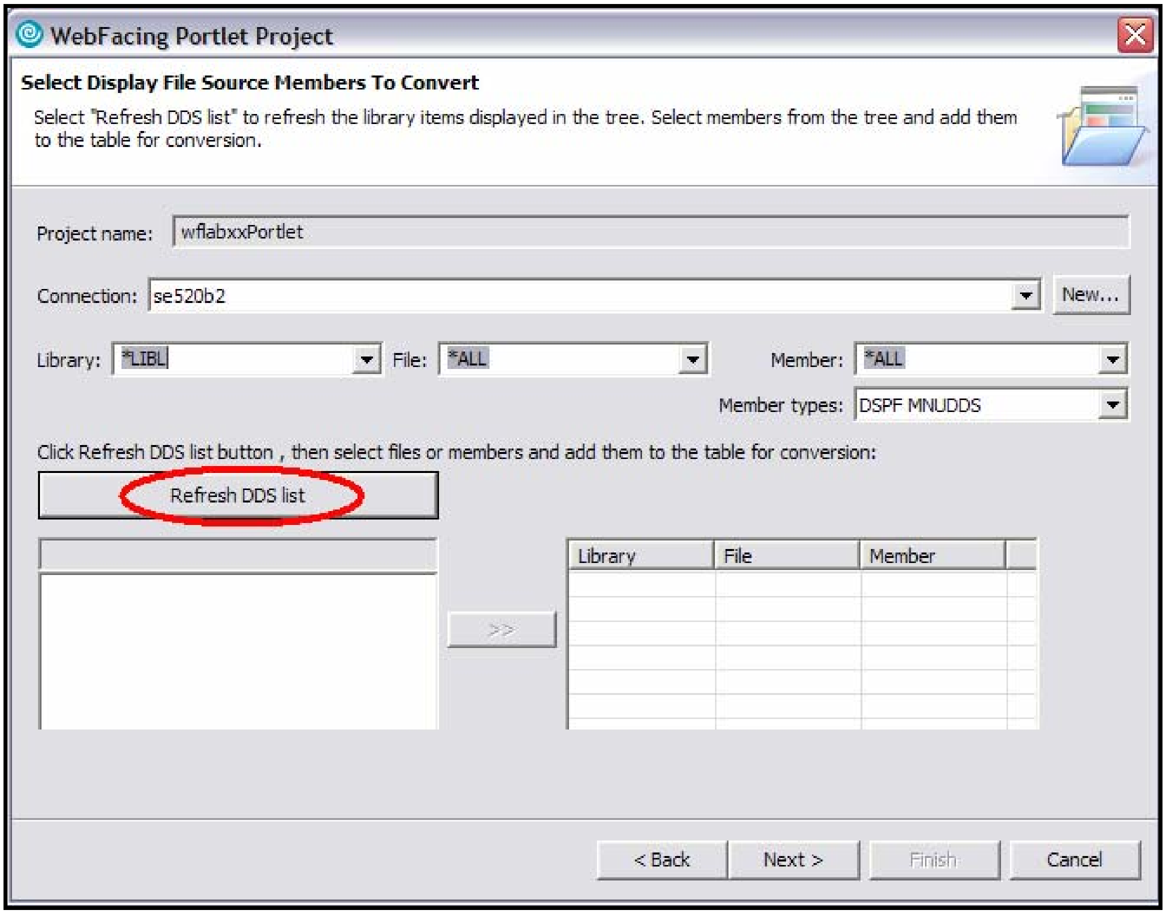
Exercise 14.2: Selecting display file members to convert

On this page of the WebFacing Portlet project wizard, you need to specify the name of the System i that contains your display file DDS source, as well as the members of the source file you want to convert. Specifically, the IBM WebFacing Tool needs to know these names:

- Server name
- Library name
- Source file name
- Member name

1. Ensure ***LIBL** is selected in the **Library** list.

   Click **Refresh DDS list** (see Figure 247).



WebFacing Portlet Project

Select Display File Source Members To Convert

Select "Refresh DDS list" to refresh the library items displayed in the tree. Select members from the tree and add them to the table for conversion.

Project name: wflabxxPortlet

Connection: se520b2

Library: *LIBL

File: *ALL

Member: *ALL

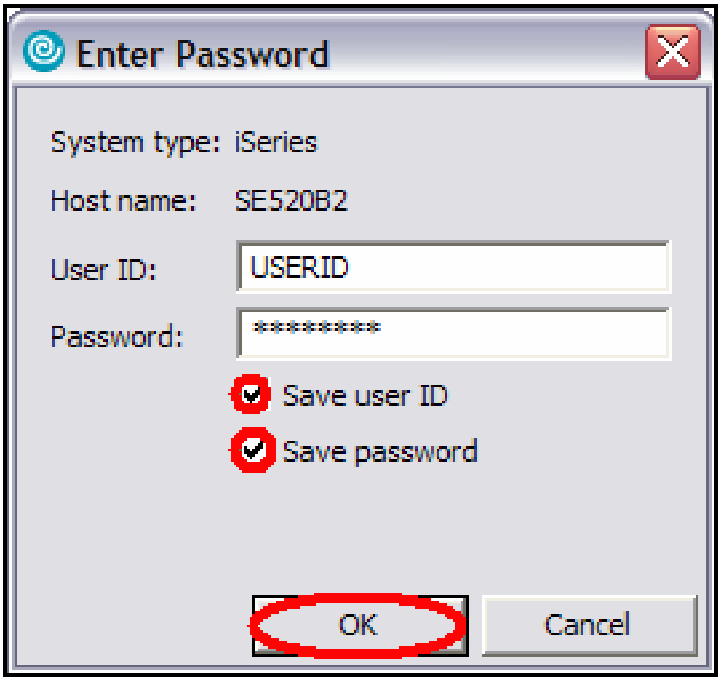
Member types: DSPF MNUDDS

Click Refresh DDS list button , then select files or members and add them to the table for conversion:

Library	File	Member

Figure 247. Click ***Refresh DDS list****.*

If this is the first time you have used this connection, the Enter Password dialog opens (see Figure 248).



System type: iSeries

Host name: SE520B2

User ID: USERID

Password: *********

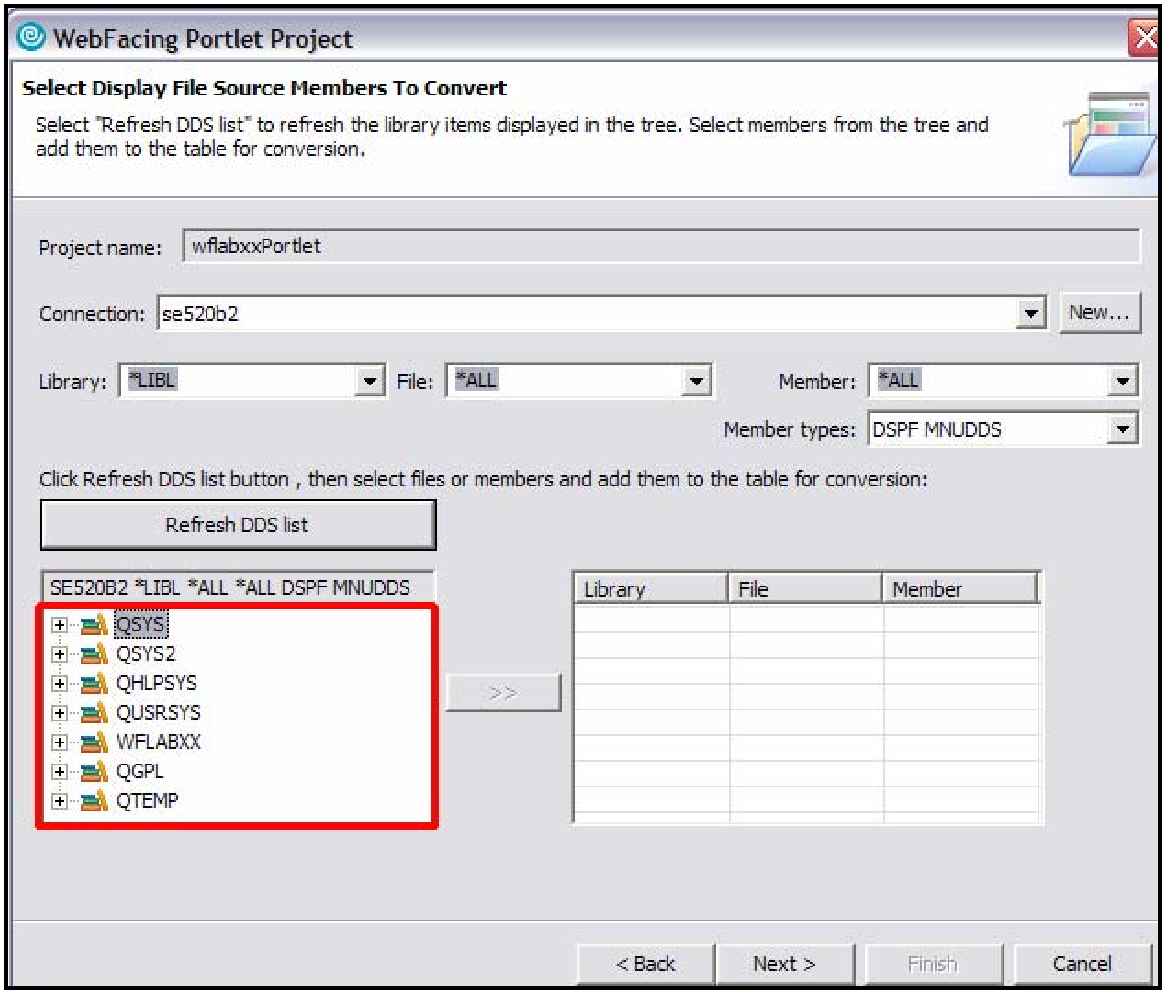
☑ Save user ID

☑ Save password

Figure 248. The Enter Password dialog

2. In the **User ID** field, type your i5OS user ID if required.
3. In the **Password** field, type your i5OS password.
4. Select the **Save user ID** check box if the option is available.
5. Select the **Save password** check box.
6. Click **OK**.

A connection to the System i model is established. The library list of your i5/OS job is displayed under the **Refresh DDS list** button on the Select Display File Source Members to Convert page (see Figure 249).



WebFacing Portlet Project

Select Display File Source Members To Convert

Select "Refresh DDS list" to refresh the library items displayed in the tree. Select members from the tree and add them to the table for conversion.

Project name: wflabxxPortlet

Connection: se520b2

Library: *LIBL File: *ALL Member: *ALL

Member types: DSPF MNUDDS

Click Refresh DDS list button , then select files or members and add them to the table for conversion:

SE520B2 *LIBL *ALL *ALL DSPF MNUDDS

- QSYS
- QSYS2
- QHLPSYS
- QUSRSYS
- WFLABXX
- QGPL
- QTEMP

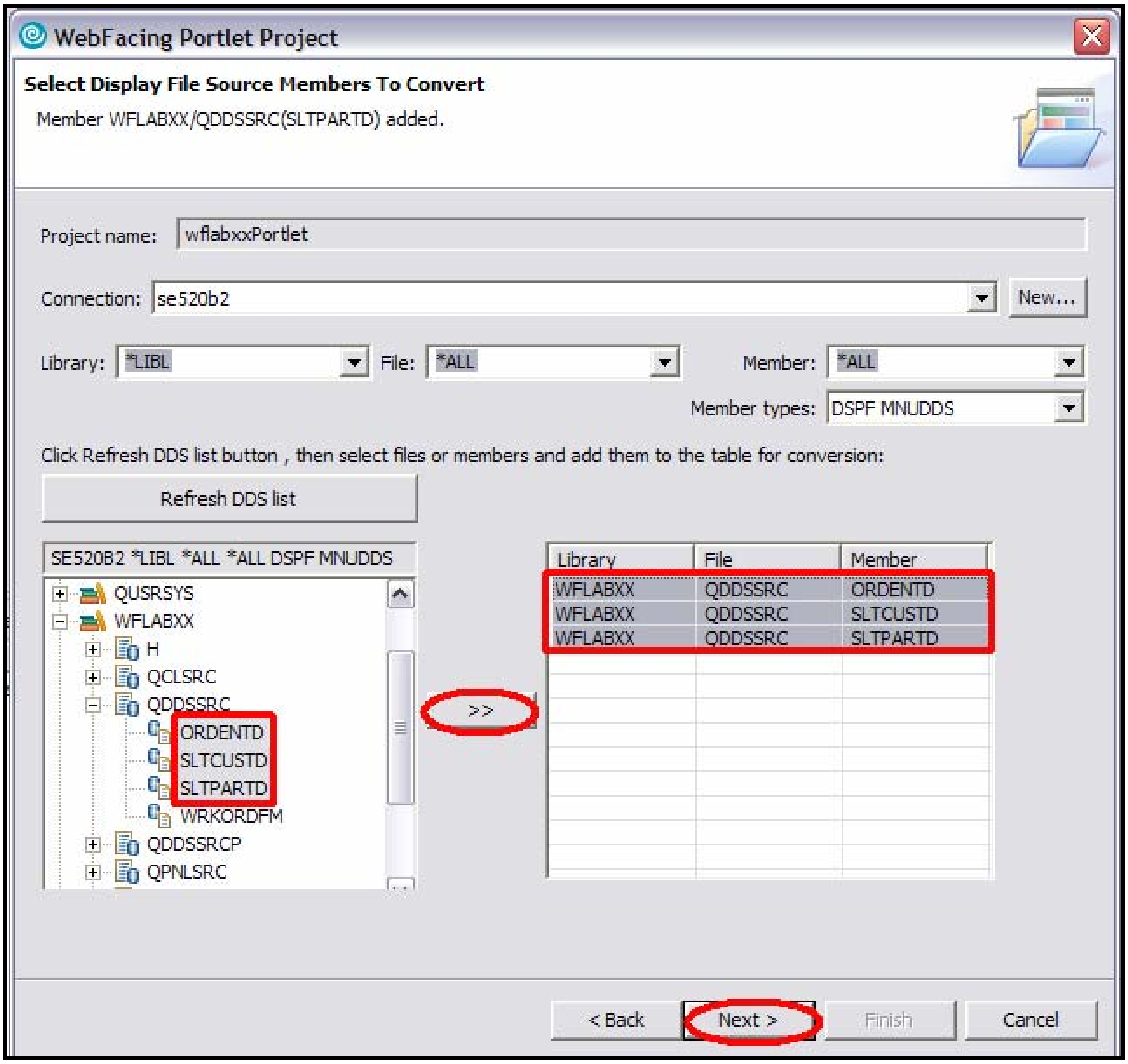
Library	File	Member

*Figure 249. The library list of your i5/OS job is displayed under the **Refresh DDS list** button.*

Selecting DDS members

To select DDS members to convert:

1. Select the **WFLABXX** library from the list.
2. Click the plus sign (**+**) beside the **WFLABXX** library to expand it (see Figure ***250***).



WebFacing Portlet Project

Select Display File Source Members To Convert

Member WFLABXX/QDDSSRC(SLTPARTD) added.

Project name: wflabxxPortlet

Connection: se520b2

Library: *LIBL File: *ALL Member: *ALL

Member types: DSPF MNUDDS

Click Refresh DDS list button , then select files or members and add them to the table for conversion:

SE520B2 *LIBL *ALL *ALL DSPF MNUDDS

Library	File	Member
WFLABXX	QDDSSRC	ORDENTD
WFLABXX	QDDSSRC	SLTCUSTD
WFLABXX	QDDSSRC	SLTPARTD

*Figure 250. Click the plus sign (**+**) beside the WFLABXX library to expand it.*

3. Expand the **QDDSSRC** source file.
4. With the **Ctrl** key pressed on the keyboard, click the members **ORDENTD**, **SLTCUSTD** and **SLTPARTD.**
5. Click >> push button in the middle of the page to copy the selected members over to the list of members to be converted at the right.

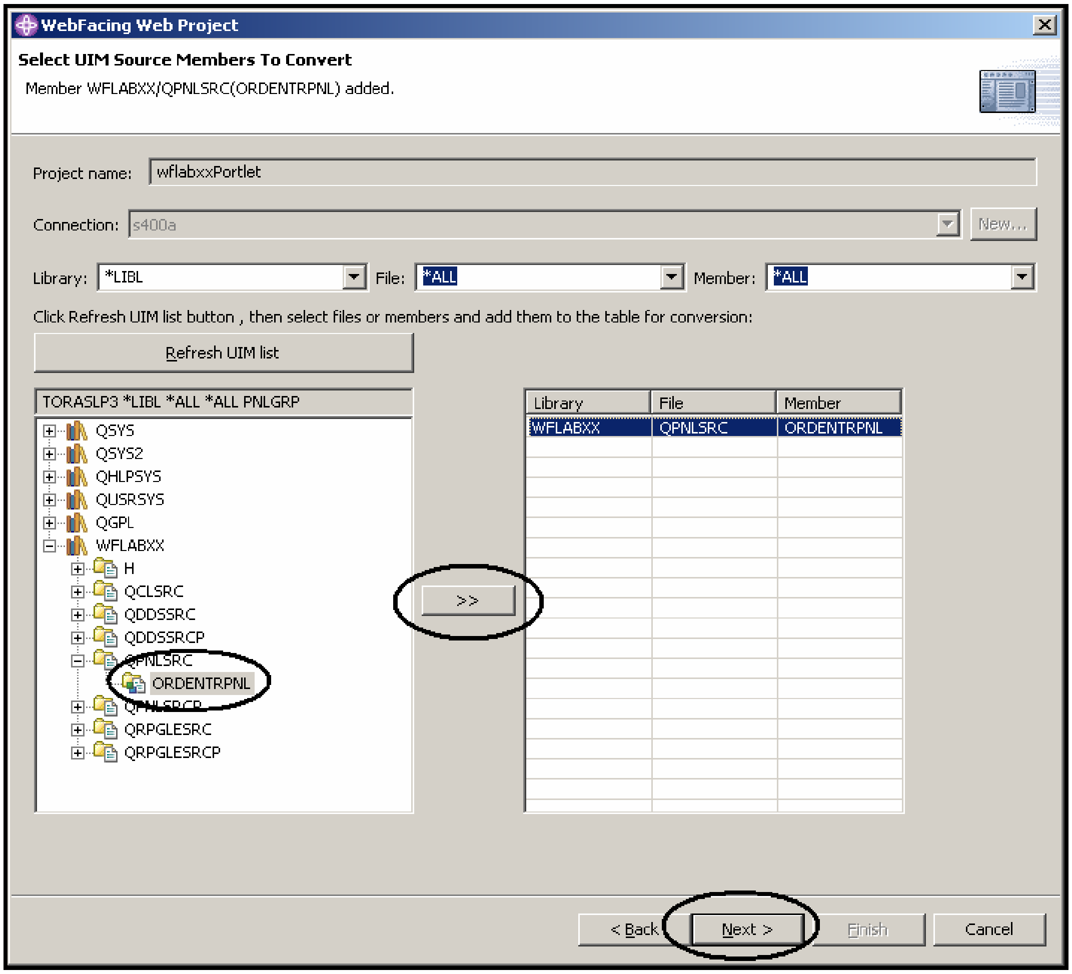
Next, you convert UIM source members.

Selecting UIM source members

To select UIM source members to convert:

1. Click **Next** to proceed to the next page of the wizard.

   The Select UIM Source Members To Convert page opens (see Figure 251).



WebFacing Web Project

Select UIM Source Members To Convert

Member WFLABXX/QPNLSRC(ORDENTRPNL) added.

Project name: wflabxxPortlet

Connection: s400a

Library: *LIBL File: *ALL Member: *ALL

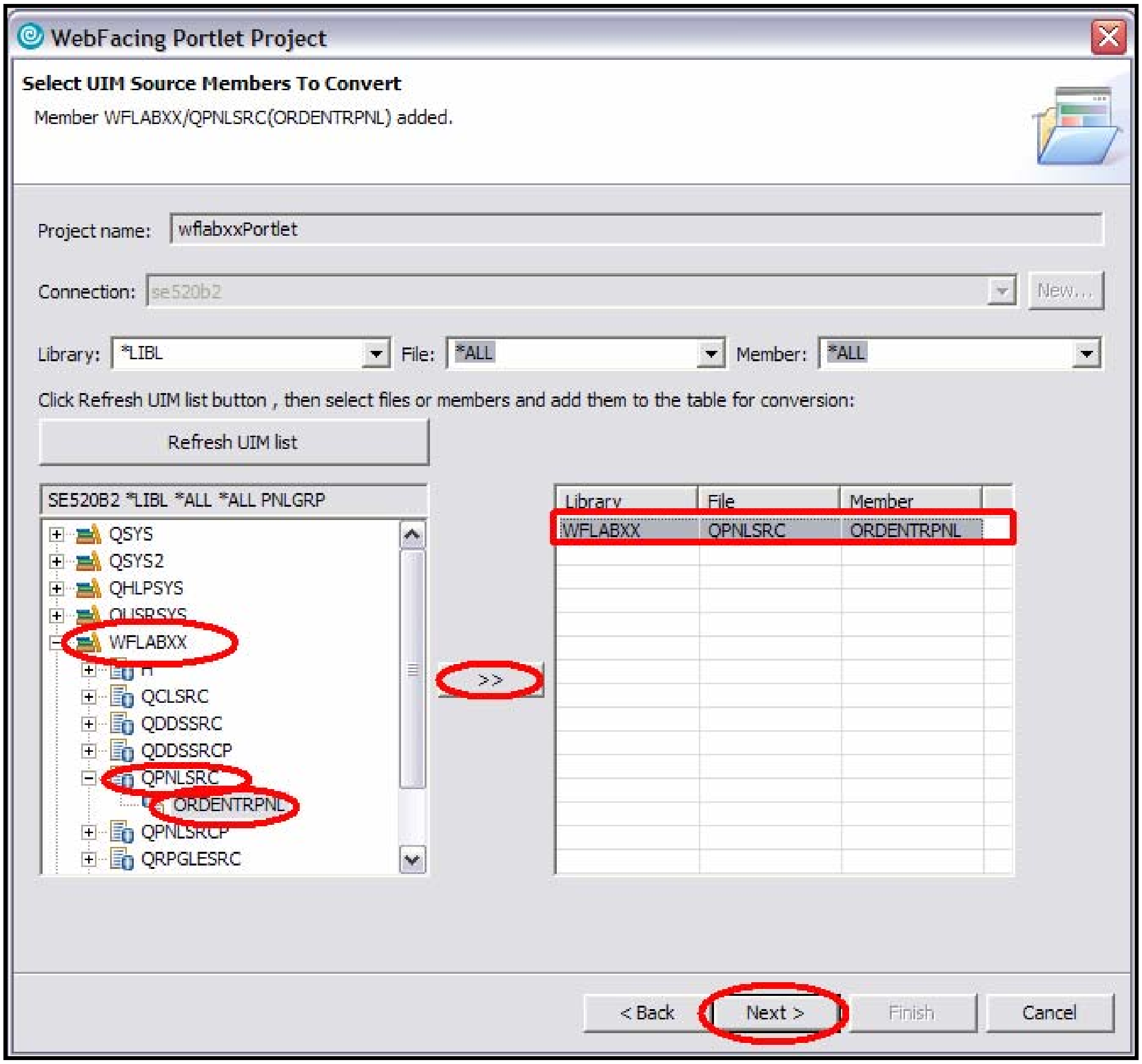
Click Refresh UIM list button , then select files or members and add them to the table for conversion:

TORASLP3 *LIBL *ALL *ALL PNLGRP

Library	File	Member
WFLABXX	QPNLSRC	ORDENTRPNL

Figure 251. Selecting UIM source members

Now, you have to identify the panel group source containing the help information for the Order Entry application (see Figure 252).



WebFacing Portlet Project

Select UIM Source Members To Convert

Member WFLABXX/QPNLSRC(ORDENTRPNL) added.

Project name: wflabxxPortlet

Connection: se520b2

Library: *LIBL File: *ALL Member: *ALL

Click Refresh UIM list button , then select files or members and add them to the table for conversion:

SE520B2 *LIBL *ALL *ALL PNLGRP

- QSYS
- QSYS2
- QHLPSYS
- QUSRSYS
- WFLABXX
  - H
  - QCLSRC
  - QDDSSRC
  - QDDSSRCP
  - QPNLSRC
    - ORDENTRPNL
  - QPNLSRCP
  - QRPGLESRC

Library	File	Member
WFLABXX	QPNLSRC	ORDENTRPNL

Figure 252. The Select UIM Source Members To Convert page

2. Click the **Refresh UIM list** push button.
3. Expand library **WFLABXX**.
4. Select the **QPNLSRC** source file from the expanded list of library **WFLABXX**.
5. Click the >> push button in the middle of the page to copy all members over to the list of members to be converted at the right.

   There is actually only one panel group member **ORDENTRPNL** in this source file.

6. Click **Next** to proceed to the next page of the wizard.

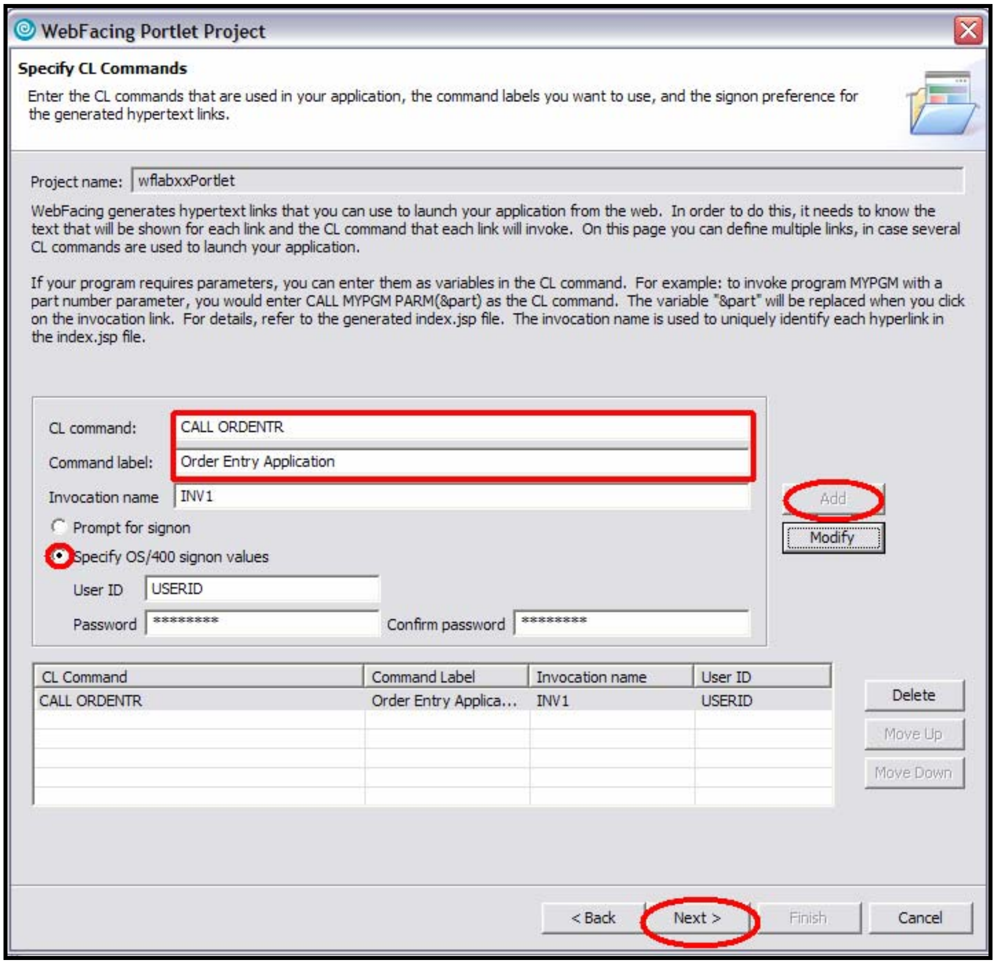
   The Specify CL Commands page opens.

Exercise 14.3: Specifying the CL command to launch the application

You now provide the information that allows the WebFacing Portlet Project wizard to create the initial index.jsp page to start the Order Entry application.

To specify the CL commands (see Figure 253):

1. In the **CL command** field, type **CALL ORDENTR**.
2. In the **Command label** field, type **Order Entry Application**.
3. Leave the default value **INV1** in the **Invocation name** field.
4. Click the **Specify OS/400 signon values** radio button.
5. Click **Add** on the right side of the page.



Specify CL Commands

Enter the CL commands that are used in your application, the command labels you want to use, and the signon preference for the generated hypertext links.

Project name: wflabxxPortlet

WebFacing generates hypertext links that you can use to launch your application from the web. In order to do this, it needs to know the text that will be shown for each link and the CL command that each link will invoke. On this page you can define multiple links, in case several CL commands are used to launch your application.

If your program requires parameters, you can enter them as variables in the CL command. For example: to invoke program MYPGM with a part number parameter, you would enter CALL MYPGM PARM(&part) as the CL command. The variable "&part" will be replaced when you click on the invocation link. For details, refer to the generated index.jsp file. The invocation name is used to uniquely identify each hyperlink in the index.jsp file.

CL command: CALL ORDENTR

Command label: Order Entry Application

Invocation name: INV1

○ Prompt for signon

● Specify OS/400 signon values

User ID: USERID

Password: ********* Confirm password: *********

CL Command	Command Label	Invocation name	User ID
CALL ORDENTR	Order Entry Applica...	INV1	USERID

Figure 253. Specify CL Commands

Note: Make sure the text and command you typed into the fields are actually shown in the CL Command table at the bottom of the page.

6. Click **Next** at the bottom of the wizard page.

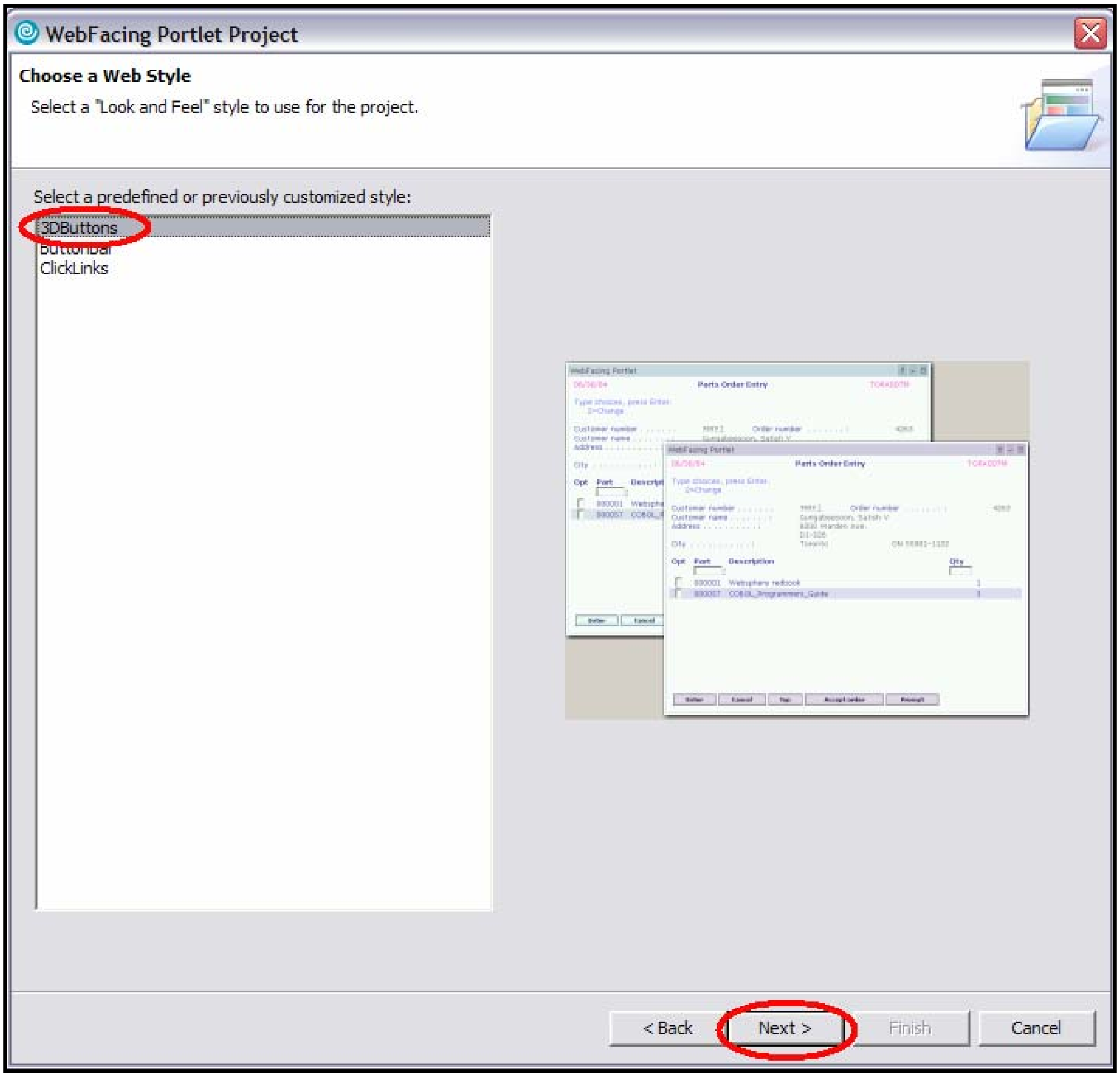
The Choose a Web Style page opens.

Exercise 14.4: Selecting a Web style

Next, select a Web style for your converted screens.

To select a Web style:

1. Select the **3DButtons** style from the list of available styles (see Figure 254).



Choose a Web Style

Select a "Look and Feel" style to use for the project.

Select a predefined or previously customized style:

3DButtons
ButtonBar
ClickLinks

*Figure 254. Select **3DButtons** style*

2. Click **Next**.

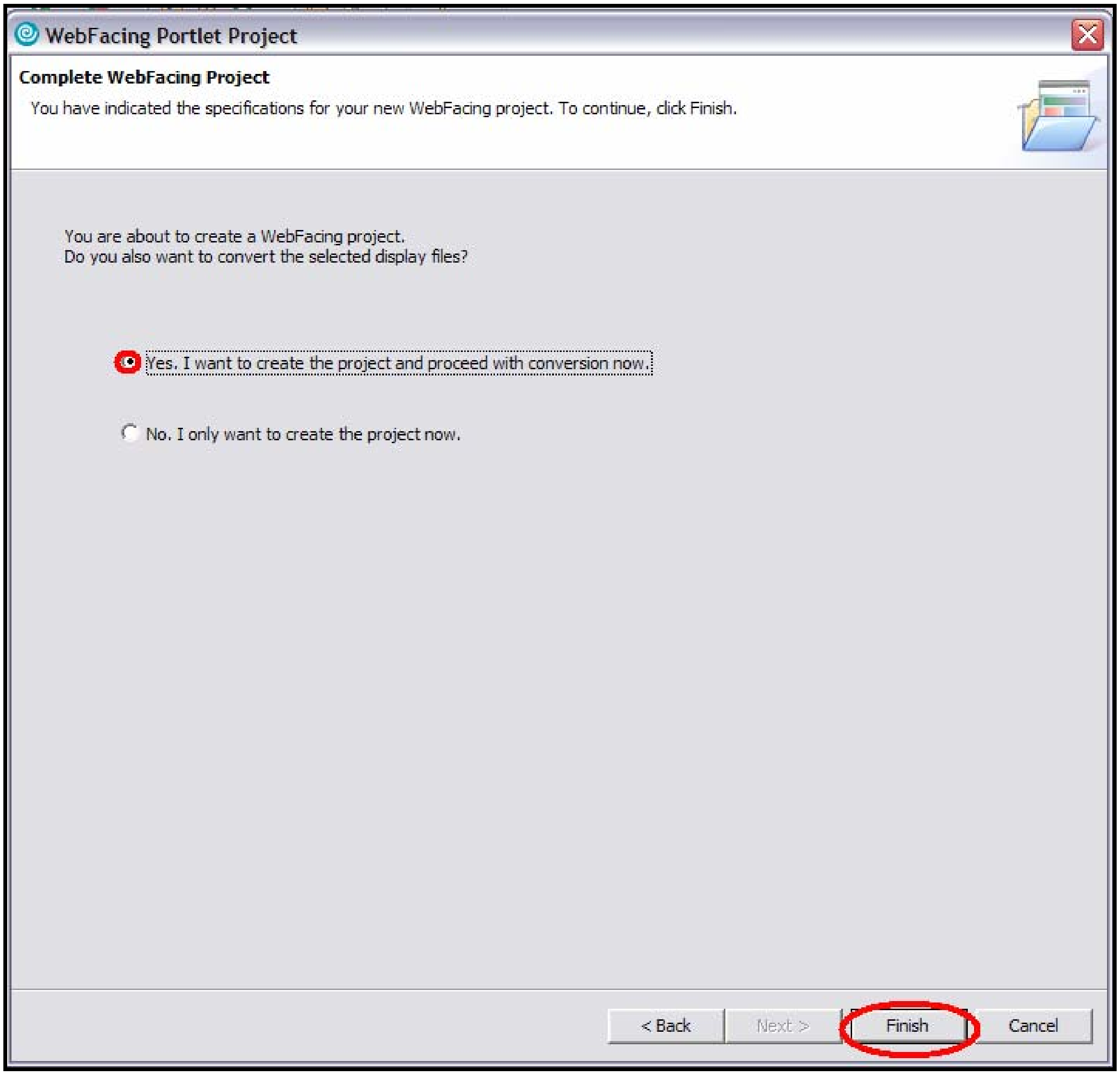
   The Complete WebFacing Project page opens.

Exercise 14.5: Completing the WebFacing project information

On this page, you have a choice of creating the project and converting the source in one step or only creating the WebFacing project.

To complete the WebFacing project information:

1. Select the **Yes. I want to create the project and proceed with conversion now** radio button (see Figure 255).



WebFacing Portlet Project

Complete WebFacing Project

You have indicated the specifications for your new WebFacing project. To continue, click Finish.

You are about to create a WebFacing project.
Do you also want to convert the selected display files?

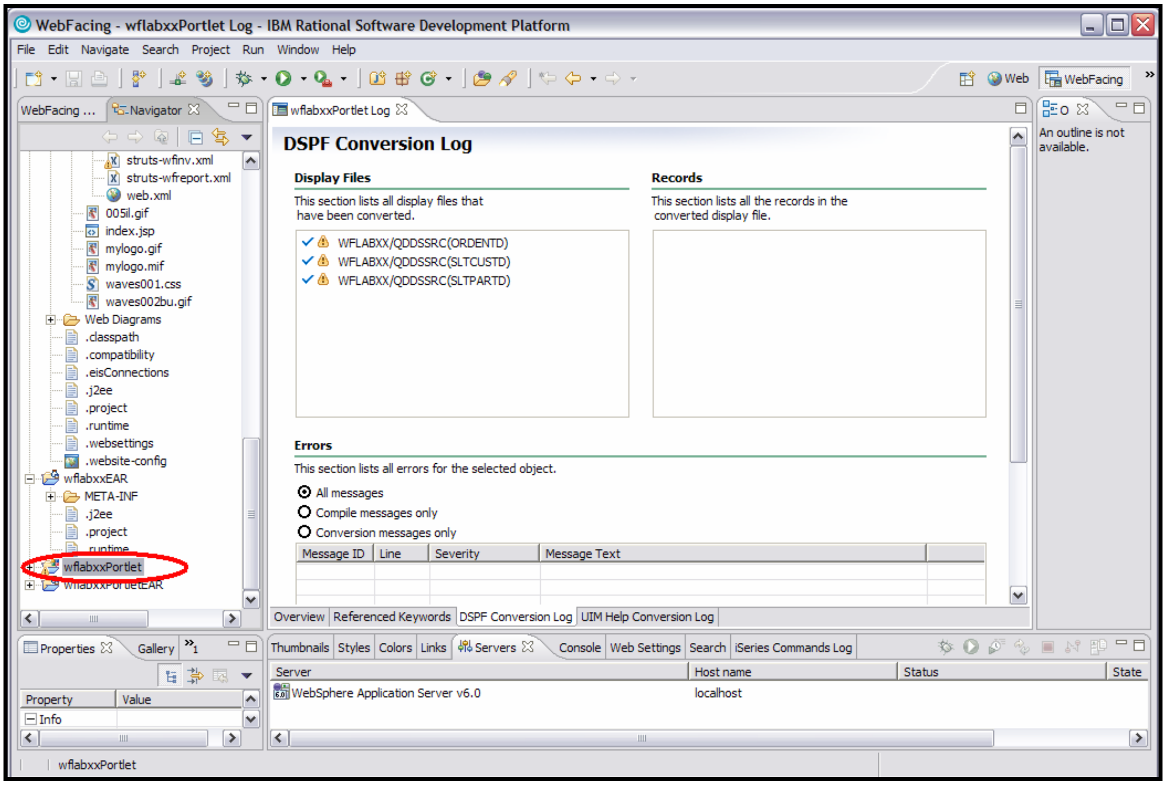
● Yes. I want to create the project and proceed with conversion now.

○ No. I only want to create the project now.

Figure 255. Select the radio button, ***Yes. I want to create the project and proceed with conversion now.***

2. Click **Finish**.

The WebFacing Portlet project is created, and the DDS is converted. The workbench opens with the DSPF Conversion Log in the main panel, and your new WebFacing portlet project highlighted in the WebFacing Navigator view (see Figure 256).



DSPF Conversion Log

Display Files

This section lists all display files that have been converted.

- ✓ ⚠ WFLABXX/QDDSSRC(ORDENTD)
- ✓ ⚠ WFLABXX/QDDSSRC(SLTCUSTD)
- ✓ ⚠ WFLABXX/QDDSSRC(SLTPARTD)

Records

This section lists all the records in the converted display file.

Errors

This section lists all errors for the selected object.

● All messages
○ Compile messages only
○ Conversion messages only

Message ID	Line	Severity	Message Text

Server	Host name	Status	State
WebSphere Application Server v6.0	localhost		

Figure 256. The WebFacing Portlet project is created.

Exercise 14.6: Creating the Portal test server

Before you can test the new WebFacing portlet, you must create a Portal test server. The WebFacing portlet is published to this new server.

To create a Portal test server:

1. If you are currently in the Navigator perspective, switch to the Webfacing Projects perspective. In the WebFacing perspective, click **File > New > Other** from the workbench menu (see
Figure ***257***).

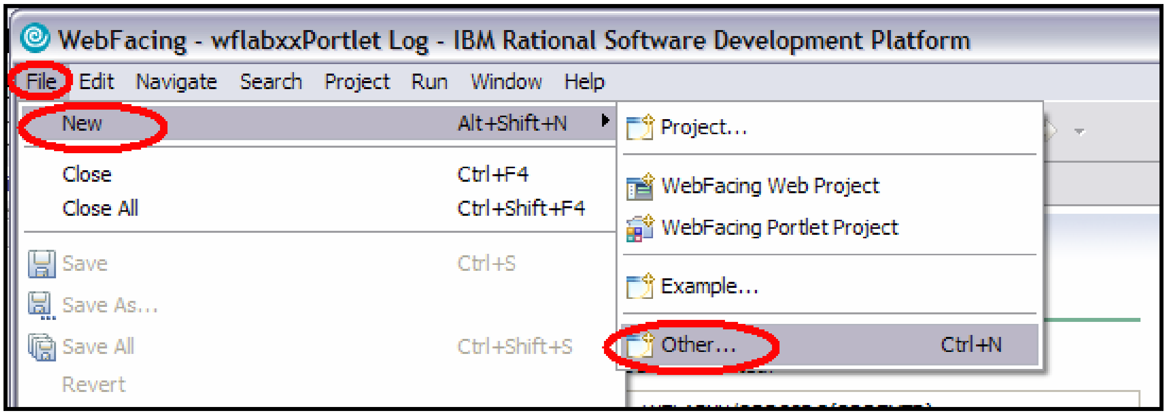
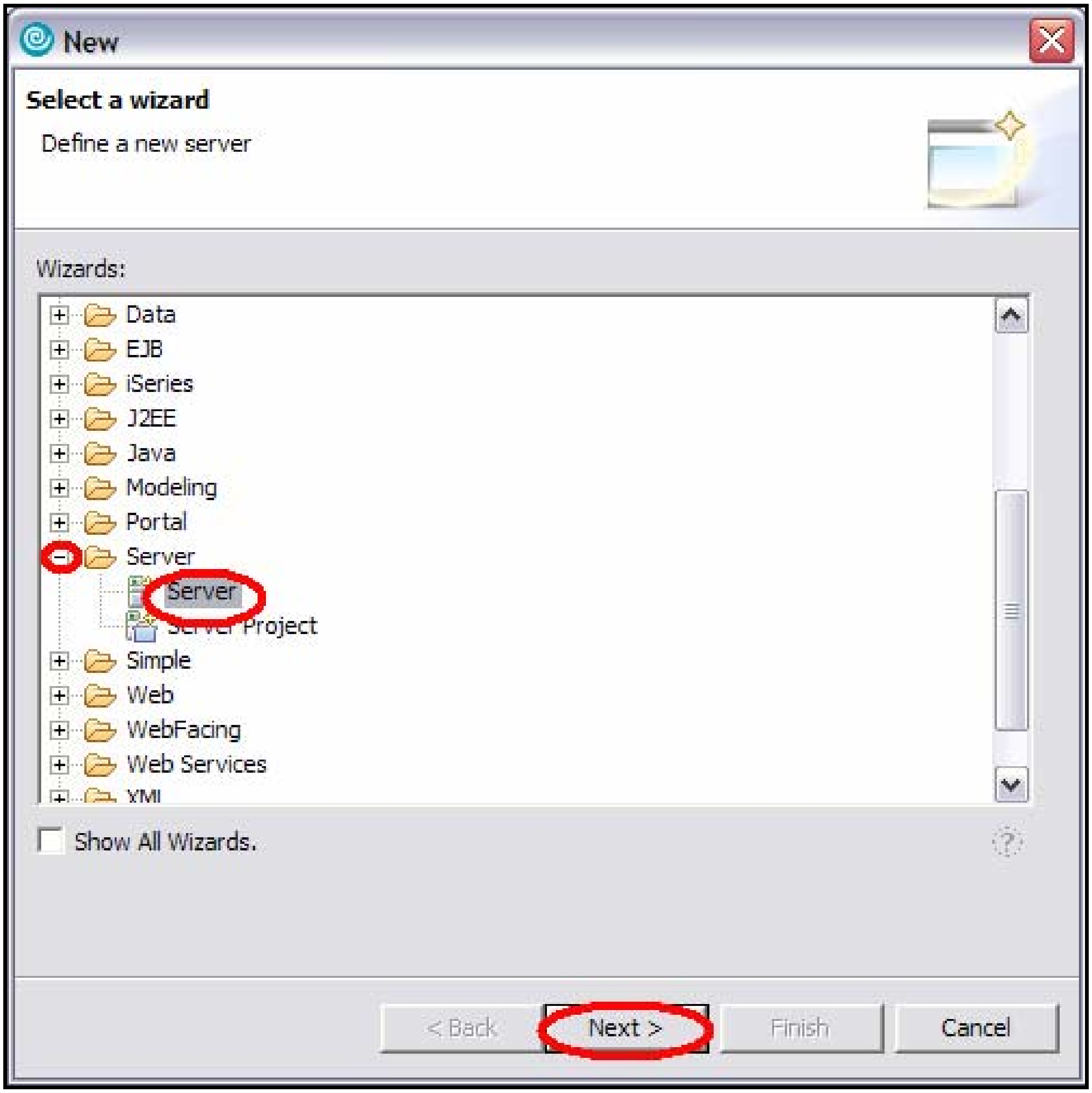


Figure 257. Creating the Portal test server

2. Expand the **Server** section, select **Server**, and click **Next** (see Figure 258).



New

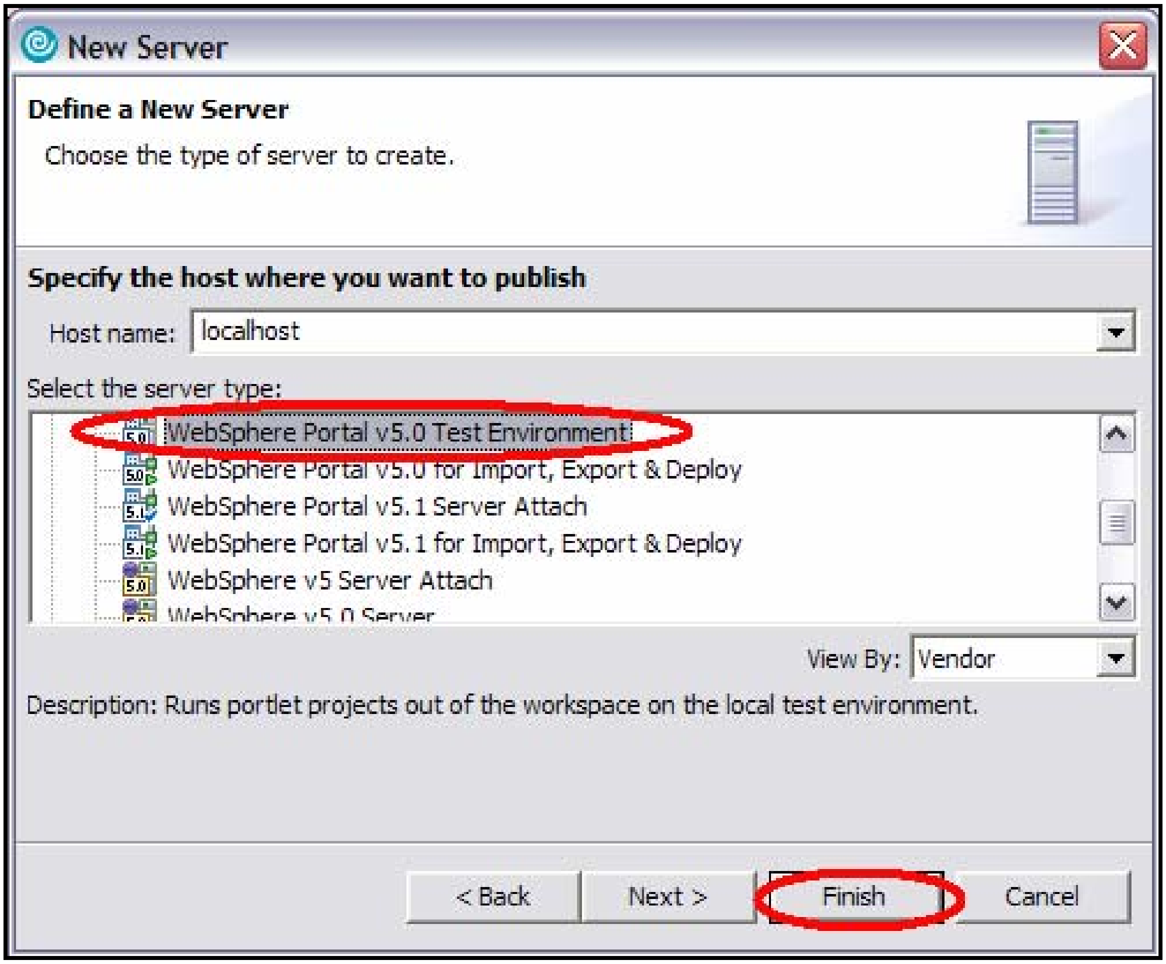
Select a wizard

Define a new server

Wizards:

*Figure 258. Expand the Server section, select **Server**, and click **Next**.*

3. The Define a New Server dialog is displayed (see Figure 259).



New Server

Define a New Server

Choose the type of server to create.

Specify the host where you want to publish

Host name: localhost

Select the server type:

WebSphere Portal v5.0 Test Environment
WebSphere Portal v5.0 for Import, Export & Deploy
WebSphere Portal v5.1 Server Attach
WebSphere Portal v5.1 for Import, Export & Deploy
WebSphere v5 Server Attach
WebSphere v5.0 Server

View By: Vendor

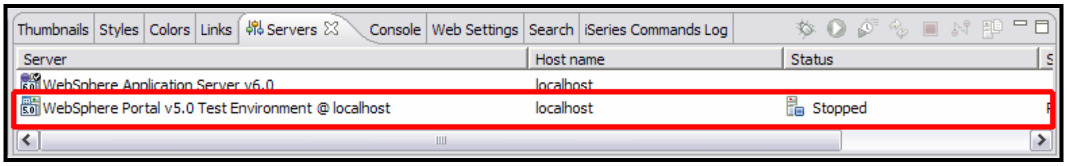
Description: Runs portlet projects out of the workspace on the local test environment.

< Back
Next >
Finish
Cancel

Figure 259. The Define a New Server dialog

4. Select **WebSphere Portal V5.0 test environment**.
5. Click **Finish**.

The new portal server is created in the Servers view (see Figure ***260***).



Server	Host name	Status	S
WebSphere Application Server v6.0	localhost		
WebSphere Portal v5.0 Test Environment @ localhost	localhost	Stopped	R

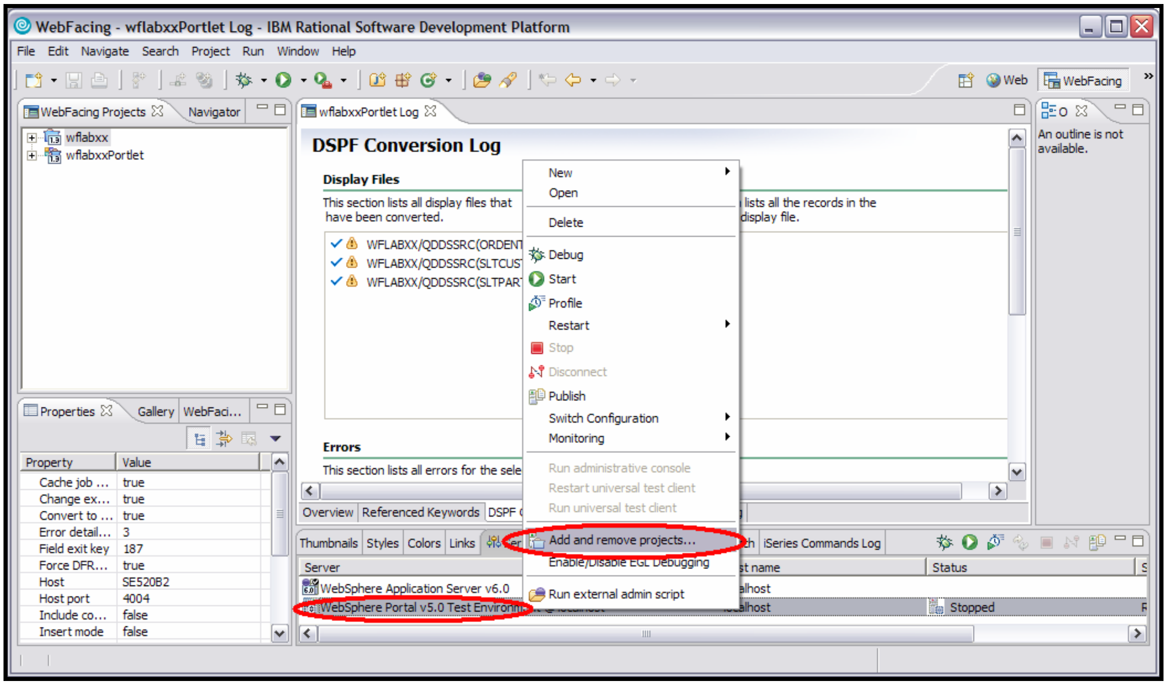
Figure 260. The new portal server is created in the Servers view.

Exercise 14.7: Adding the portlet project to the portal server

Now, you must add the new WebFacing portlet project to the portlet server.

To add the project:

1. Right-click the Portal Server configuration (see Figure 261).


WebFacing - wflabxxPortlet Log - IBM Rational Software Development Platform
File Edit Navigate Search Project Run Window Help
Web
WebFacing
WebFacing Projects
Navigator
wflabxx
wflabxxPortlet
wflabxxPortlet Log
An outline is not available.
Properties
Gallery
WebFaci...
New
Open
Delete
Debug
Start
Profile
Restart
Stop
Disconnect
Publish
Switch Configuration
Monitoring
Run administrative console
Restart universal test client
Run universal test client
Add and remove projects...
Enable/Disable EGL Debugging
Run external admin script
Overview
Referenced Keywords
DSPF
Thumbnails
Styles
Colors
Links
iSeries Commands Log

DSPF Conversion Log

Display Files

This section lists all display files that have been converted.

... lists all the records in the display file.

✓ WFLABXX/QDDSSRC(ORDENT
✓ WFLABXX/QDDSSRC(SLTCUST
✓ WFLABXX/QDDSSRC(SLTPART

Errors

This section lists all errors for the sele

Property	Value
Cache job ...	true
Change ex...	true
Convert to ...	true
Error detail...	3
Field exit key	187
Force DFR...	true
Host	SE520B2
Host port	4004
Include co...	false
Insert mode	false

Server	...st name	Status
WebSphere Application Server v6.0	...alhost	
WebSphere Portal v5.0 Test Environm...	...alhost	Stopped

Figure 261. Adding the portlet project to the portlet server

2. From the pop-up menu, select **Add and remove projects.**

3. The Add and Remove Projects dialog is displayed (see Figure ***262***).

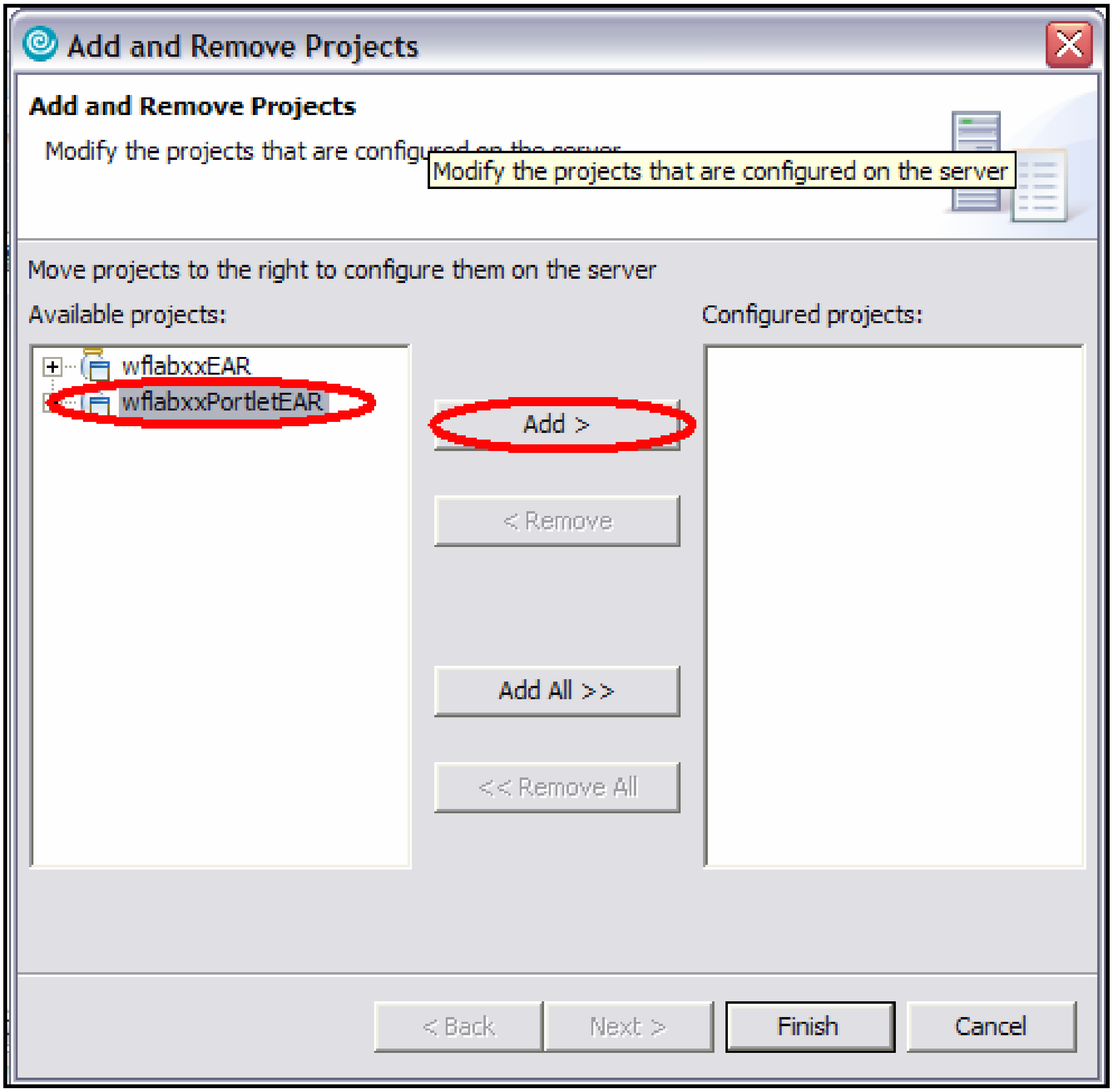
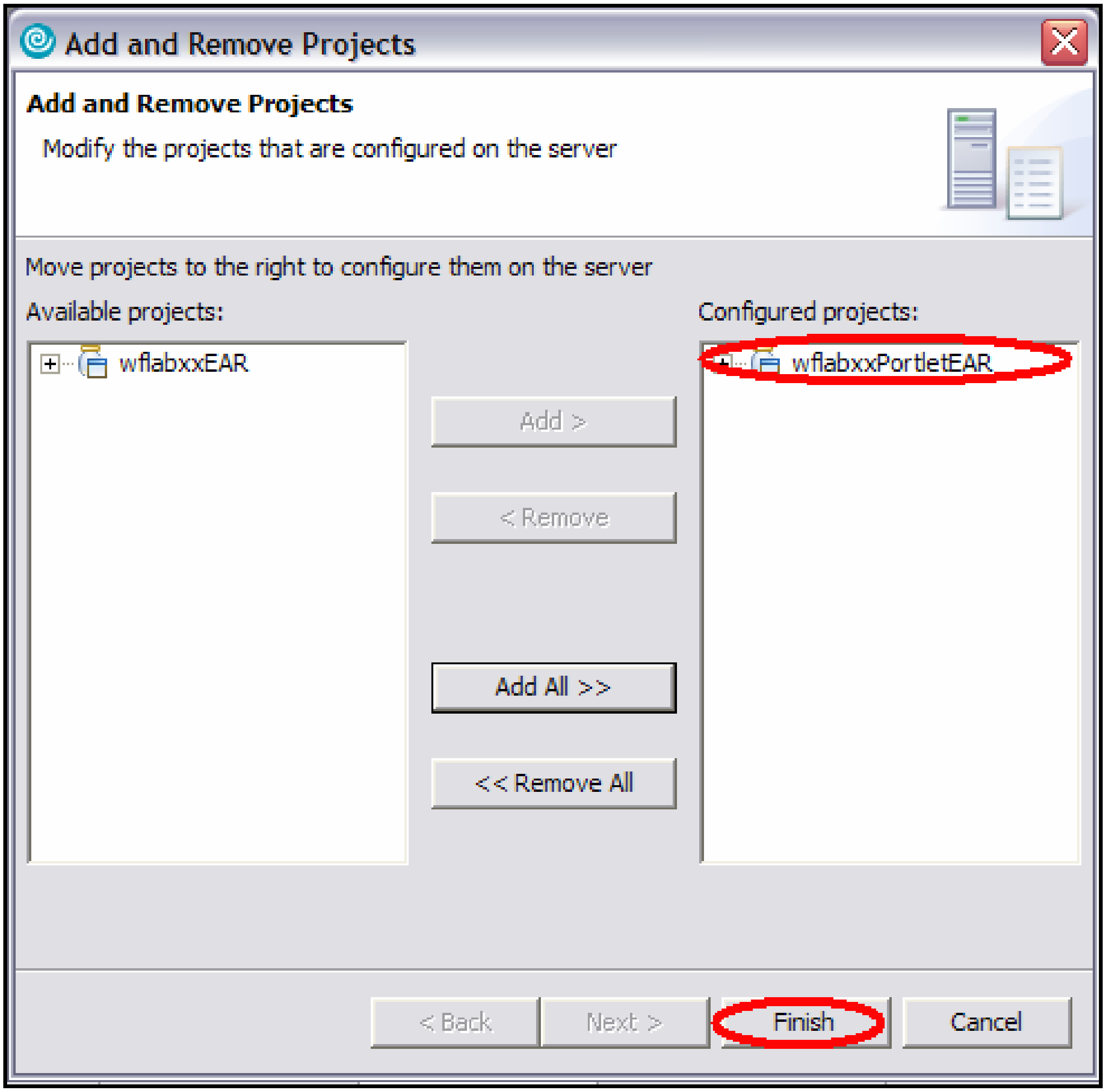


Figure 262. The Add and Remove Projects dialog

4. In the left pane, select the **wflabxxPortletEAR** project.
5. Click **Add** .
6. The project appears in the right pane (see Figure ***263***).



Add and Remove Projects

Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Available projects:

- wflabxxEAR

Configured projects:

- wflabxxPortletEAR

Figure 263. Configured projects appear in the right pane.

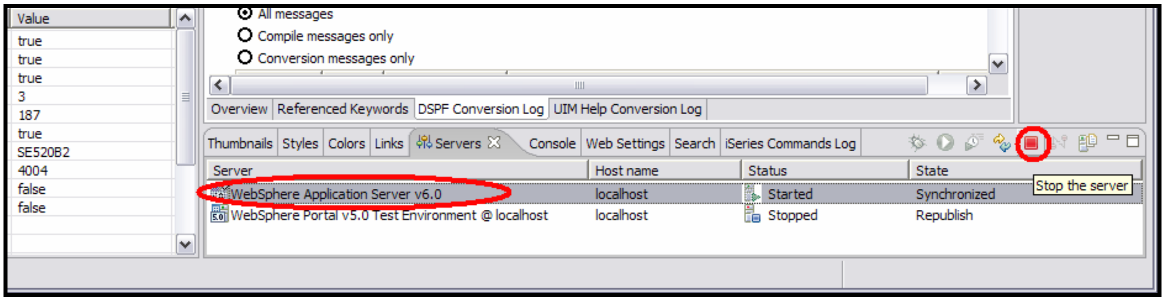
7. Click **Finish**. The project is added to the test portal server.

Exercise 14.8: Starting the portal server

To test the WebFacing portlet application, you must start the portal server. However, before you do so, you must ensure no other servers are running. This is due to the fact, that by default, the test servers share the same port numbers.

To stop any running servers:

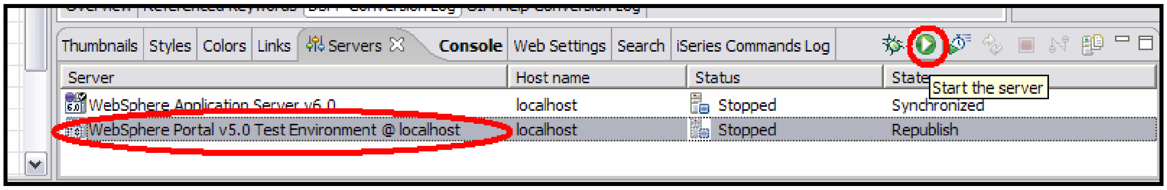
1. Click the **Server** tab. This is usually located in the lower right corner of the workbench (see
Figure ***264***).



Server	Host name	Status	State
WebSphere Application Server v6.0	localhost	Started	Synchronized
WebSphere Portal v5.0 Test Environment @ localhost	localhost	Stopped	Republish

*Figure 264. Click the **Server** tab.*

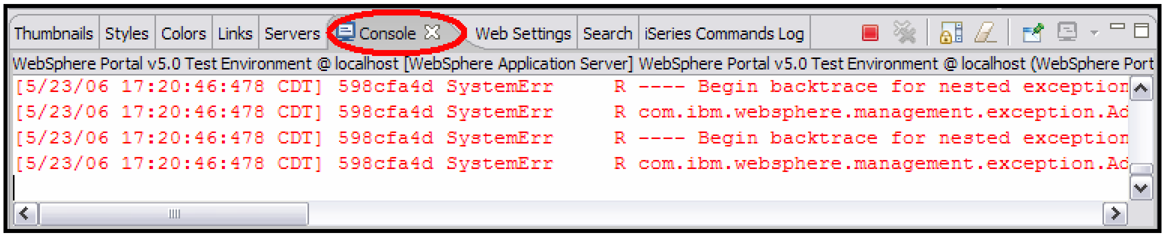
2. If a server is shown with a status of **Started**, it must be stopped.
3. Click the running server to select it, and then click the Stop the Server icon ■. The server Status changes to Stopped (see
Figure ***265***).



Server	Host name	Status	State
WebSphere Application Server v6.0	localhost	Stopped	Synchronized
WebSphere Portal v5.0 Test Environment @ localhost	localhost	Stopped	Republish

Figure 265. The server Status changes to Stopped.

4. Click the **WebSphere Portal v5.0 Test Environment @ localhost** server, then click the Start the Server icon .The view switches to the **Console** tab view (see Figure 266).

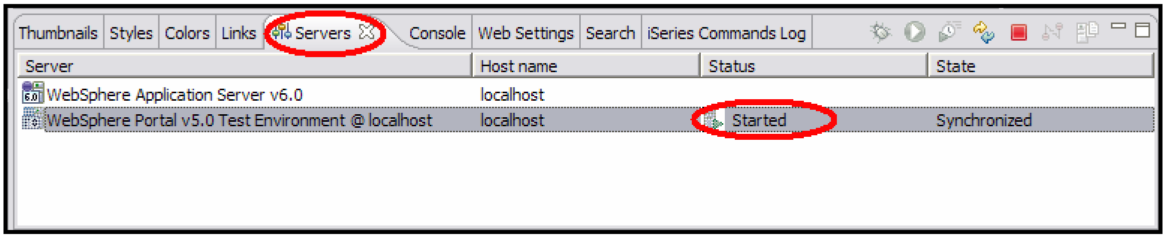


WebSphere Portal v5.0 Test Environment @ localhost [WebSphere Application Server] WebSphere Portal v5.0 Test Environment @ localhost (WebSphere Port

```
[5/23/06 17:20:46:478 CDT] 598cfa4d SystemErr     R ---- Begin backtrace for nested exception
[5/23/06 17:20:46:478 CDT] 598cfa4d SystemErr     R com.ibm.websphere.management.exception.Ad
[5/23/06 17:20:46:478 CDT] 598cfa4d SystemErr     R ---- Begin backtrace for nested exception
[5/23/06 17:20:46:478 CDT] 598cfa4d SystemErr     R com.ibm.websphere.management.exception.Ad
```

Figure 266. The Console tab view.

5. Switch back to the **Server** tab. If the startup of the portal test server completes successfully, the Status goes to **Started** (see Figure 267)



Server	Host name	Status	State
WebSphere Application Server v6.0	localhost		
WebSphere Portal v5.0 Test Environment @ localhost	localhost	Started	Synchronized

Figure 267. If the startup of the portal test server completes successfully, the Status goes to Started.

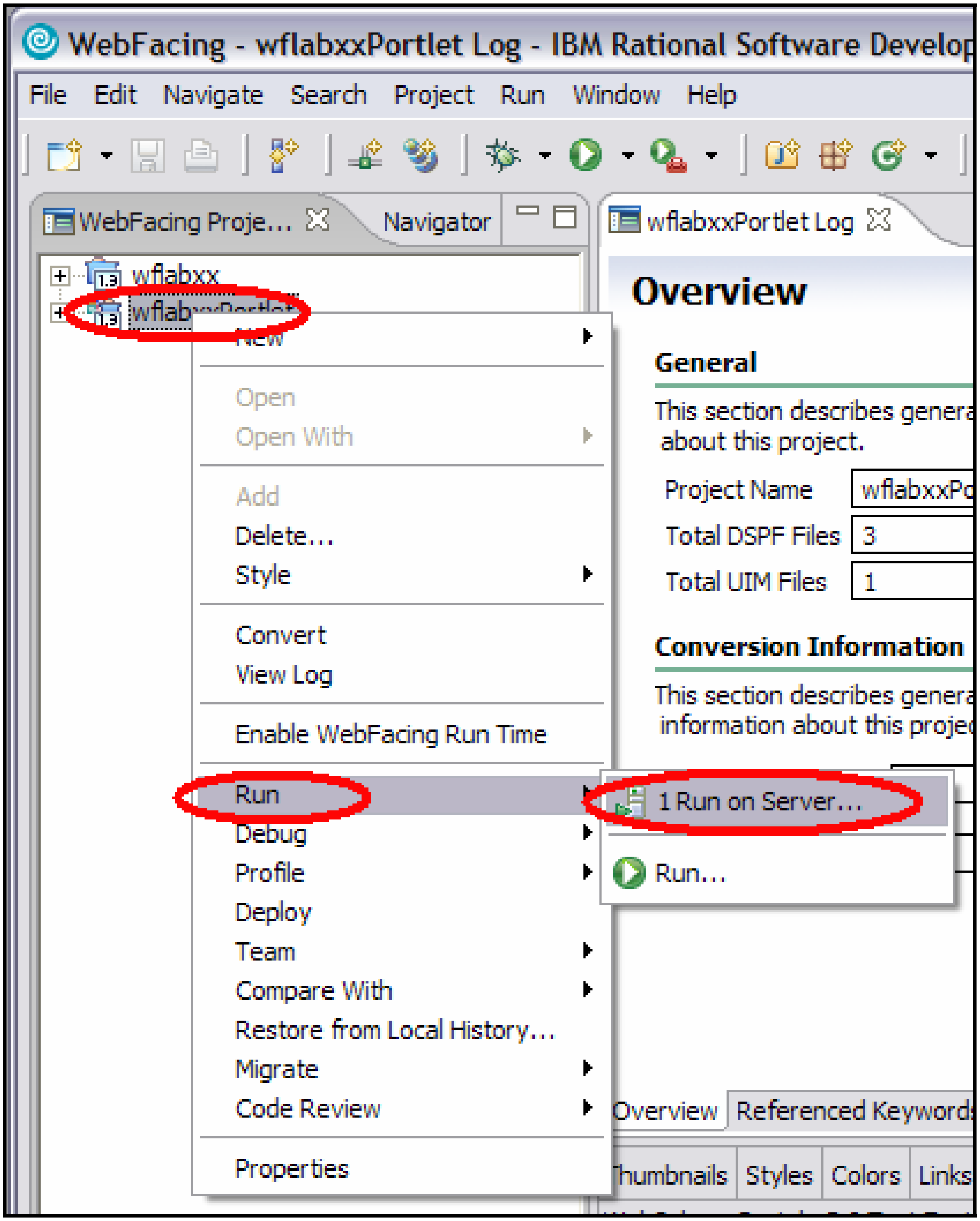
Exercise 14.9: Testing the WebFacing portlet application

Now that the portal server has been configured, the portlet has been published to the portal server, and the portal server is started, you can test the WebFacing portlet.

To test the portlet:

1. In the WebFacing view, select the WebFacing projects tab.

2. The WebFacing projects are displayed (see Figure 268).



This section describes genera
about this project.

Project Name: wflabxxPo

Total DSPF Files: 3

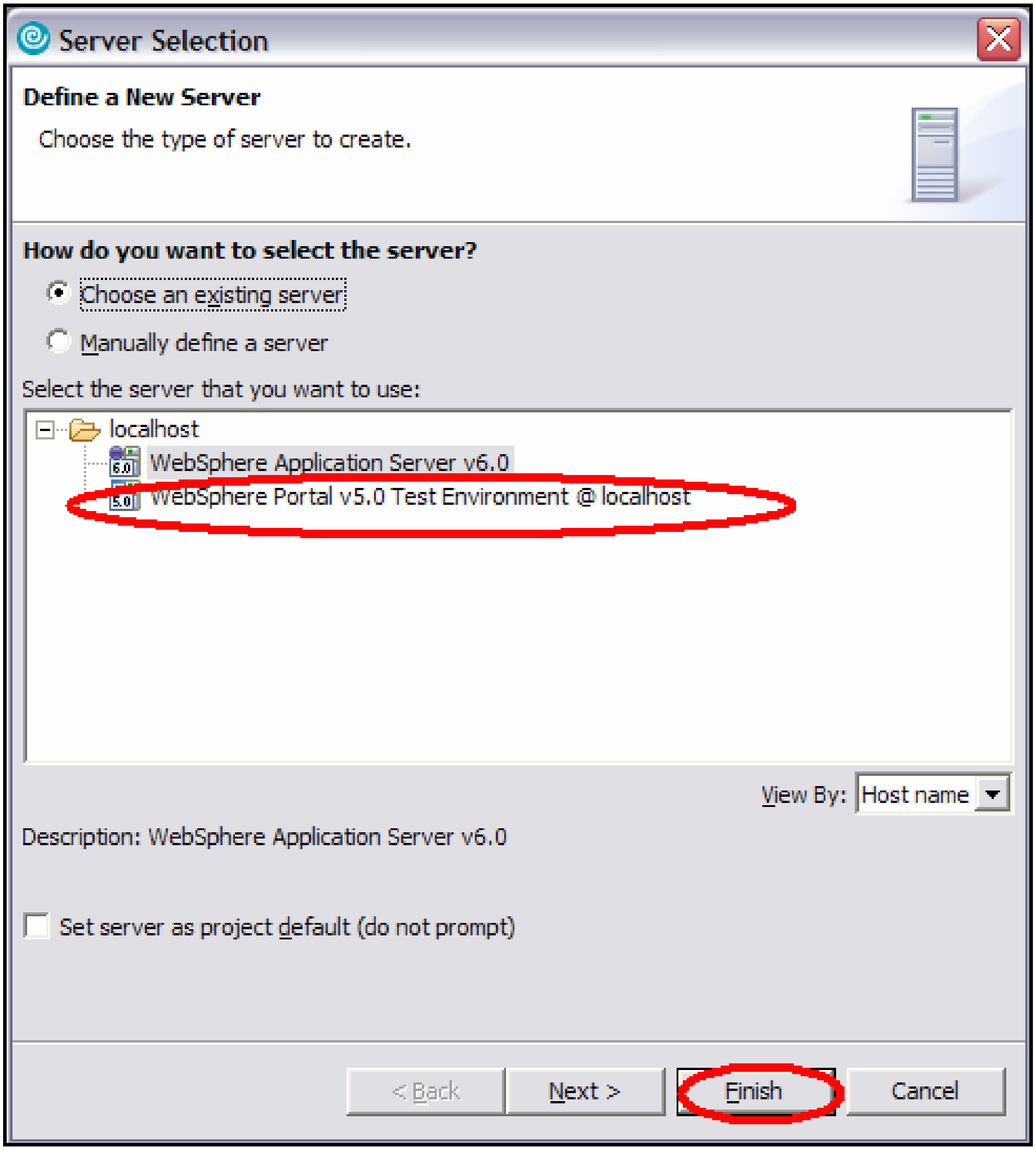
Total UIM Files: 1

This section describes genera
information about this projec

Figure 268. The WebFacing projects are displayed.

3. Right-click the **wflabxxPortlet** project, and select **Run>Run on Server…** from the pop-up menu.

4. The Server Selection dialog is displayed (see Figure 269).



Server Selection

Define a New Server

Choose the type of server to create.

How do you want to select the server?

● Choose an existing server

○ Manually define a server

Select the server that you want to use:

- localhost
  - WebSphere Application Server v6.0
  - WebSphere Portal v5.0 Test Environment @ localhost

View By: Host name

Description: WebSphere Application Server v6.0

□ Set server as project default (do not prompt)

Figure 269. The Server Selection dialog

5. Select **Choose an existing server** if not already selected.
6. Click the **WebSphere Portal v5.0 Test Environment @ localhost** server.
7. Click **Finish**.

8. The index.jsp page of the project is displayed in the internal browser (see Figure 270).

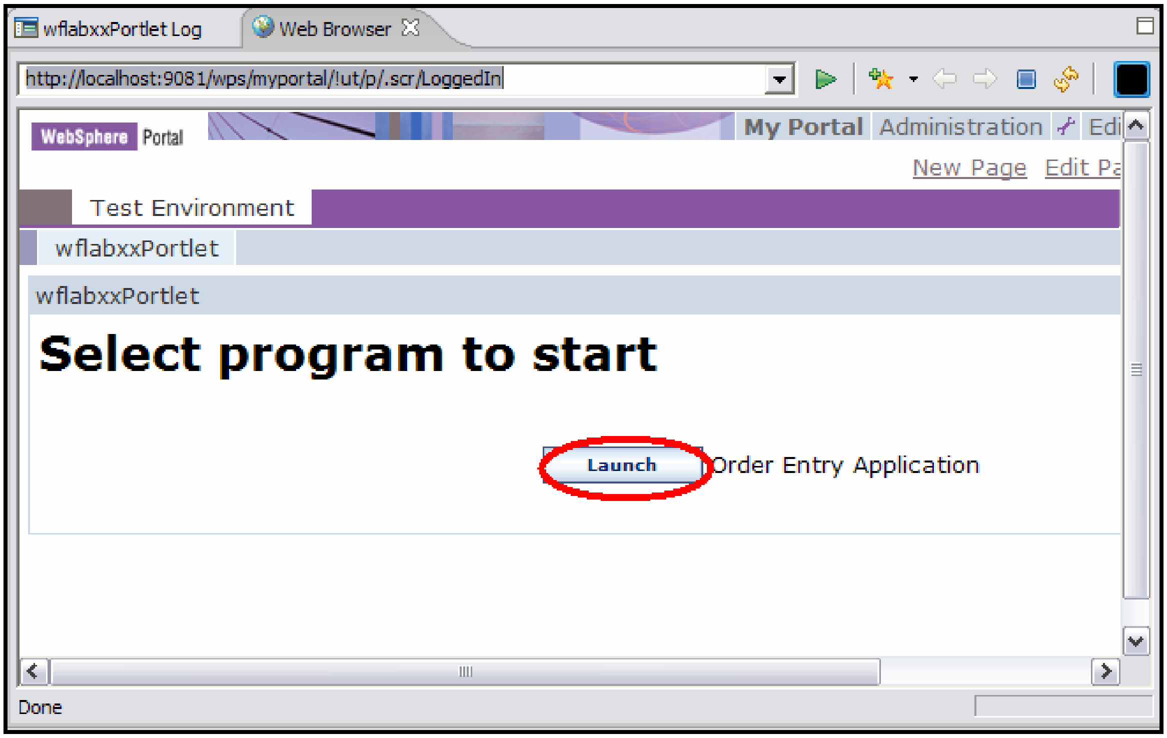
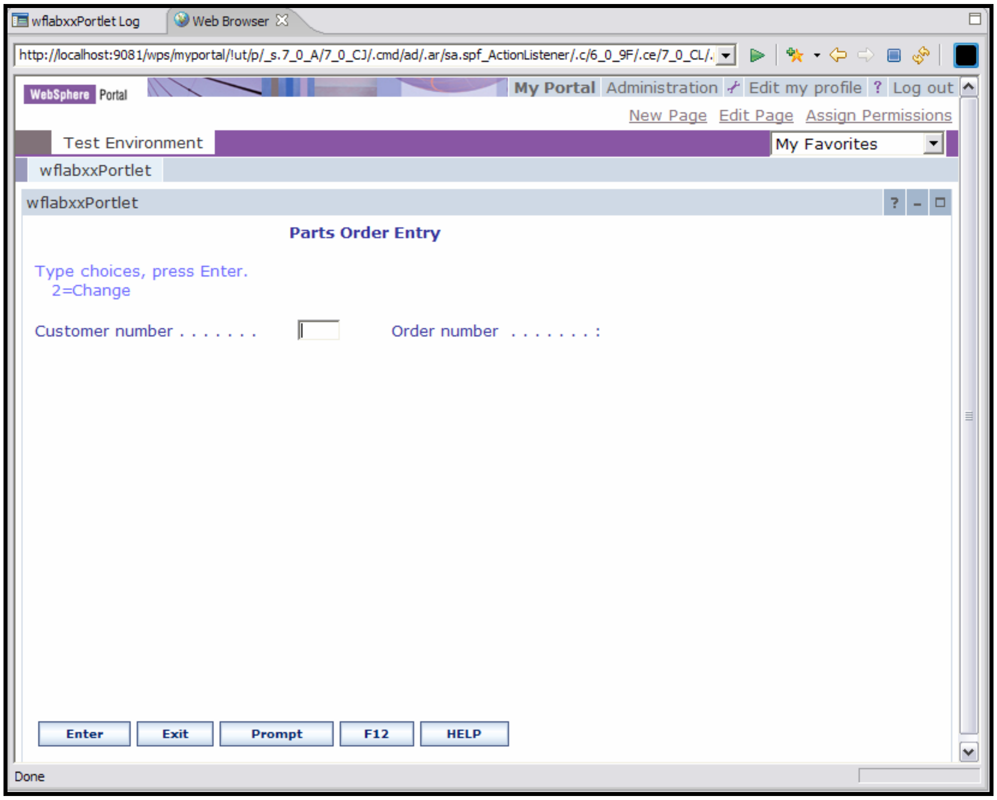


Figure 270. The index.jsp page of the project is displayed in the internal browser.

9. Click the **Launch** push button to start the application.

10. The first page of the Order Entry application is displayed (see Figure ***271***).



http://localhost:9081/wps/myportal/!ut/p/_s.7_0_A/7_0_CJ/.cmd/ad/.ar/sa.spf_ActionListener/.c/6_0_9F/.ce/7_0_CL/.

Parts Order Entry

Type choices, press Enter.
  2=Change

Customer number . . . .   Order number . . . . :

Figure 271. The first page of the Order Entry application

At this point, the application runs the same as it did in the test environment. Press **F4** to prompt for a customer number and exercise the application to see its behavior in the portal environment.

Recap

You have completed “Creating the WebFacing Portlet project.” You now have the information to understand how to:

- Launch the WebFacing Portlet project wizard
- Complete the pages of this wizard to create a new WebFacing Portlet project called wflabxxPortlet
- Add to this WebFacing Portlet project the following information:
    - Which DDS display file source members to use for the WebFacing conversion
    - Which UIM panel group source members to convert
    - The CL command that is used to create the initial index.jsp page to start the Order Entry application
    - The Web style to be used for the resulting Web pages
- Create a test portal server
- Deploy the WebFacing portlet to the test portal server
- Start the test portal server
- Test the WebFacing portlet in the test portal server

Testing system screen support

In this chapter, you learn how to test system screen support in a WebFaced application. System screen support enables your WebFaced application to display i5/OS system screens, such as Work with All Spooled Files (WRKSPLF), in the browser window.

Note: If you have experience with previous versions of WebSphere Development Studio Client Advanced Edition, notice that you no longer need to enable system screen support for your WebFacing Project when using WebSphere Development Studio Client V6.0.

In this chapter, you convert an existing sample 5250 application that displays the Work with All Spooled Files (WRKSPLF) screen when a command key is pressed.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 15.1: Running the sample 5250 application
- Exercise 15.2: Creating a new WebFacing application with system screen support
- Exercise 15.3: Test the application with system screen support.

The exercises within each chapter must be completed in order. Start with “Exercise 15.1: Reviewing the 5250 sample application.”

Length of time

This chapter takes approximately 20 minutes to complete.

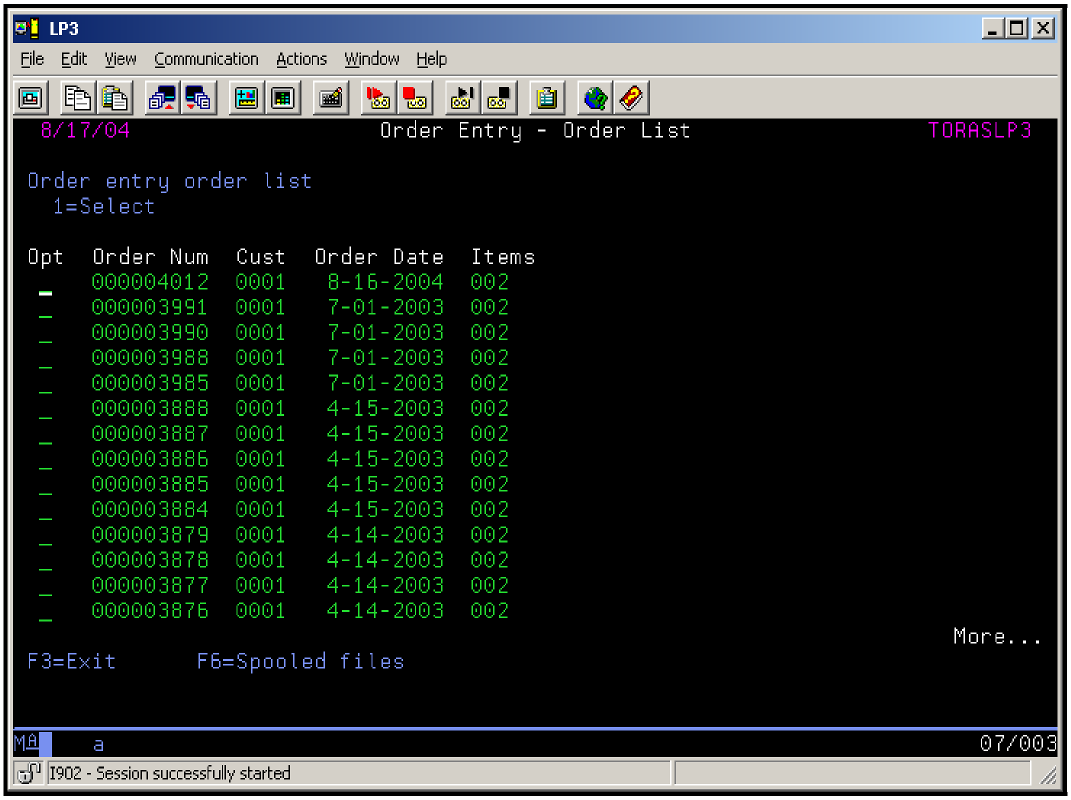
Exercise 15.1: Reviewing the 5250 sample application

You need a 5250 emulator on your workstation to start this sample application. The display for this application uses a subfile to display the orders created by the Order Entry application. Pressing **F6=Spooled files** on this display invokes the WRKSPLF system command to display the spooled files.

To run the 5250 application:

1. Start a 5250 emulation session.
2. In the **User ID** field, type your User ID.
3. In the **Password** field, type your password.
4. On the command line of the 5250 screen, invoke the sample application: CALL WRKORD.

   The application starts and displays the Order List screen (see Figure 272).



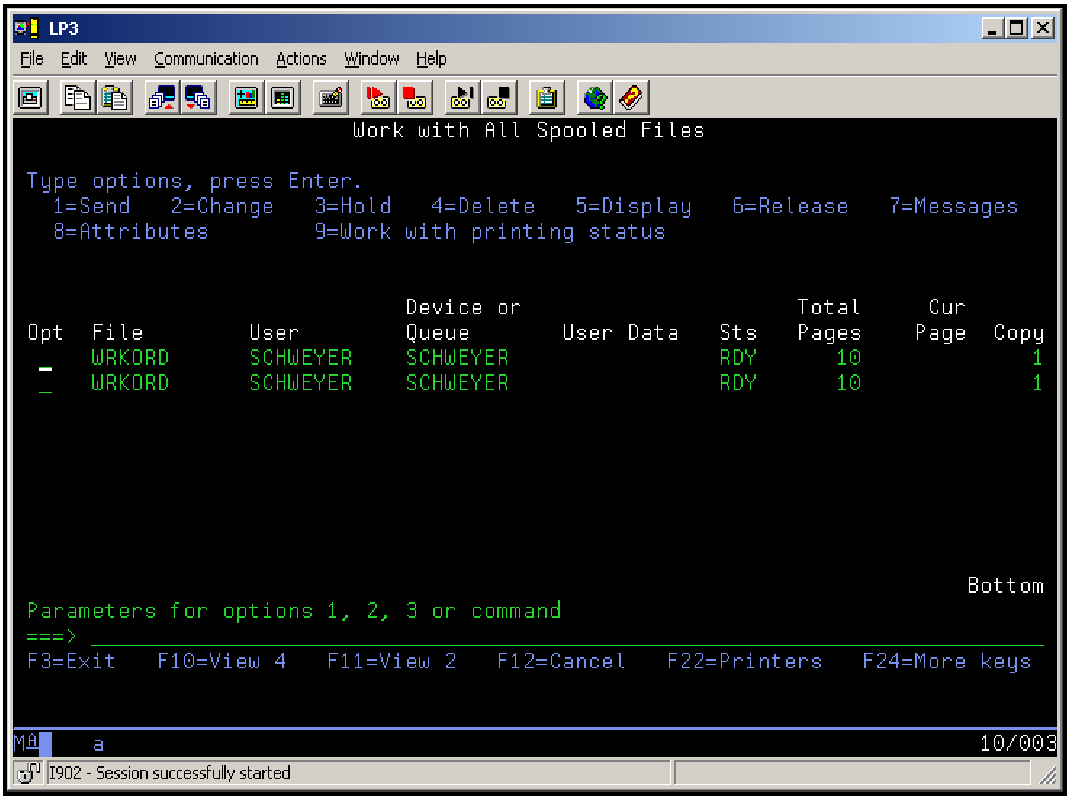
```
 8/17/04                       Order Entry - Order List                 TORASLP3

Order entry order list
  1=Select

Opt  Order Num  Cust   Order Date  Items
 _   000004012  0001    8-16-2004  002
 _   000003991  0001    7-01-2003  002
 _   000003990  0001    7-01-2003  002
 _   000003988  0001    7-01-2003  002
 _   000003985  0001    7-01-2003  002
 _   000003888  0001   4-15-2003   002
 _   000003887  0001   4-15-2003   002
 _   000003886  0001   4-15-2003   002
 _   000003885  0001   4-15-2003   002
 _   000003884  0001   4-15-2003   002
 _   000003879  0001   4-14-2003   002
 _   000003878  0001   4-14-2003   002
 _   000003877  0001   4-14-2003   002
 _   000003876  0001   4-14-2003   002
                                                                        More...
F3=Exit      F6=Spooled files

MA     a                                                                07/003
```

I902 - Session successfully started

Figure 272. The Order List screen

5. Note that this program does not support any operations on the orders listed. It is just used for this example. You can use the page keys to scroll through the orders.

6. Press **F6=Spooled files**. This causes the program to use QCMDEXC to invoke the **WRKSPLF** command (see Figure 273).



```
                          Work with All Spooled Files

 Type options, press Enter.
   1=Send   2=Change   3=Hold   4=Delete   5=Display   6=Release   7=Messages
   8=Attributes        9=Work with printing status

                               Device or                     Total     Cur
 Opt  File        User        Queue       User Data   Sts    Pages    Page  Copy
  _   WRKORD      SCHWEYER    SCHWEYER                RDY       10             1
  _   WRKORD      SCHWEYER    SCHWEYER                RDY       10             1







                                                                           Bottom
 Parameters for options 1, 2, 3 or command
 ===> ______________________________________________________________________________
 F3=Exit   F10=View 4   F11=View 2   F12=Cancel   F22=Printers   F24=More keys

MA     a                                                                   10/003
I902 - Session successfully started
```

*Figure 273. The program uses QCMDEXC to invoke the **WRKSPLF** command.*

7. Press **F3** to exit the Work with All Spooled Files screen.
8. Press **F3=Exit** to end the sample application.

In the next exercise, you create a new WebFacing project and reface this application. During the project creation, WebSphere Development Studio Client automatically enables the WRKSPLF system screen support that is invoked by the WRKORD application.

Exercise 15.2: Creating a new WebFacing project with system screen support

If WebSphere Development Studio Client is not started, start it now and switch to the WebFacing project.

To create a new WebFacing project:

1. In the WebFacing perspective, click **File > New > WebFacing Web Project** from the workbench menu (see
Figure ***274***).

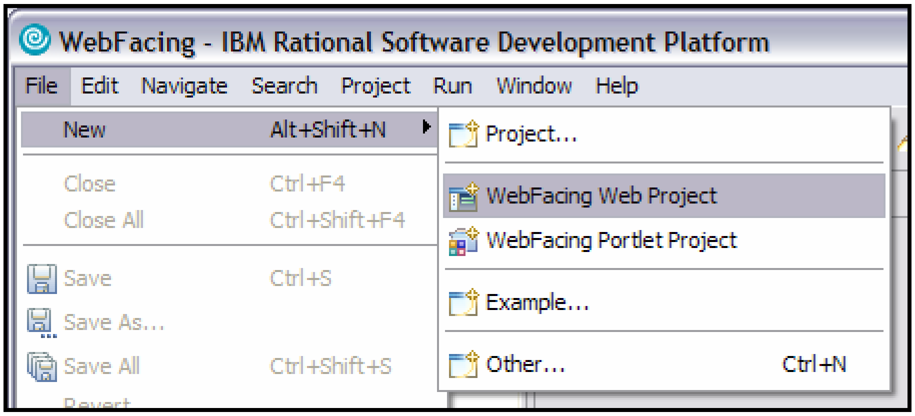
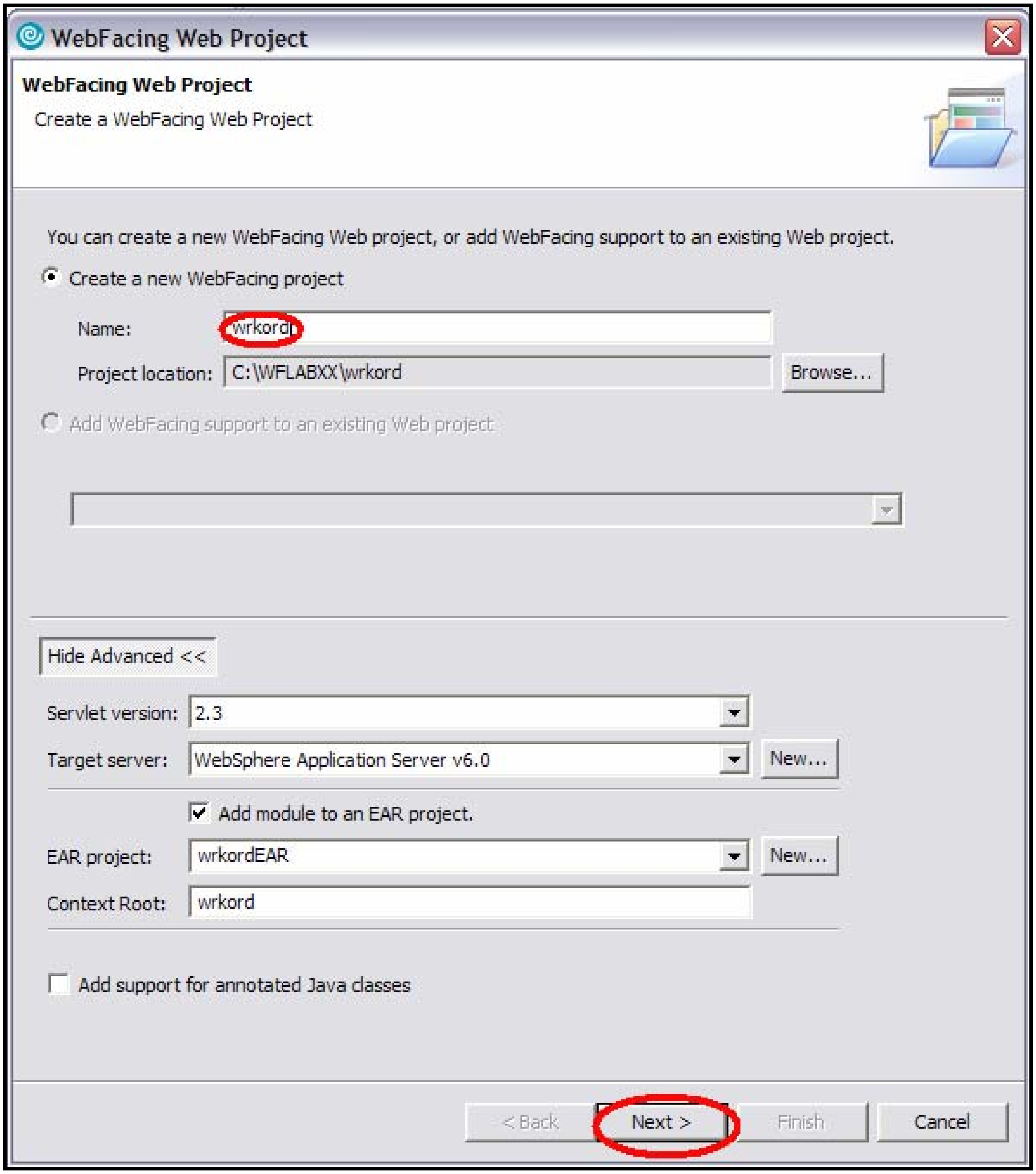


Figure 274. Creating a new WebFacing project

2. The first page of the WebFacing Web Project wizard is displayed (see Figure 275).



WebFacing Web Project

WebFacing Web Project

Create a WebFacing Web Project

You can create a new WebFacing Web project, or add WebFacing support to an existing Web project.

● Create a new WebFacing project

Name: wrkord

Project location: C:\WFLABXX\wrkord

○ Add WebFacing support to an existing Web project

Servlet version: 2.3

Target server: WebSphere Application Server v6.0

☑ Add module to an EAR project.

EAR project: wrkordEAR

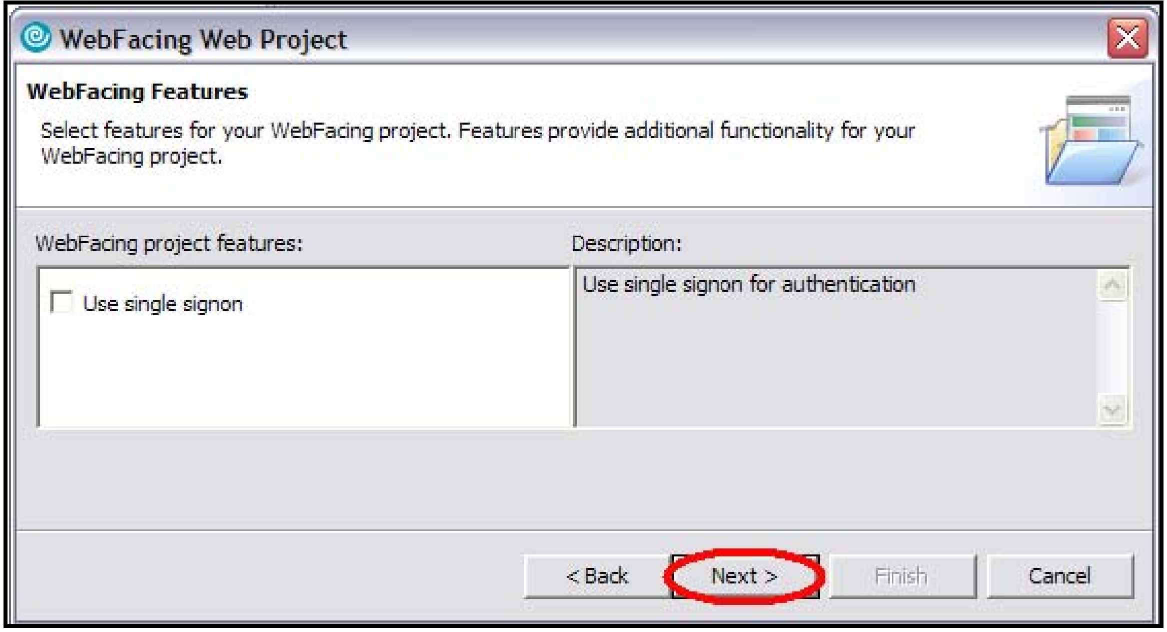
Context Root: wrkord

☐ Add support for annotated Java classes

Figure 275. The first page of the WebFacing Web Project wizard

3. In the **Project name** field, type the project name **wrkord**. The **EAR project** field is updated automatically.
4. Click **Next**.

5. The Webfacing Features dialog is displayed. Click **Next** (see Figure 276).



WebFacing Web Project

WebFacing Features

Select features for your WebFacing project. Features provide additional functionality for your WebFacing project.

WebFacing project features:

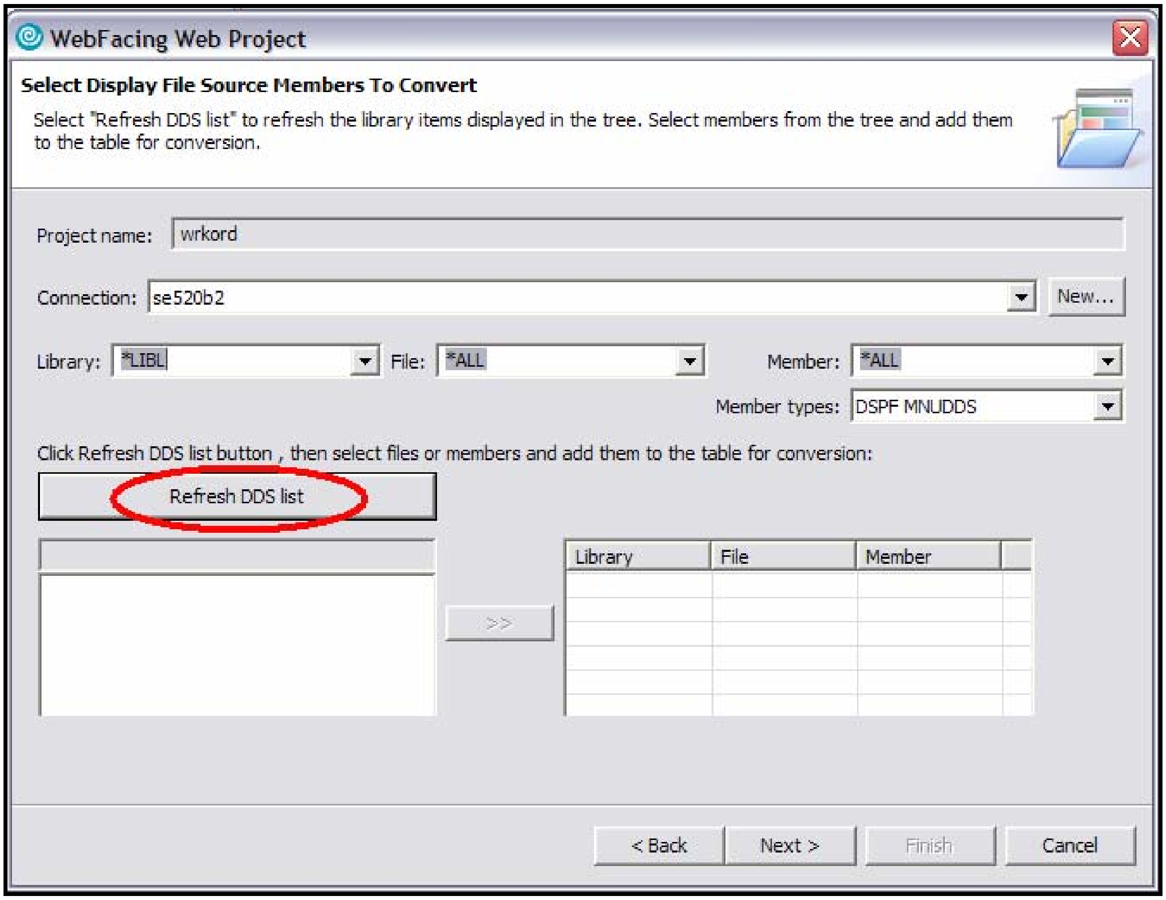
□ Use single signon

Description:

Use single signon for authentication

*Figure 276. On the Webfacing Features dialog page, click **Next**.*

6. The Select Display File Source Members to Convert panel is displayed (see Figure 277).



WebFacing Web Project

Select Display File Source Members To Convert

Select "Refresh DDS list" to refresh the library items displayed in the tree. Select members from the tree and add them to the table for conversion.

Project name: wrkord

Connection: se520b2

Library: *LIBL

File: *ALL

Member: *ALL

Member types: DSPF MNUDDS

Click Refresh DDS list button , then select files or members and add them to the table for conversion:

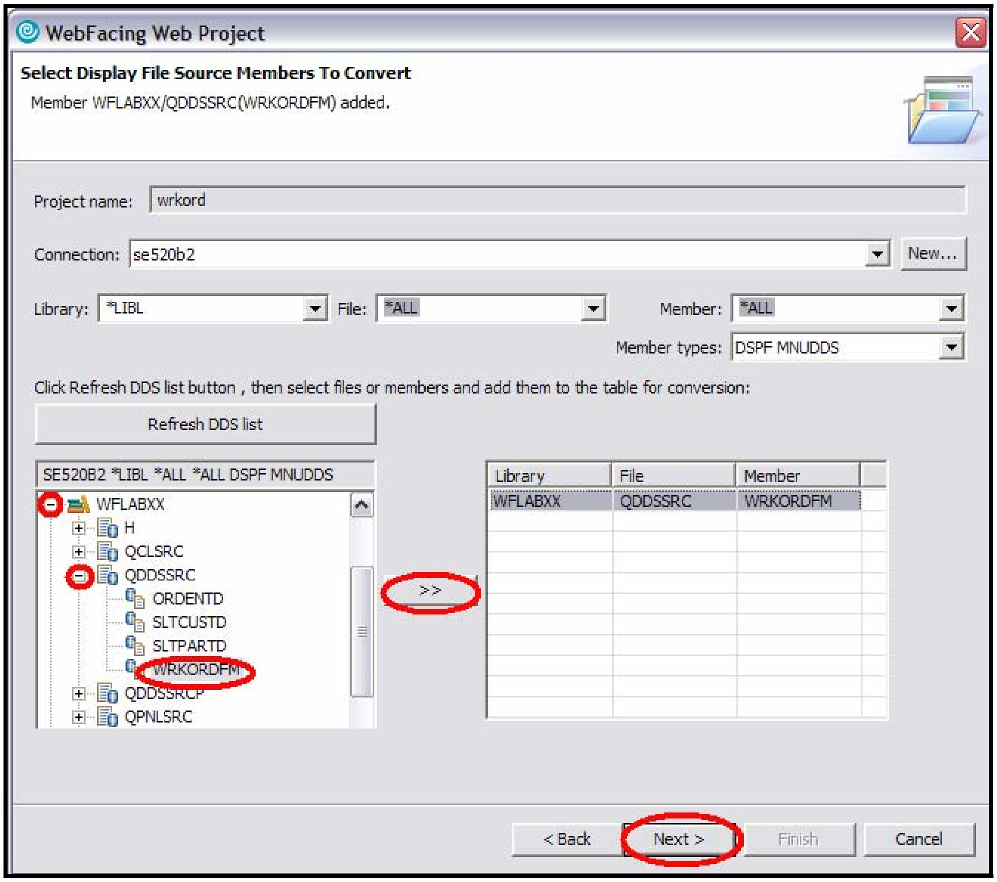
Library	File	Member

Figure 277. The Select Display File Source Members to Convert panel

7. Click the **Refresh DDS** list button to display the library list. If prompted, specify your i5/OS user ID and password.

Selecting the DDS source members

To select the DDS members to convert for this example, perform steps 1 through 6 (see Figure ***278***):



WebFacing Web Project

Select Display File Source Members To Convert

Member WFLABXX/QDDSSRC(WRKORDFM) added.

Project name: wrkord

Connection: se520b2

Library: *LIBL File: *ALL Member: *ALL

Member types: DSPF MNUDDS

Click Refresh DDS list button , then select files or members and add them to the table for conversion:

SE520B2 *LIBL *ALL *ALL DSPF MNUDDS

Library	File	Member
WFLABXX	QDDSSRC	WRKORDFM

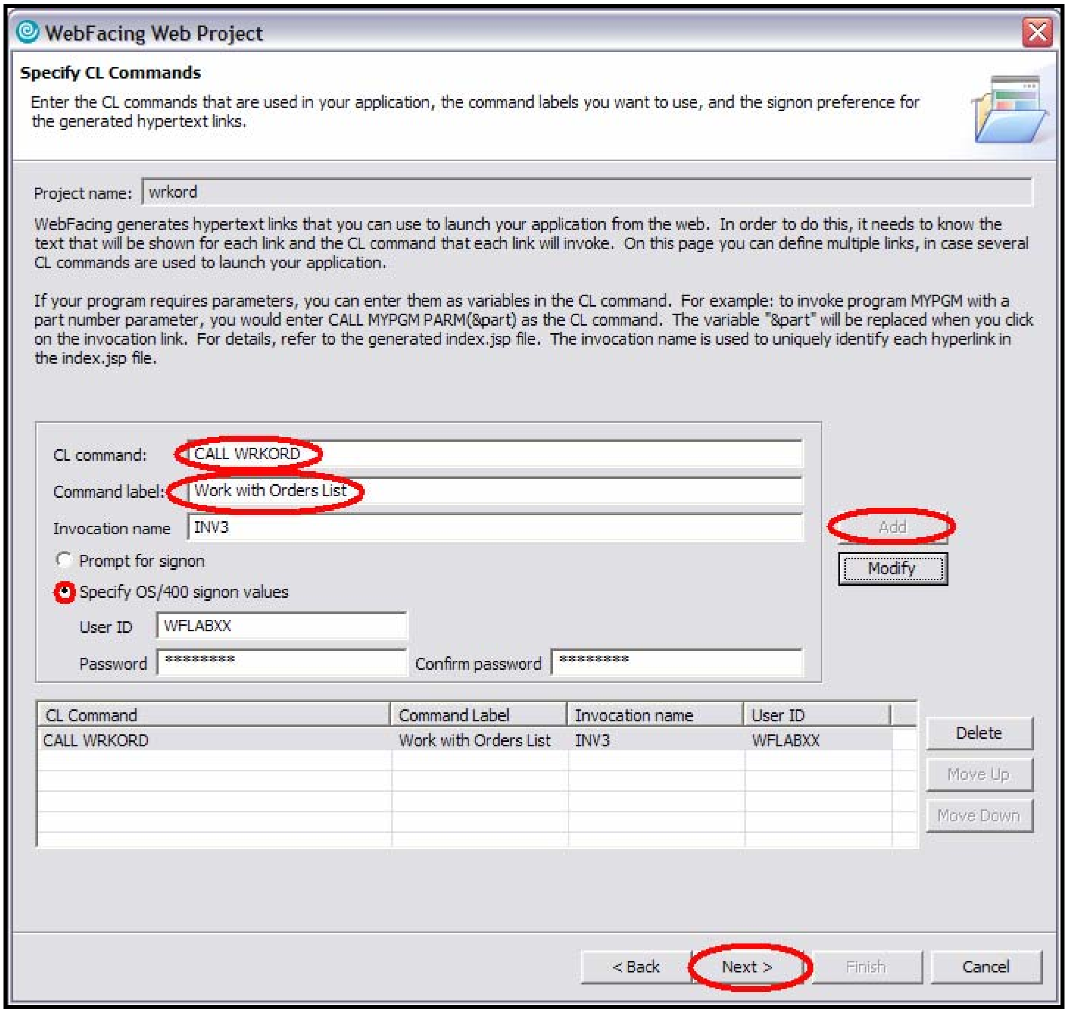
Figure 278. Selecting the DDS source members

1. Expand library **WFLABXX**.
2. Expand source member **QDDSSRC**.
3. Select the member **WRKORDFM**.
4. Click the >> button to add this member to the list of members to convert.
5. Click **Next**.
6. When the Select UIM Source Members to Convert page is displayed, just click **Next**.

   The Specify CL Commands page is displayed.

Specifying the CL command

Now, you specify the command to launch the sample application. The command you specify here, is the same command you used when you reviewed the sample application on the 5250 emulator: CALL WRKORD (see
Figure ***279***).



WebFacing Web Project

Specify CL Commands

Enter the CL commands that are used in your application, the command labels you want to use, and the signon preference for the generated hypertext links.

Project name: wrkord

WebFacing generates hypertext links that you can use to launch your application from the web. In order to do this, it needs to know the text that will be shown for each link and the CL command that each link will invoke. On this page you can define multiple links, in case several CL commands are used to launch your application.

If your program requires parameters, you can enter them as variables in the CL command. For example: to invoke program MYPGM with a part number parameter, you would enter CALL MYPGM PARM(&part) as the CL command. The variable "&part" will be replaced when you click on the invocation link. For details, refer to the generated index.jsp file. The invocation name is used to uniquely identify each hyperlink in the index.jsp file.

CL command: CALL WRKORD

Command label: Work with Orders List

Invocation name INV3

○ Prompt for signon

● Specify OS/400 signon values

User ID WFLABXX

Password ********* Confirm password *********

CL Command	Command Label	Invocation name	User ID
CALL WRKORD	Work with Orders List	INV3	WFLABXX

Figure 279. Specifying the CL command

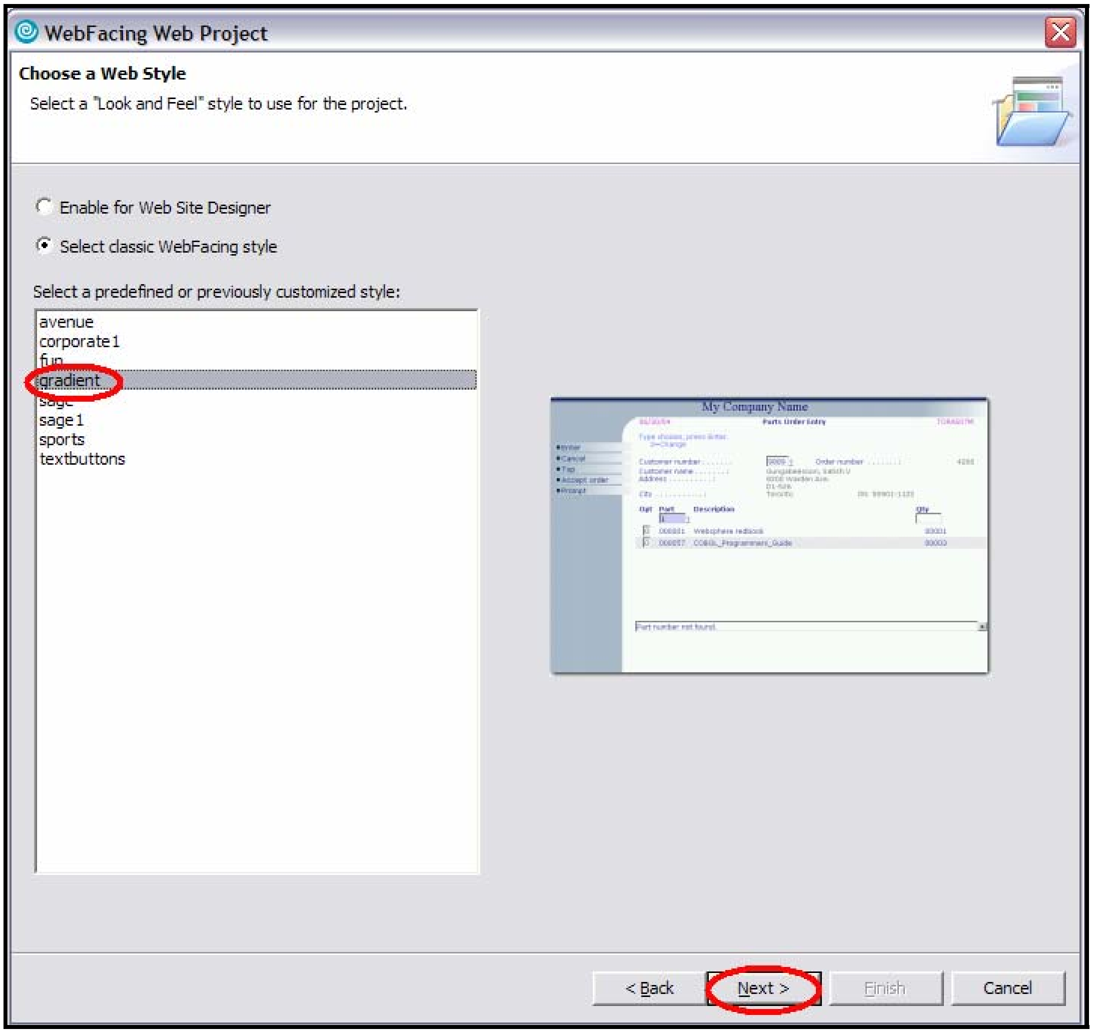
To specify the CL command:

1. In the **CL command** field, type **`CALL WRKORD`**.
2. In the **Command label** field, type **`Work with Orders List`**.
3. Check the radio button labeled **Specify OS/400 signon values**.
4. In the **User ID** field, type your i5/OS user ID.
5. In the **Password** and **Confirm password** fields, type your i5/OS password.
6. Click the **Add** button.
7. Click **Next**.

Selecting a Web style

On the next page of the wizard, you select a style for your application.

To select a Web style, perform the next two steps (see Figure ***280***):



WebFacing Web Project

Choose a Web Style

Select a "Look and Feel" style to use for the project.

○ Enable for Web Site Designer

● Select classic WebFacing style

Select a predefined or previously customized style:

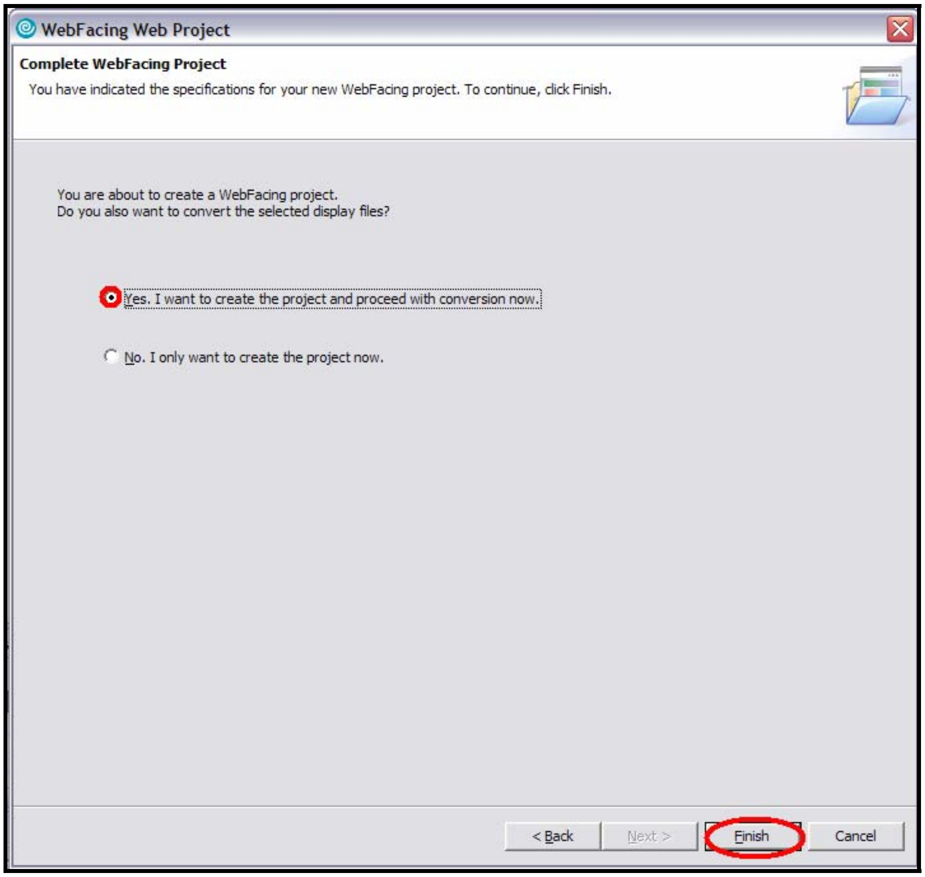
avenue
corporate1
fun
gradient
sage
sage1
sports
textbuttons

Figure 280. Selecting a Web Style

1. From the list of available styles, select **gradient**.
2. Click **Next**.

   The Complete WebFacing Project page is displayed (see Figure ***281***).

1. Select the radio button labeled **Yes I want to create the project and proceed with conversion now**.
2. Click **Finish**.



WebFacing Web Project

Complete WebFacing Project

You have indicated the specifications for your new WebFacing project. To continue, click Finish.

You are about to create a WebFacing project.
Do you also want to convert the selected display files?

● Yes. I want to create the project and proceed with conversion now.

○ No. I only want to create the project now.

Figure 281. The Complete WebFacing Project page

The WebFacing project wizard continues and creates the project. After the project is created, the conversion begins automatically.

After the project is created and the conversion is finished, the new project appears in the workbench (see Figure 282).

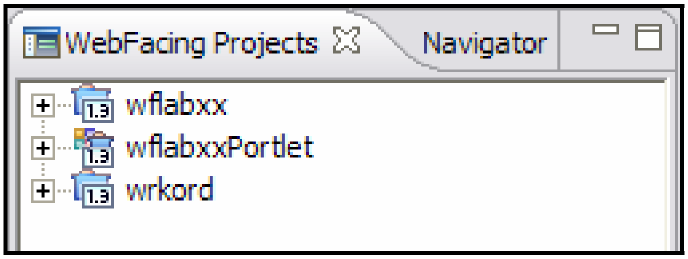


Figure 282. The new project appears in the workbench.

Exercise 15.3: Test the application with system screen support

To test the application:

1. Right-click the **wrkord** project in the WebFacing Projects view (see Figure ***283***).

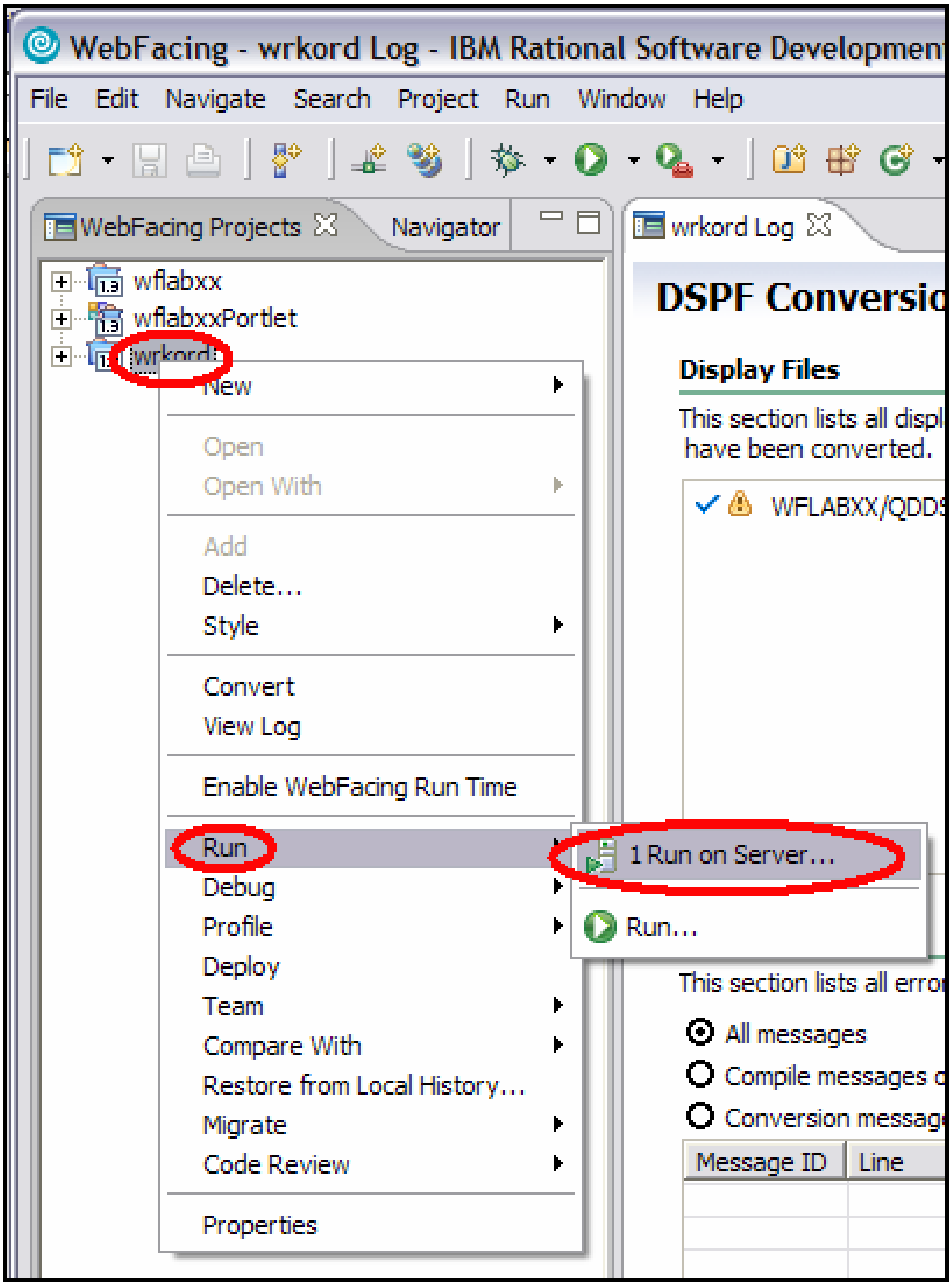
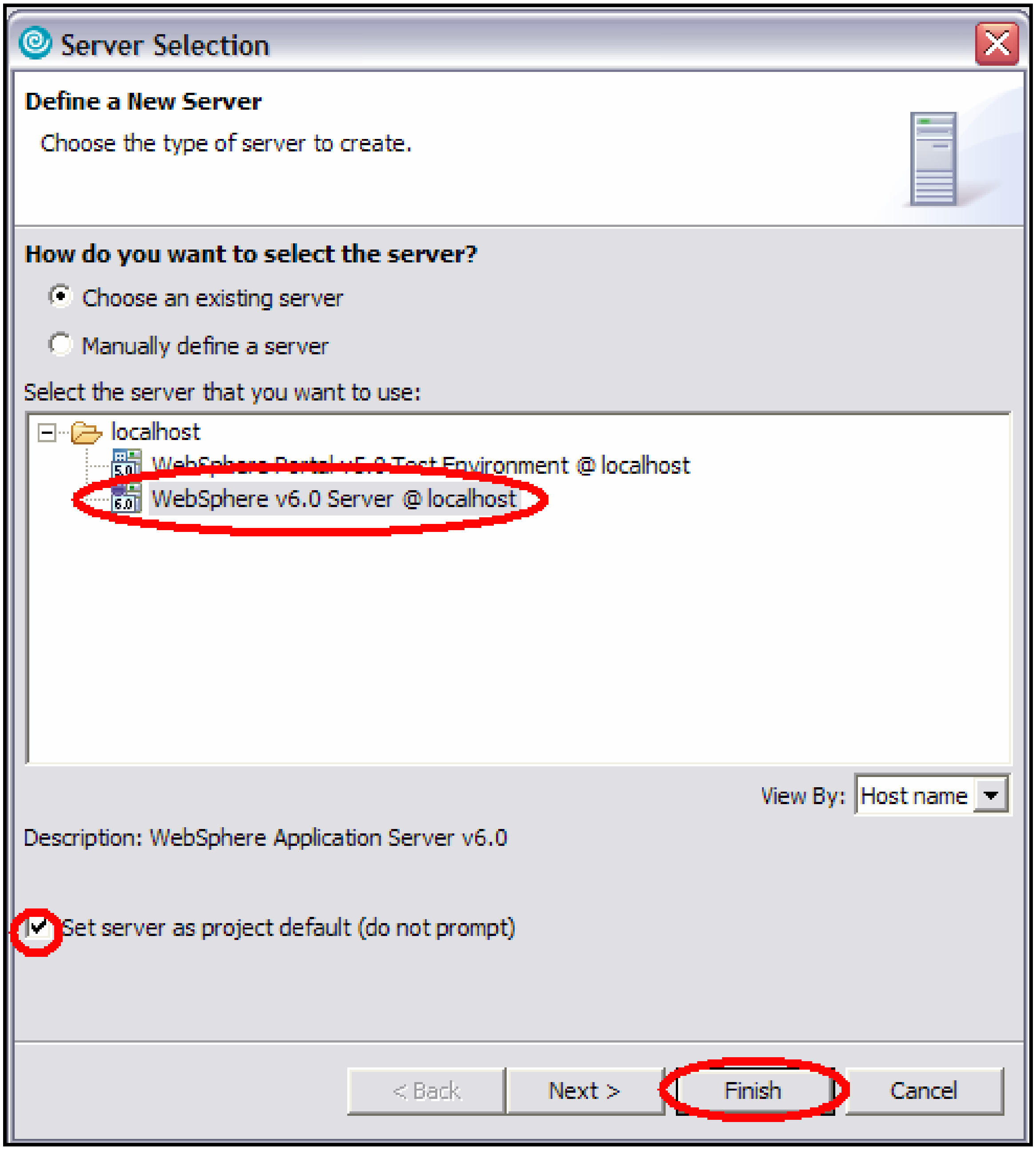


*Figure 283. Right-click the **wrkord** project in the WebFacing Projects view.*

2. Click **Run>Run on Server** on the pop-up menu.

   The Server Selection dialog is displayed (see Figure ***284***).



Server Selection

Define a New Server

Choose the type of server to create.

How do you want to select the server?

● Choose an existing server

○ Manually define a server

Select the server that you want to use:

- localhost
  - WebSphere Portal v5.0 Test Environment @ localhost
  - WebSphere v6.0 Server @ localhost

View By: Host name

Description: WebSphere Application Server v6.0

☑ Set server as project default (do not prompt)

Figure 284. The Server Selection dialog

3. Check the check box labeled **Set server as project default (do not prompt)** to avoid having this dialog appear each time you run this application.
4. Click **Finish**.

The index.jsp page of the application is displayed (see Figure ***285***).

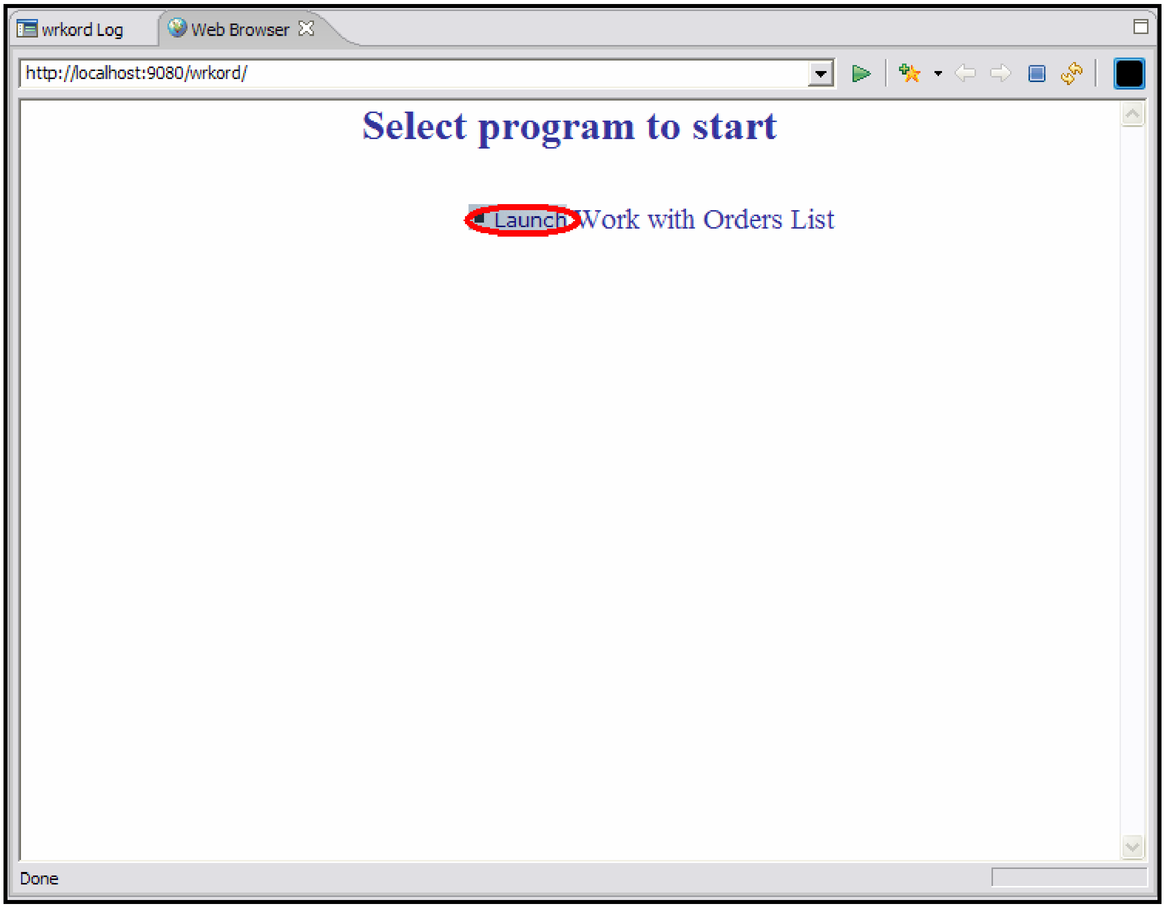
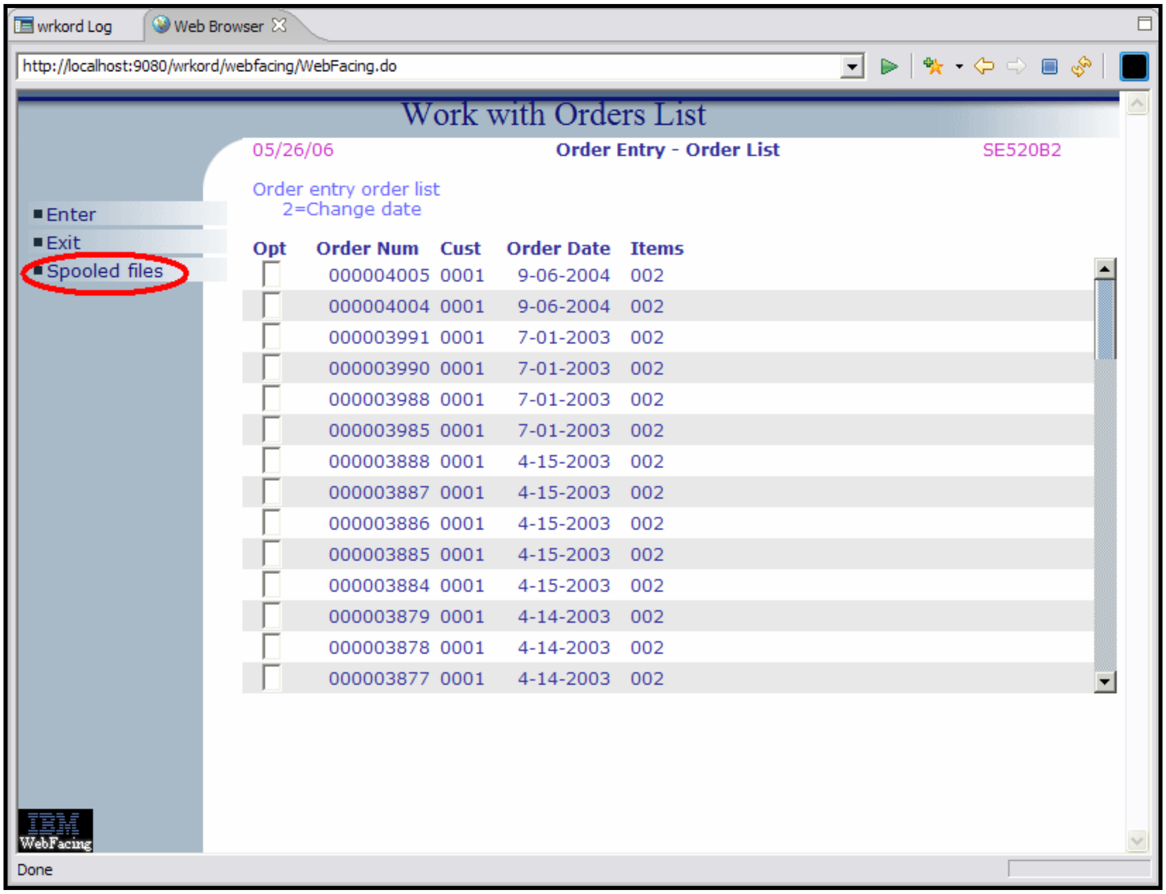


Figure 285. The index.jsp page of the application

5. Click the Launch button labeled **Work with Orders List**.

The display screen for the sample application is displayed (see Figure ***286***).



http://localhost:9080/wrkord/webfacing/WebFacing.do

Work with Orders List

05/26/06 Order Entry - Order List SE520B2

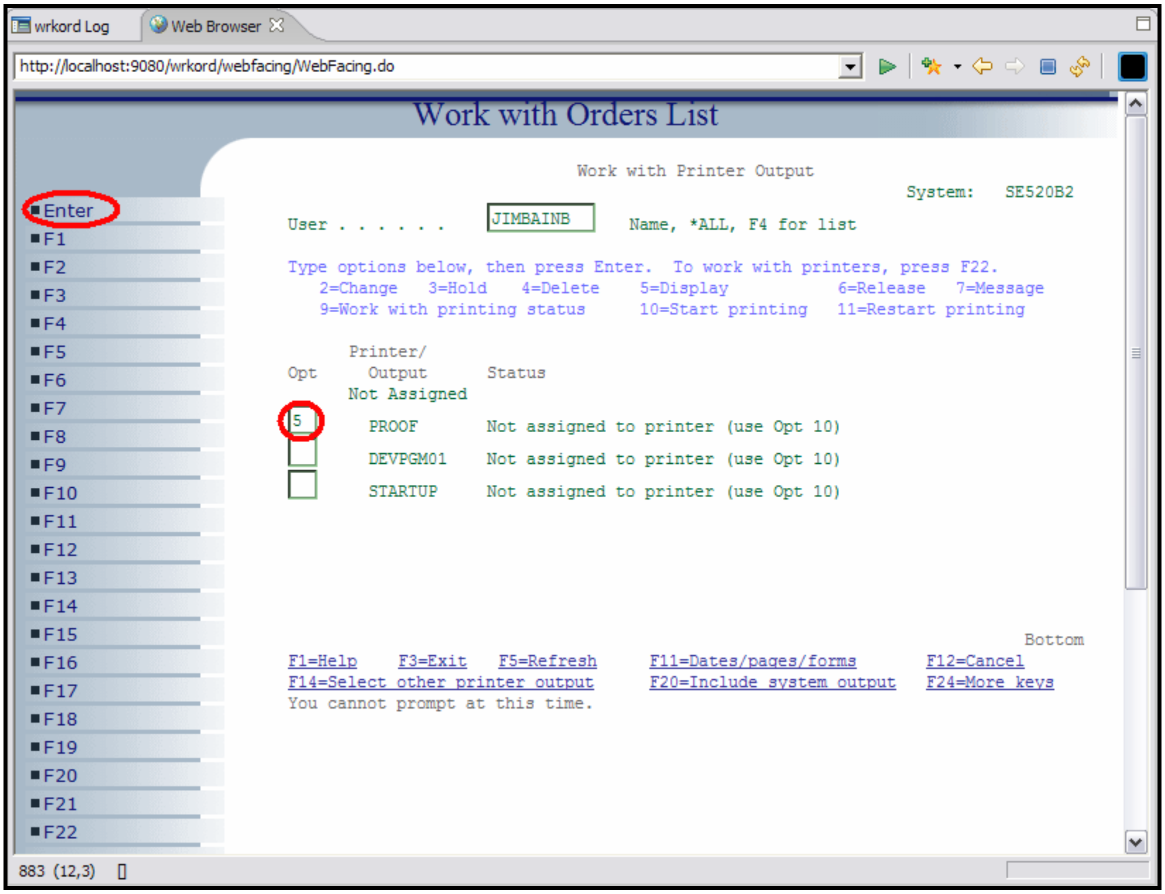
Order entry order list
2=Change date

Opt	Order Num	Cust	Order Date	Items
□	000004005	0001	9-06-2004	002
□	000004004	0001	9-06-2004	002
□	000003991	0001	7-01-2003	002
□	000003990	0001	7-01-2003	002
□	000003988	0001	7-01-2003	002
□	000003985	0001	7-01-2003	002
□	000003888	0001	4-15-2003	002
□	000003887	0001	4-15-2003	002
□	000003886	0001	4-15-2003	002
□	000003885	0001	4-15-2003	002
□	000003884	0001	4-15-2003	002
□	000003879	0001	4-14-2003	002
□	000003878	0001	4-14-2003	002
□	000003877	0001	4-14-2003	002

Figure 286. The display screen for the sample application

6. Click the button labeled **Spooled files**, or press **F6**.

   The Work with All Spooled Files system screen is displayed in the browser (see Figure ***286***).



http://localhost:9080/wrkord/webfacing/WebFacing.do

Work with Orders List

```
                         Work with Printer Output
                                                          System:   SE520B2
User . . . . . .       JIMBAINB      Name, *ALL, F4 for list

Type options below, then press Enter.  To work with printers, press F22.
   2=Change   3=Hold   4=Delete   5=Display          6=Release   7=Message
   9=Work with printing status    10=Start printing   11=Restart printing

      Printer/
Opt     Output      Status
      Not Assigned
 5      PROOF       Not assigned to printer (use Opt 10)
        DEVPGM01    Not assigned to printer (use Opt 10)
        STARTUP     Not assigned to printer (use Opt 10)

                                                                   Bottom
F1=Help    F3=Exit   F5=Refresh    F11=Dates/pages/forms      F12=Cancel
F14=Select other printer output    F20=Include system output  F24=More keys
You cannot prompt at this time.
```

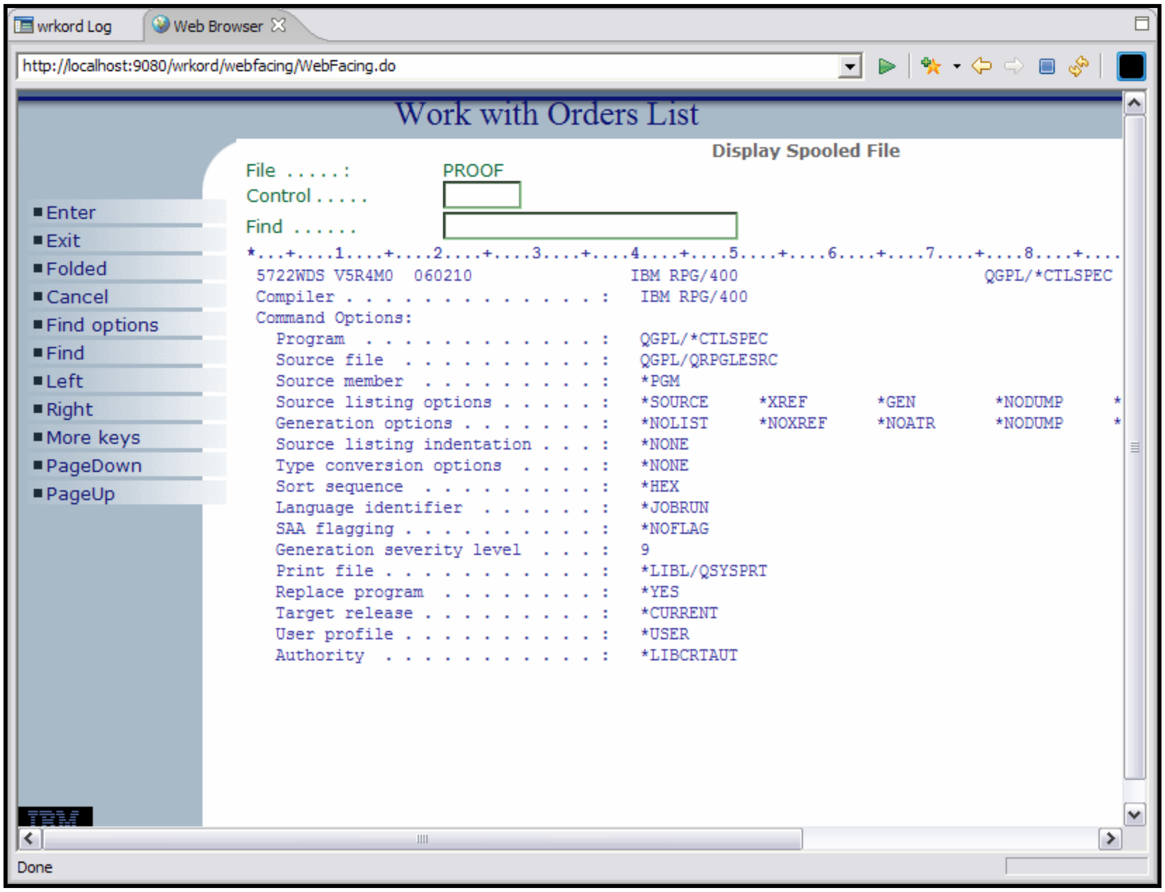
883 (12,3)

Figure 287. The Work with All Spooled Files system screen

7. To work with a spooled file entry:

   a. Use the **page up** and **page down** keys on the keyboard to scroll through the list of spooled files.
   b. Type the option you wish to perform in the **Opt** column f one of the items. For example, type 5 to display an item.
   c. Press **Enter** or click the **Enter** push button.

The Display Spooled File system screen is displayed (see Figure ***288***).



Display Spooled File

File . . . . . : PROOF
Control . . . . .
Find . . . . . .

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....
 5722WDS V5R4M0  060210                    IBM RPG/400                         QGPL/*CTLSPEC
 Compiler . . . . . . . . . . . . . . :   IBM RPG/400
 Command Options:
   Program  . . . . . . . . . . . . . :   QGPL/*CTLSPEC
   Source file  . . . . . . . . . . . :   QGPL/QRPGLESRC
   Source member  . . . . . . . . . . :   *PGM
   Source listing options . . . . . . :   *SOURCE      *XREF        *GEN         *NODUMP      *
   Generation options . . . . . . . . :   *NOLIST      *NOXREF      *NOATR       *NODUMP      *
   Source listing indentation . . . . :   *NONE
   Type conversion options  . . . . . :   *NONE
   Sort sequence  . . . . . . . . . . :   *HEX
   Language identifier  . . . . . . . :   *JOBRUN
   SAA flagging . . . . . . . . . . . :   *NOFLAG
   Generation severity level  . . . . :   9
   Print file . . . . . . . . . . . . :   *LIBL/QSYSPRT
   Replace program  . . . . . . . . . :   *YES
   Target release . . . . . . . . . . :   *CURRENT
   User profile . . . . . . . . . . . :   *USER
   Authority  . . . . . . . . . . . . :   *LIBCRTAUT
```

Figure 288. The Display Spooled File system screen

Recap

You have completed “Testing System Screen support.” You now have the information to understand how to:

- Create a WebFacing project for a program with system screen support
- Test the project in the WebSphere test environment and verify the system screen support

Invoking a Java application from a WebFaced Web page

In this chapter, you learn to invoke a Java program using a hyperlink on a WebFacing Web page. Integrating an IBM WebFacing Tool application with other Java-based applications is a great way to provide new functionality not available in the original RPG application.

For this example, the lab provides a simple Java application that does nothing more than display the two arguments passed to it in a new browser window. The arguments passed to the Java application are data fields from the WebFacing application.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 16.1: Import and understand CustomerOrder
- Exercise 16.2: Deploying the Java application
- Exercise 16.3: Changing the Web Settings to call CustomerOrder.jsp
- Exercise 16.4: Test the integration of Java application with WebFacing

The exercises within each chapter must be completed in order. Start with “Exercise 16.1: Import and Understand CustomerOrder.”

Length of time

This chapter takes approximately 20 minutes to complete.

Exercise 16.1: Import and understand CustomerOrder

First, you need to import the Java application into WebSphere Development Studio Client, and then you can explore the Java application. This application is a dynamic Web application that can be deployed to an application server much like a WebFacing application.

The goal of this exercise is to demonstrate how a refacecd application might call Java applications. It iscome apparent that this example does not do anything useful, but can easily be modified to be useful. DispArgs.java can run a DataBase Query based on the input data, called a Web Service, or a whole host of other valid business functions that can increase the functionality of the WebFacing application.

Before you can examine the CustomerOrder application, you need to import it into WebSphere Development Studio Client.

.

1. In WebSphere Development Studio Client click **File** from the menu and select **Import** (see
   Figure ***289***).

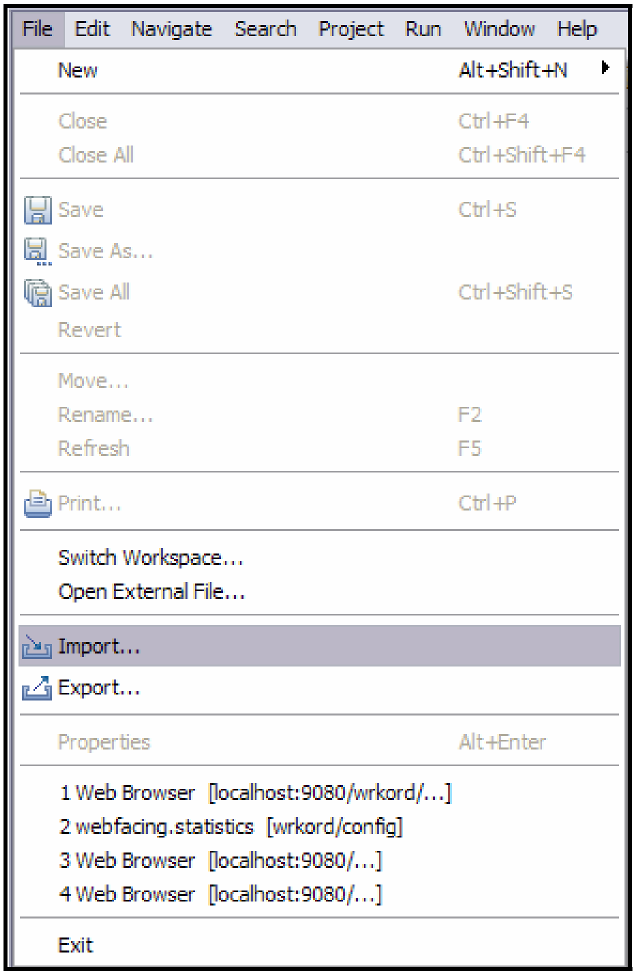


*Figure 289. In WebSphere Development Studio Client, click **File** from the menu. Select **Import**.*

2. Select **EAR file** from the list, click **Next** (see Figure ***290***).

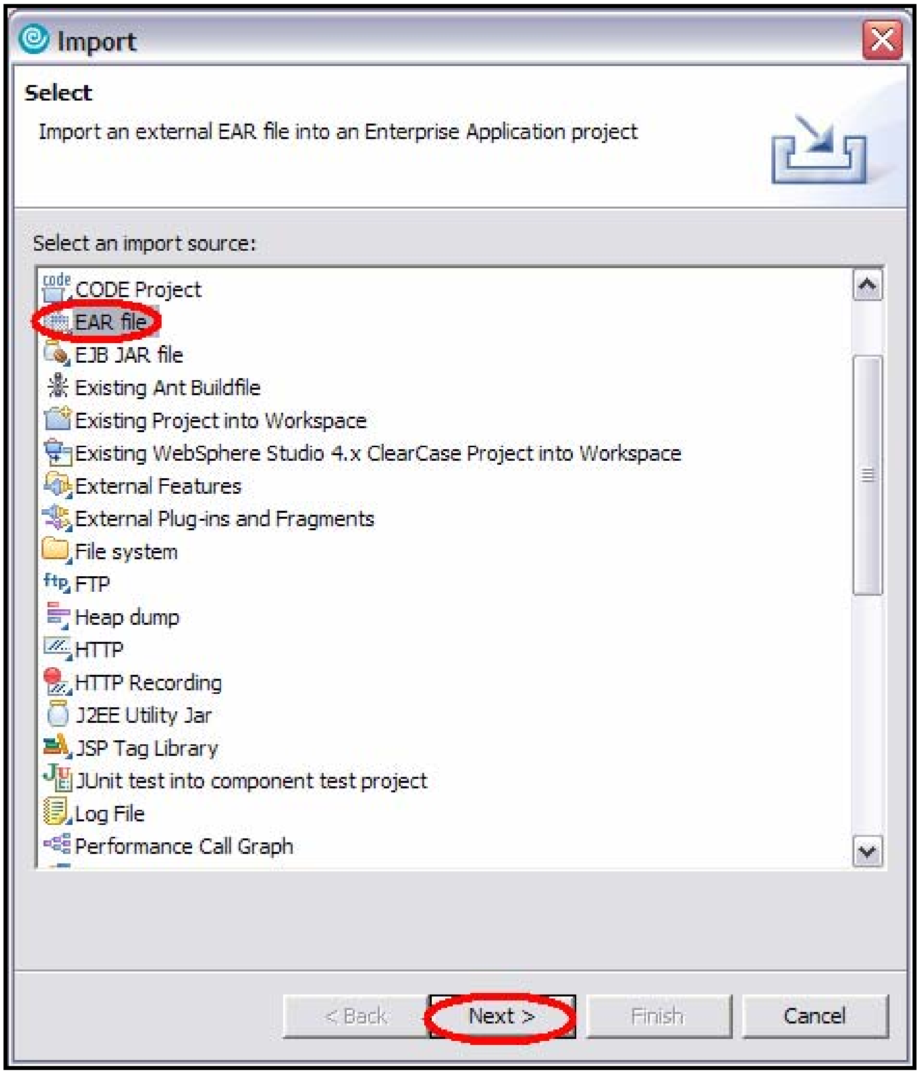
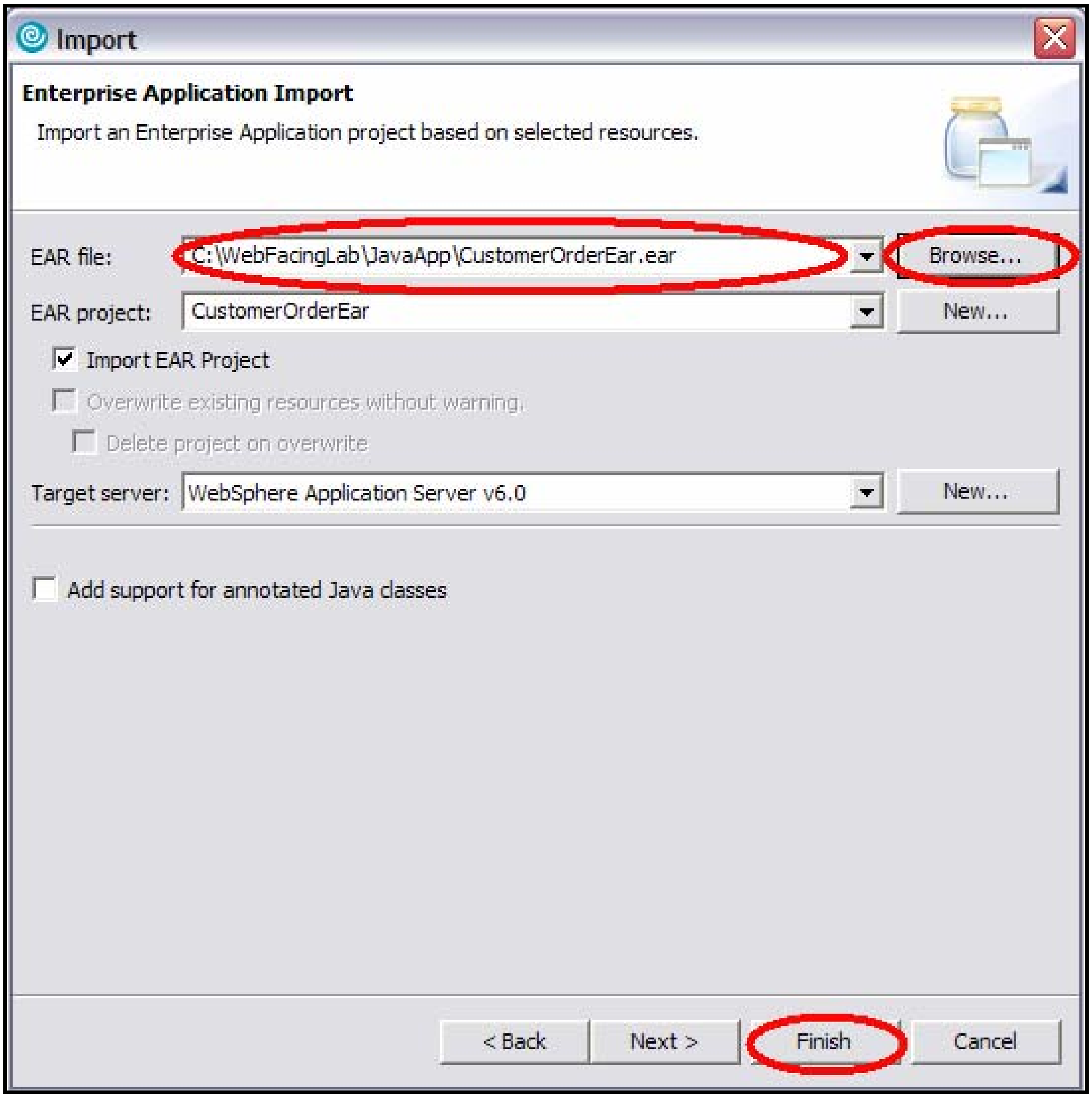


*Figure 290. Select **EAR file** from the list, click **Next**.*

3. Browse to the CustomerOrderEar.ear file in the **C:\WebFacingLab\JavaApp\CustomerOrderEar.ear** directory. Accept the rest of the defaults and click **Finish** (see
Figure ***291***).



Import

Enterprise Application Import

Import an Enterprise Application project based on selected resources.

EAR file: C:\WebFacingLab\JavaApp\CustomerOrderEar.ear

EAR project: CustomerOrderEar

☑ Import EAR Project

□ Overwrite existing resources without warning.

□ Delete project on overwrite

Target server: WebSphere Application Server v6.0

□ Add support for annotated Java classes

*Figure 291. Accept the defaults and click **Finish**.*

4. Switch to the Navigator view of the WebFacing Perspective. You can now see CustomerOrder. This application has three files of interest (see Figure ***292***):
   - DispArgs.java: Java servlet called by WebFaced application
   - ViewBean.java: Java class that stores the arguments passed to DispArgs.java
   - CustomerOrder.jsp: Java Server Page that displays the results

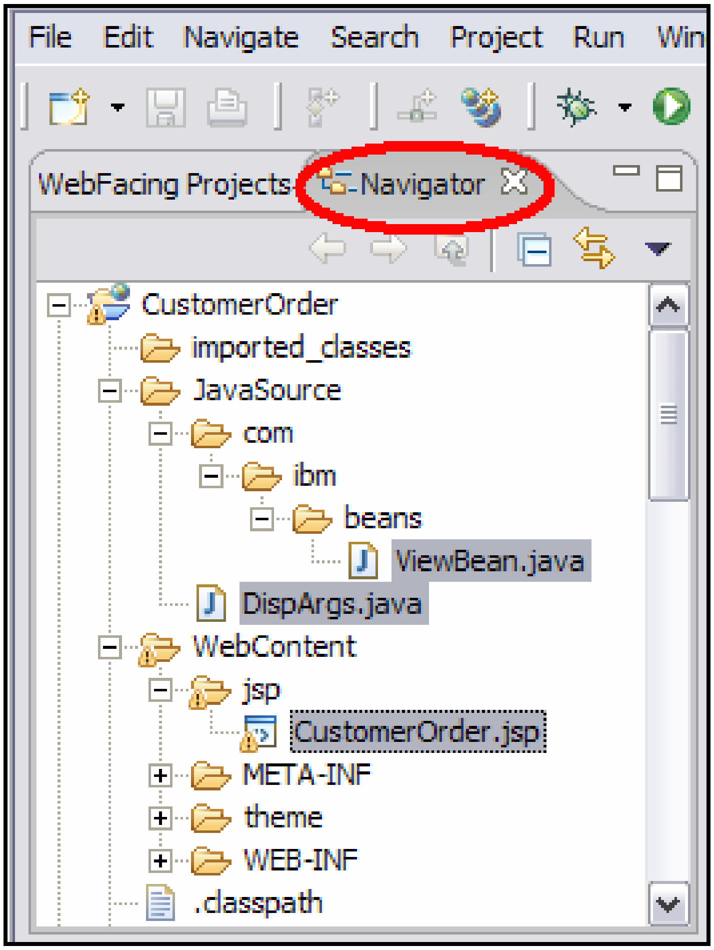
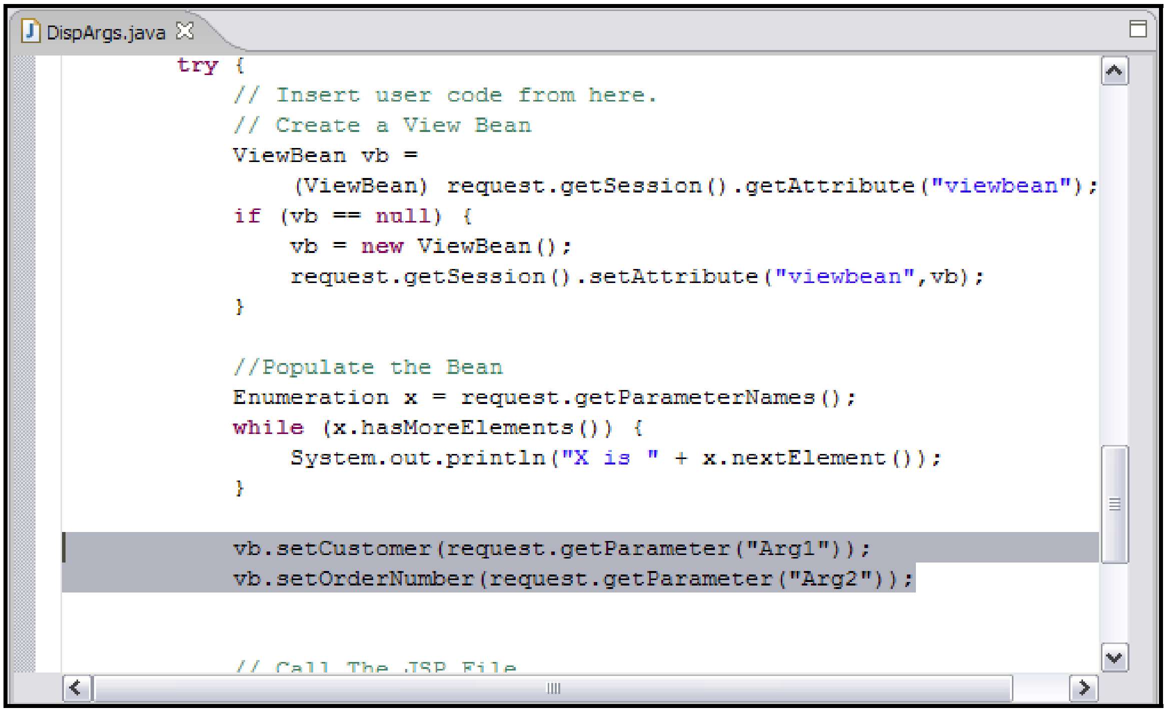


Figure 292. The CustomerOrder.jsp (JavaServer Page) displays the results.

At this time, review the DispArgs.java, ViewBean.java and CustomerOrder.jsp files. Of most importance to the student is the performTask() method within the DispArgs.java file. Notice within this method, the parameters for **Arg1**, and **Arg2** are retrieved from the request object (see
Figure ***293***). The reason for this importance is apparent later.



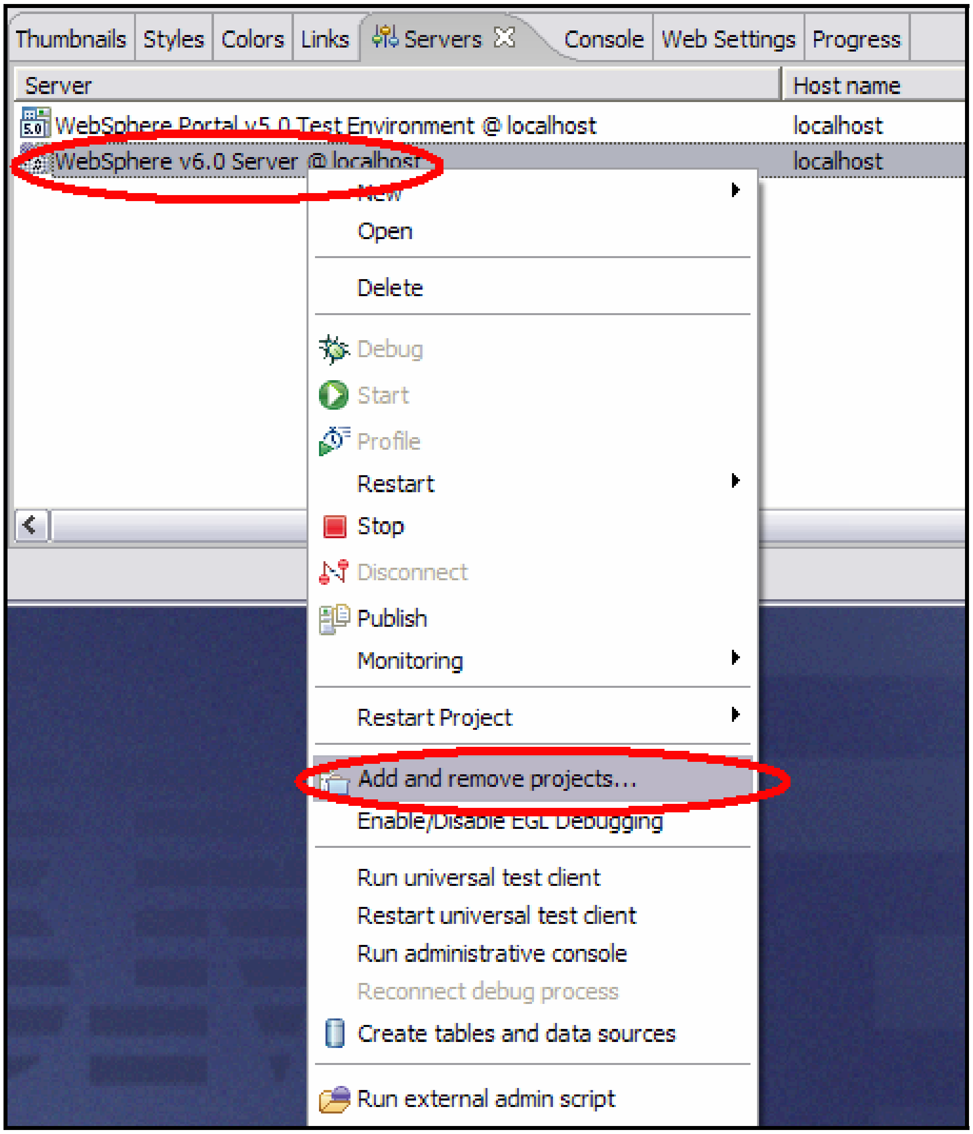
```
        try {
            // Insert user code from here.
            // Create a View Bean
            ViewBean vb =
                (ViewBean) request.getSession().getAttribute("viewbean");
            if (vb == null) {
                vb = new ViewBean();
                request.getSession().setAttribute("viewbean",vb);
            }

            //Populate the Bean
            Enumeration x = request.getParameterNames();
            while (x.hasMoreElements()) {
                System.out.println("X is " + x.nextElement());
            }

            vb.setCustomer(request.getParameter("Arg1"));
            vb.setOrderNumber(request.getParameter("Arg2"));


            // Call The JSP File
```

Figure 293. The parameters for Arg1, and Arg2 are retrieved from the request object.

Exercise 16.2: Deploying the Java application

Before you make changes to the WebFacing application, make sure the CustomerOrder application has been deployed to the same WebSphere Application Server test environment within WebSphere Development Studio Client.

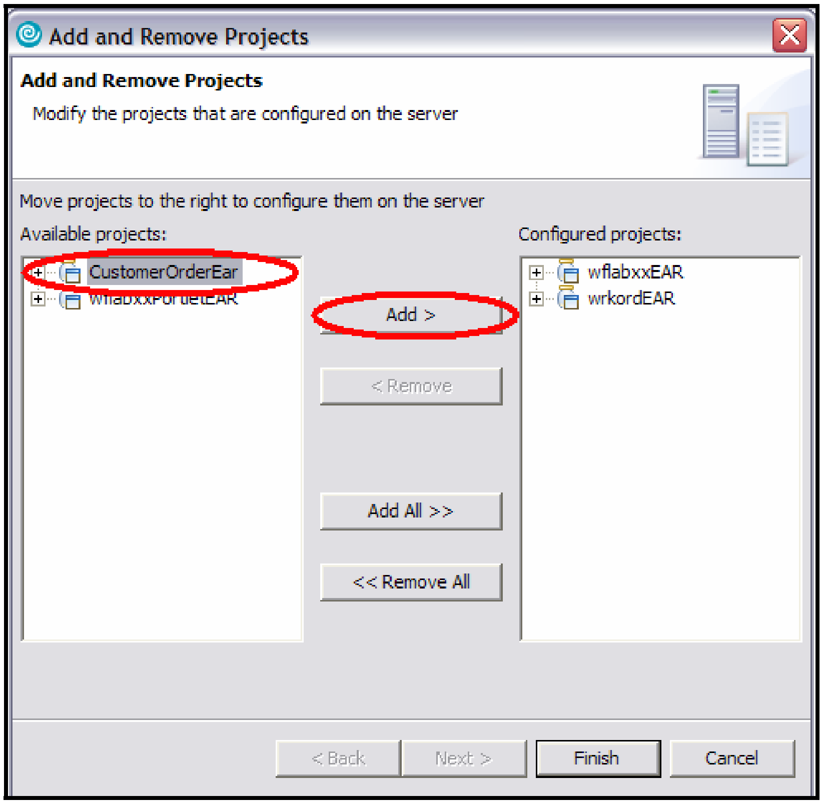
1. This can be done in the Navigator view of the WebFacing Perspective. Expand your WebSphere V6.0 test environment to see what projects are currently associated with it. Right-click your **WebSphere v6.0 Server @ localhost server** and select **Add and remove projects…** (see
   Figure ***294***).



Thumbnails
Styles
Colors
Links
Servers
Console
Web Settings
Progress
Server
Host name
WebSphere Portal v5.0 Test Environment @ localhost
localhost
WebSphere v6.0 Server @ localhost
localhost
New
Open
Delete
Debug
Start
Profile
Restart
Stop
Disconnect
Publish
Monitoring
Restart Project
Add and remove projects…
Enable/Disable EGL Debugging
Run universal test client
Restart universal test client
Run administrative console
Reconnect debug process
Create tables and data sources
Run external admin script

*Figure 294. Right-click WebSphere V6.0 Server @ localhost server and select **Add and remove projects***

2. On the **Add and Remove Projects** pop-up, select **CustomerOrderEar**, click the **Add** button (see
Figure ***295***), and click **Finish** (see
Figure ***296***).



Add and Remove Projects

Add and Remove Projects

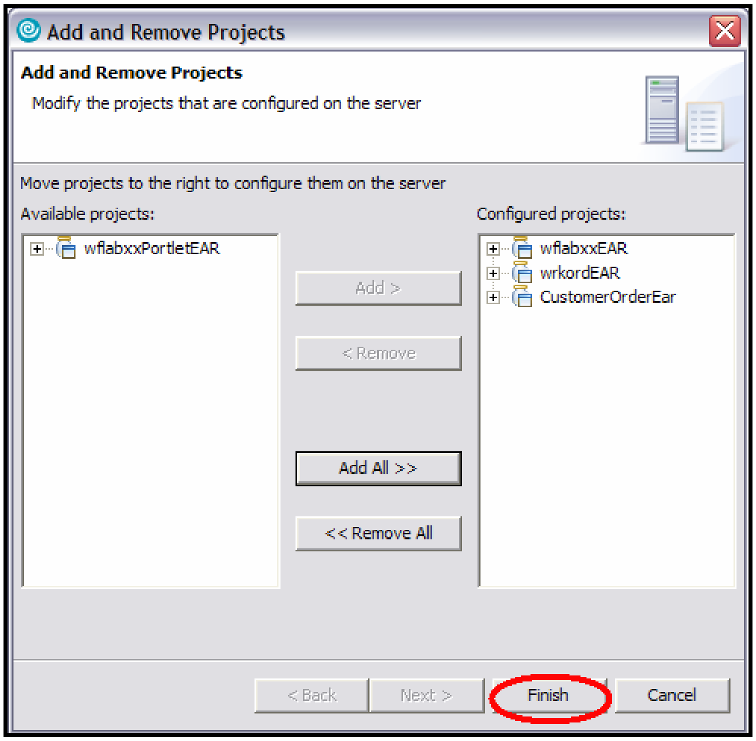
Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Available projects:
- CustomerOrderEar
- WflabxxPortletEAR

Configured projects:
- wflabxxEAR
- wrkordEAR

*Figure 295. Select **CustomerOrderEar**, click **Add**.*



Add and Remove Projects

Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Available projects:
- wflabxxPortletEAR

Configured projects:
- wflabxxEAR
- wrkordEAR
- CustomerOrderEar

*Figure 296. Click **Finish**.*

Now when the server is started, the Java application is installed and ready to run.

3. You can test the application in the Web Perspective by expanding: CustomerOrder > WebContent > jsp, right-clicking CustomerOrder.jsp and selecting **Run>Run on Server** (see Figure ***297***).

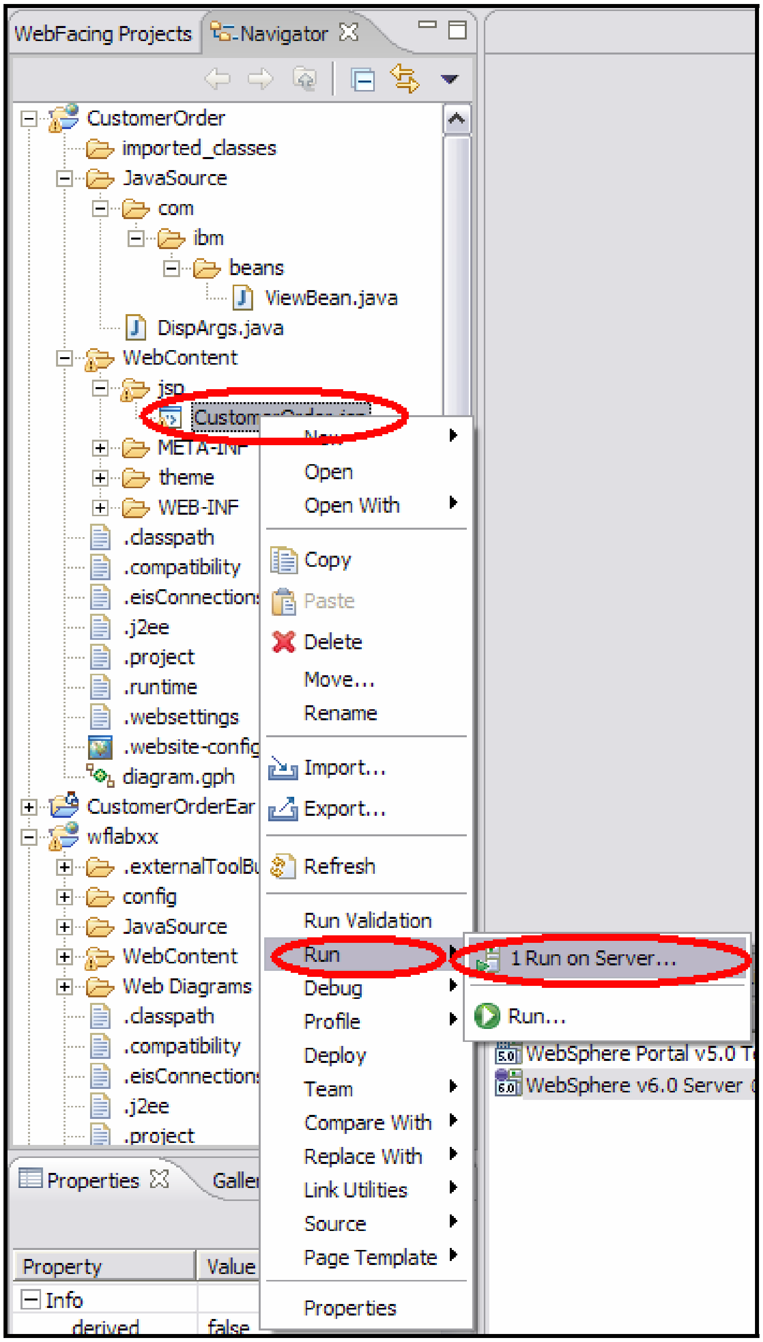
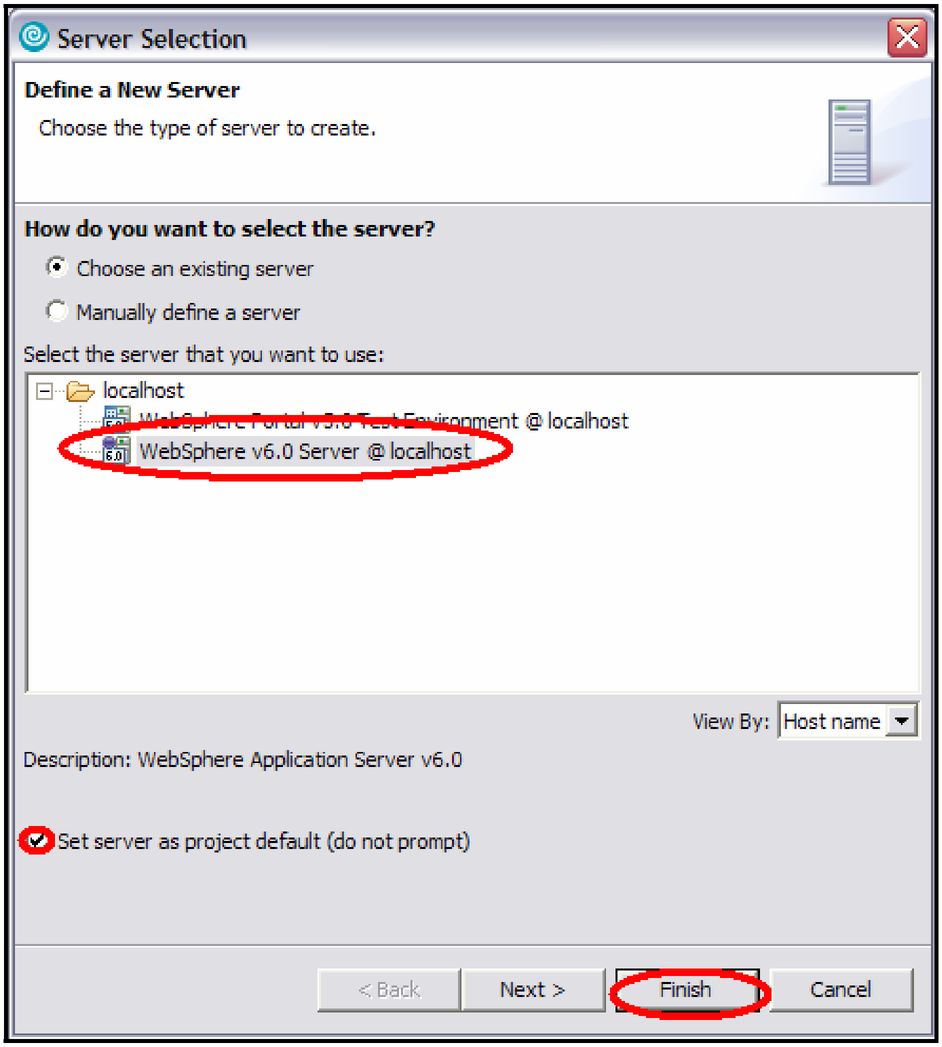


Figure 297. Testing the application in the Web perspective

4. If the Server Selection dialog is displayed, ensure the **WebSphere v6.0 Server @ localhost** is highlighted, select the **Set server as project default (do not prompt),** and click **Finish** (see

Figure ***298***).



Server Selection

Define a New Server

Choose the type of server to create.

How do you want to select the server?

● Choose an existing server

○ Manually define a server

Select the server that you want to use:

- localhost
  - WebSphere Portal v5.0 Test Environment @ localhost
  - WebSphere v6.0 Server @ localhost

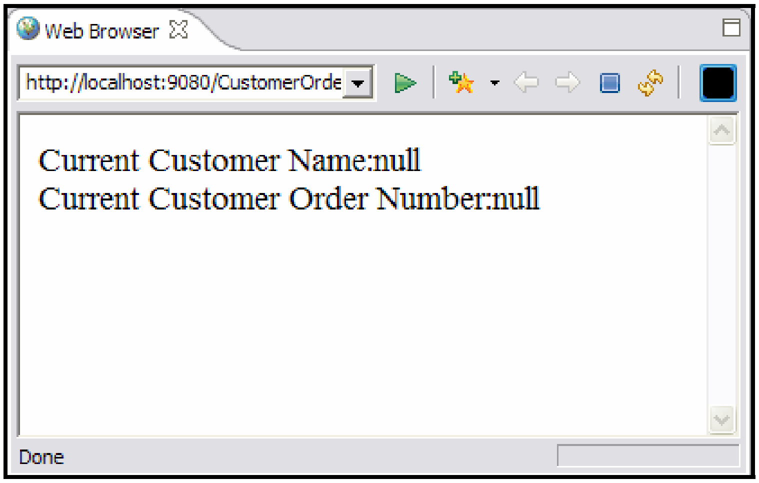
View By: Host name

Description: WebSphere Application Server v6.0

☑ Set server as project default (do not prompt)

*Figure 298. Select **Set server as project default (do not prompt)** and click **Finish**.*

5. In the Web Browser**,** you can see the output results of the jsp. Do not worry about the null values, they are null because in this test you did not pass any values (see Figure ***299***).



```
Current Customer Name:null
Current Customer Order Number:null
```

Figure 299. The output results of the jsp

Exercise 16.3: Changing the Web Settings to call CustomerOrder.jsp

For this example, change the WebFacing application to include a hyperlink of the customer name field that calls the Java Application when clicked, passing in two arguments from the screen.

1. From the WebFacing Project view of the WebFacing Perspective. Expand your **WFLabxx project WFLabxx > DDS**, right-click the DDS source ORDENTD and select **Open With > CODE Designer** (see
   Figure ***300***).

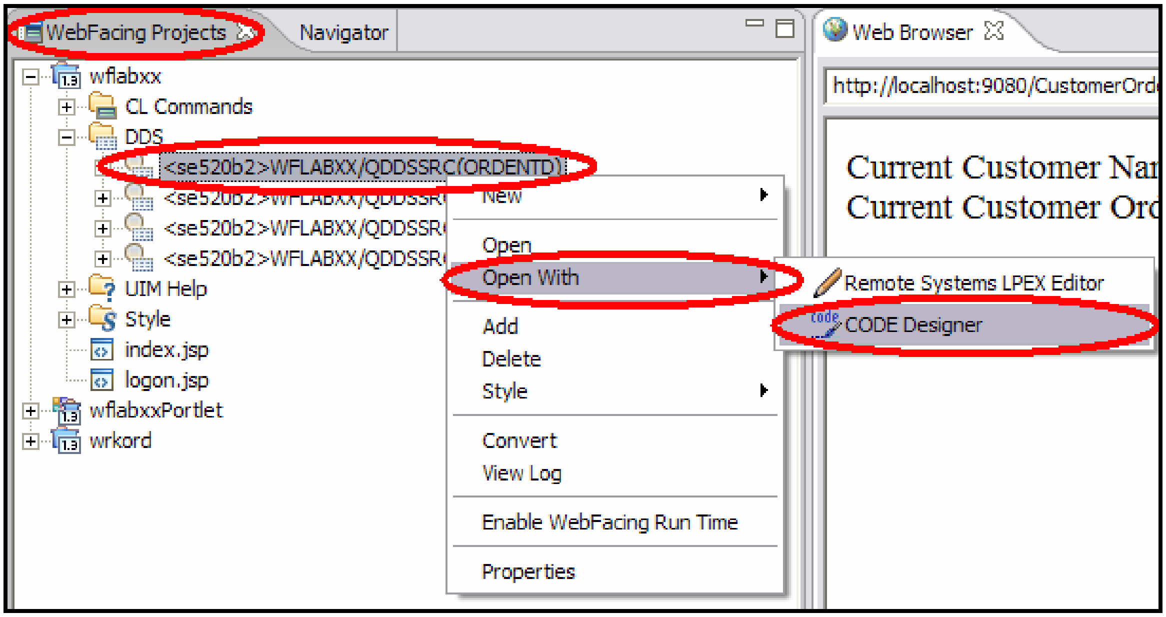
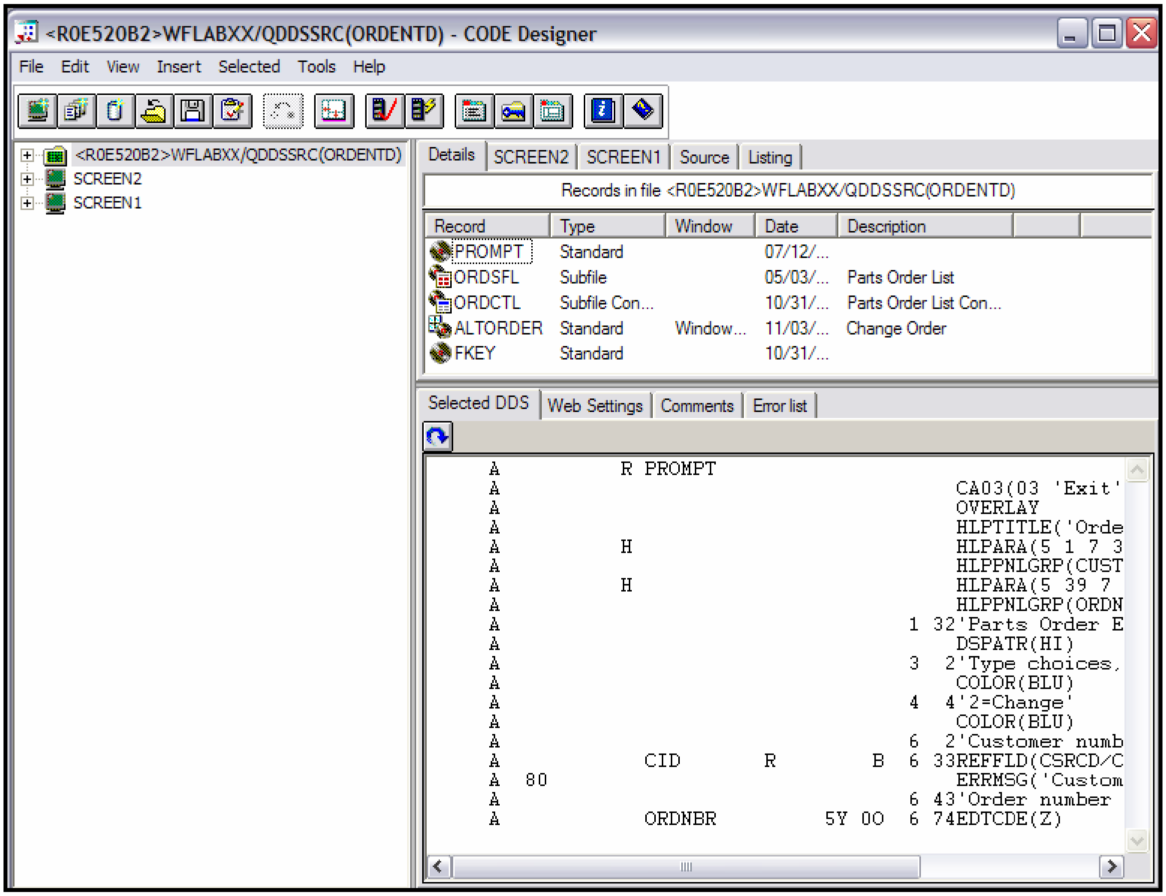


Figure 300. Changing the Web Settings to call CustomerOrder.jsp

2. The CODE Designer development environment is displayed (see Figure ***301***).



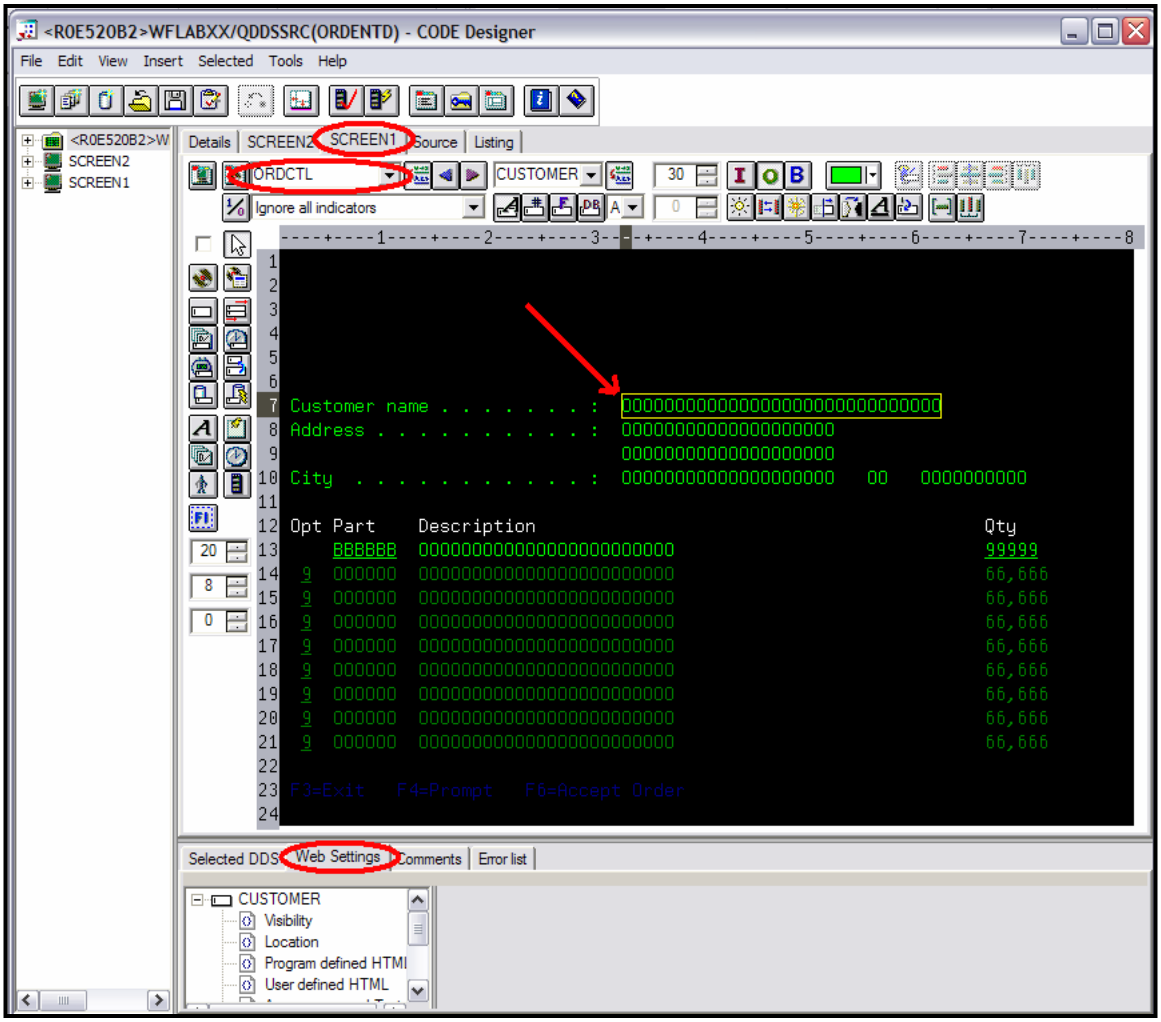
Records in file <R0E520B2>WFLABXX/QDDSSRC(ORDENTD)

Record	Type	Window	Date	Description
PROMPT	Standard		07/12/...	
ORDSFL	Subfile		05/03/...	Parts Order List
ORDCTL	Subfile Con...		10/31/...	Parts Order List Con...
ALTORDER	Standard	Window...	11/03/...	Change Order
FKEY	Standard		10/31/...	

```
A          R PROMPT
A                                      CA03(03 'Exit'
A                                      OVERLAY
A                                      HLPTITLE('Orde
A          H                           HLPARA(5 1 7 3
A                                      HLPPNLGRP(CUST
A          H                           HLPARA(5 39 7
A                                      HLPPNLGRP(ORDN
A                                  1 32'Parts Order E
A                                      DSPATR(HI)
A                                  3  2'Type choices,
A                                      COLOR(BLU)
A                                  4  4'2=Change'
A                                      COLOR(BLU)
A                                  6  2'Customer numb
A            CID       R        B  6 33REFFLD(CSRCD/C
A  80                                  ERRMSG('Custom
A                                  6 43'Order number
A            ORDNBR         5Y 00  6 74EDTCDE(Z)
```

Figure 301. The CODE Designer development environment

3. In CODE Designer, click the **SCREEN1** tab and select **ORDCTL** from the dropdown menu. Then click the **Customer name field** and finally click the **Web Settings** tab (see Figure ***302***).

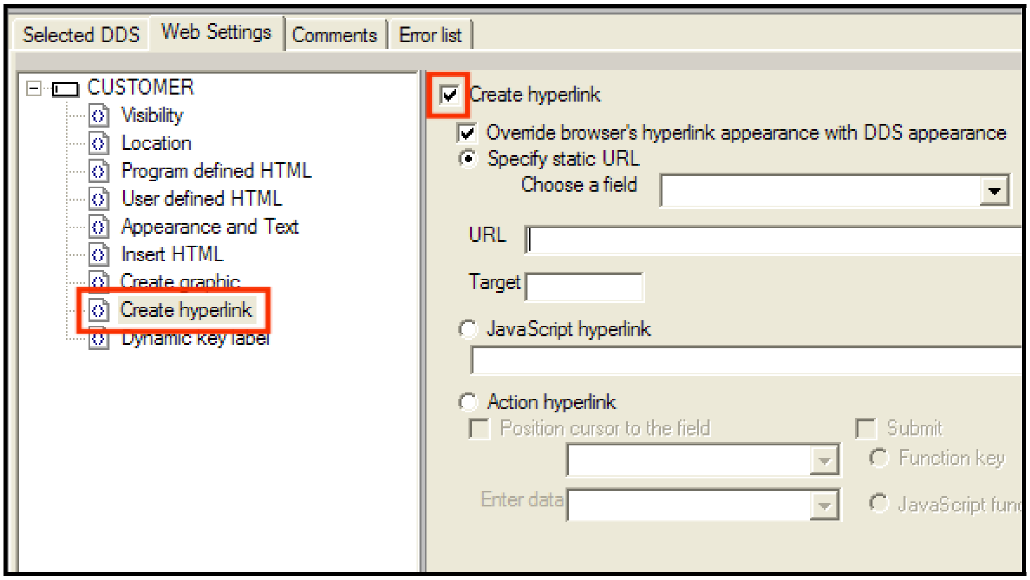


<R0E520B2>WFLABXX/QDDSSRC(ORDENTD) - CODE Designer
File Edit View Insert Selected Tools Help
<R0E520B2>W
SCREEN2
SCREEN1
Details SCREEN2 SCREEN1 Source Listing
ORDCTL
CUSTOMER
30
Ignore all indicators
20
8
0
Selected DDS Web Settings Comments Error list
CUSTOMER
Visibility
Location
Program defined HTML
User defined HTML

```
 7 Customer name . . . . . . . :  000000000000000000000000000000
 8 Address . . . . . . . . . . :  00000000000000000000
 9                                00000000000000000000
10 City  . . . . . . . . . . . :  00000000000000000000   00   0000000000
11
12 Opt Part   Description                                            Qty
13     BBBBBB 000000000000000000000000                               99999
14  9  000000 000000000000000000000000                               66,666
15  9  000000 000000000000000000000000                               66,666
16  9  000000 000000000000000000000000                               66,666
17  9  000000 000000000000000000000000                               66,666
18  9  000000 000000000000000000000000                               66,666
19  9  000000 000000000000000000000000                               66,666
20  9  000000 000000000000000000000000                               66,666
21  9  000000 000000000000000000000000                               66,666
22
23 F3=Exit   F4=Prompt   F6=Accept Order
24
```

Figure 302. CODE Designer prompts towards creating a hyperlink

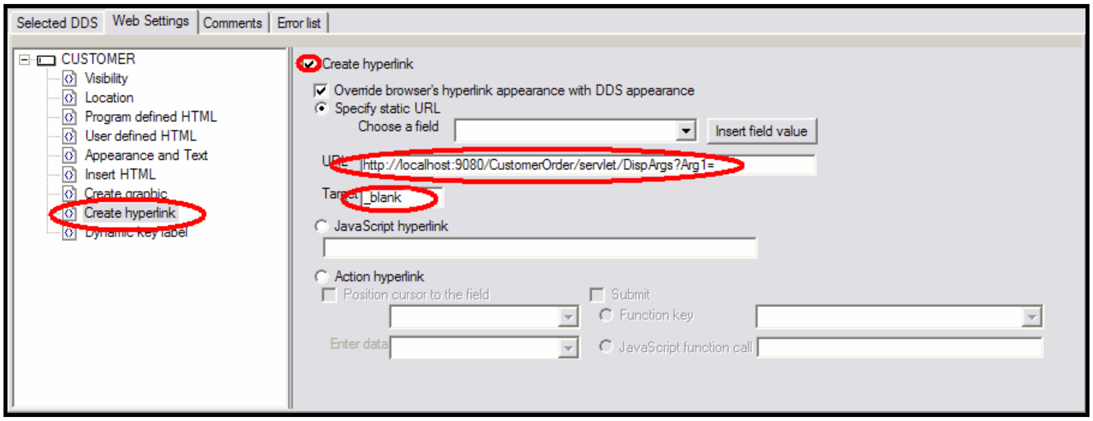
4. In the Web Setting tab, select **Create hyperlink** from the list and then click the **Create hyperlink** check box (see Figure ***303***).



☑ Create hyperlink
☑ Override browser's hyperlink appearance with DDS appearance
● Specify static URL
Choose a field
URL
Target
○ JavaScript hyperlink
○ Action hyperlink

Figure 303. Creating a hyperlink

5. Type `http://localhost:9080/CustomerOrder/servlet/DispArgs?Arg1=` in the **URL** text box and **_blank** in the **Target** text field (see Figure 304).

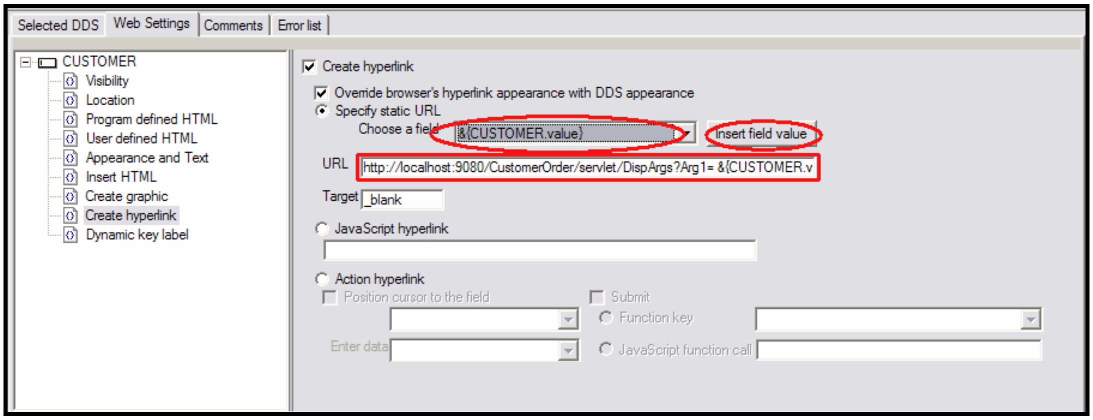


☑ Create hyperlink
☑ Override browser's hyperlink appearance with DDS appearance
● Specify static URL
Choose a field
URL: http://localhost:9080/CustomerOrder/servlet/DispArgs?Arg1=
Target: _blank
○ JavaScript hyperlink
○ Action hyperlink

Figure 304. Type hyperlink URL

6. Earlier, there was mention of the importance of Arg1 and Arg2 in the DisplayArgs.java class file. These parameters are the strings that DisplayArgs.java is looking for to obtain its values. In the **Choose a field** dropdown box select **&{CUSTOMER.value}** and click the **Insert field value** button.

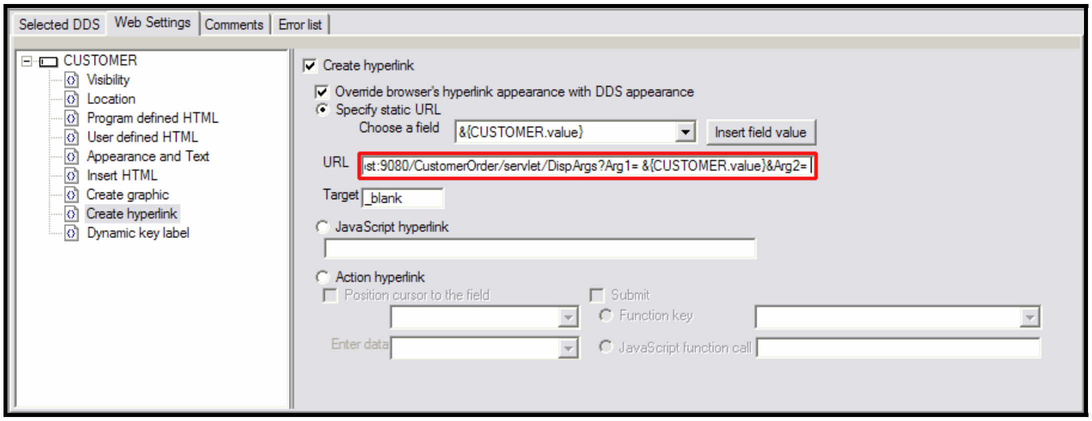
The **URL** string looks similar to the following (see Figure ***305***):
http://localhost:9080/CustomerOrder/servlet/DispArgs?Arg1= &{CUSTOMER.value}



☑ Create hyperlink
☑ Override browser's hyperlink appearance with DDS appearance
● Specify static URL
Choose a field &{CUSTOMER.value}
URL http://localhost:9080/CustomerOrder/servlet/DispArgs?Arg1= &{CUSTOMER.v
Target _blank
○ JavaScript hyperlink
○ Action hyperlink
□ Position cursor to the field
□ Submit
○ Function key
Enter data
○ JavaScript function call

Figure 305. Correct URL string

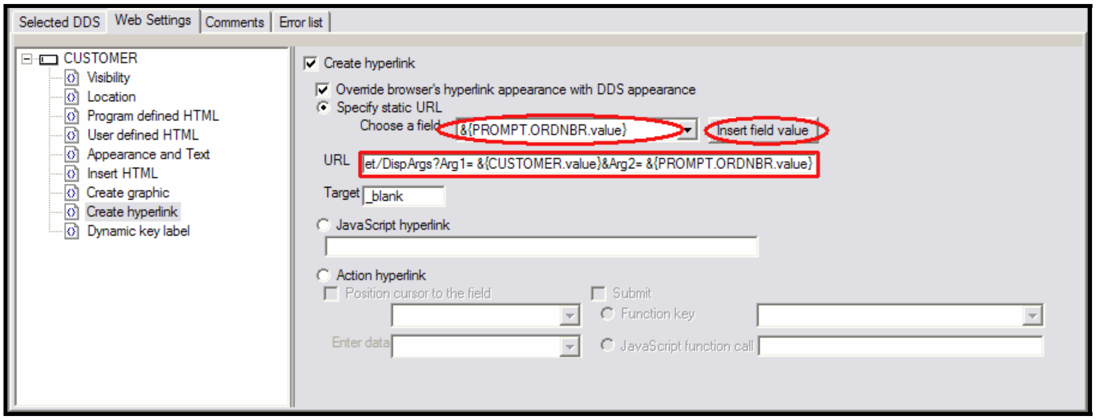
7. Add **&Arg2=** to the end of the **URL** string after the closing **}**. The URL then look similar to: http://localhost:9080/CustomerOrder/servlet/DispArgs?Arg1=&{CUSTOMER.value}&Arg2= (see Figure 306).



☑ Create hyperlink
☑ Override browser's hyperlink appearance with DDS appearance
● Specify static URL
Choose a field &{CUSTOMER.value}
URL ost:9080/CustomerOrder/servlet/DispArgs?Arg1= &{CUSTOMER.value}&Arg2=
Target _blank
○ JavaScript hyperlink
○ Action hyperlink
□ Position cursor to the field
□ Submit
○ Function key
Enter data
○ JavaScript function call

Figure 306. Adding to URL string

8. In the **Chose a field** dropdown box, select &{**PROMPT.ORDNBR.value}** and click the **Insert field value** button. Your URL string now looks similar to: http://localhost:9080/CustomerOrder/servlet/DispArgs?Arg1=&{CUSTOMER.value}&Arg2=&{PROMPT.ORDNBR.value} (see Figure 307).



☑ Create hyperlink

☑ Override browser's hyperlink appearance with DDS appearance

● Specify static URL

Choose a field &{PROMPT.ORDNBR.value}

URL et/DispArgs?Arg1= &{CUSTOMER.value}&Arg2= &{PROMPT.ORDNBR.value}

Target _blank

○ JavaScript hyperlink

○ Action hyperlink

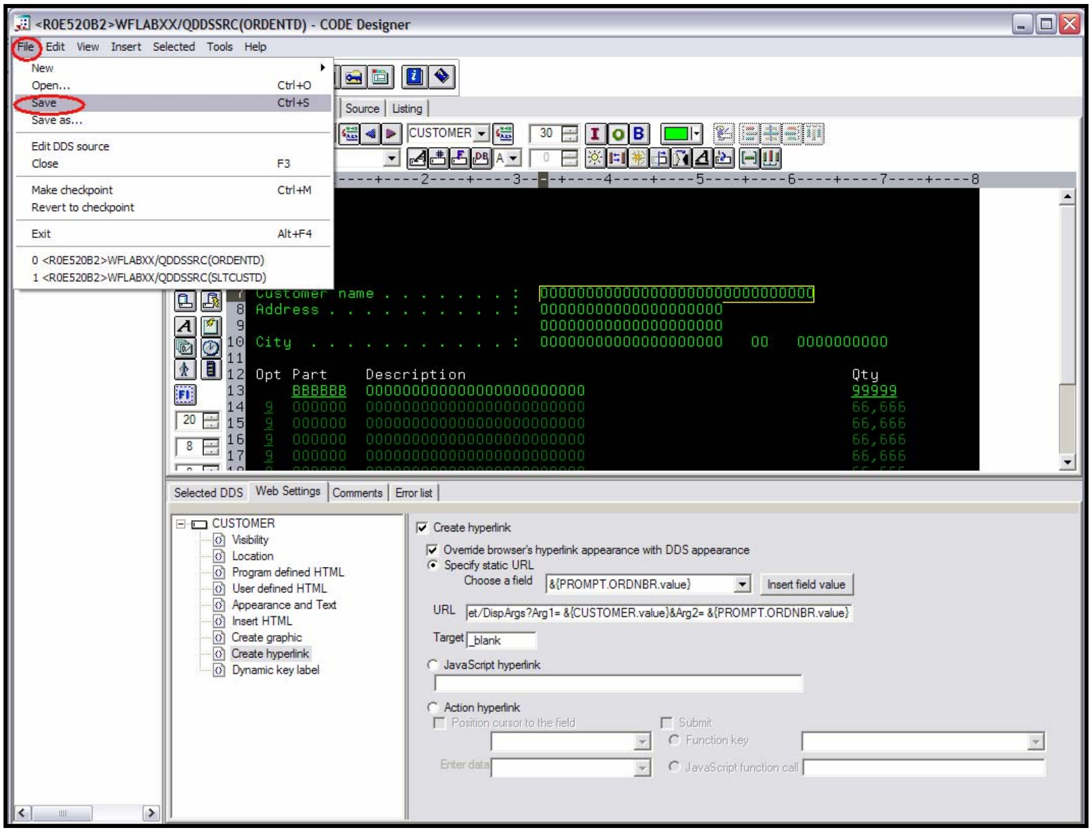
□ Position cursor to the field □ Submit

○ Function key

Enter data ○ JavaScript function call

Figure 307. Adding to URL string

9. Save the changes in CODE Designer by clicking on **File > Save**. Then exit CODE Designer (see
Figure ***308***).



```
 Customer name . . . . . . . :  OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
 Address . . . . . . . . . . :  OOOOOOOOOOOOOOOOOOOO
                                OOOOOOOOOOOOOOOOOOOO
 City  . . . . . . . . . . . :  OOOOOOOOOOOOOOOOOOOO   OO   OOOOOOOOOO

 Opt Part     Description                                       Qty
     BBBBBB   OOOOOOOOOOOOOOOOOOOOOOOOOOOOOO                    99999
  9  OOOOOO   OOOOOOOOOOOOOOOOOOOOOOOOOOOOOO                    66,666
  9  OOOOOO   OOOOOOOOOOOOOOOOOOOOOOOOOOOOOO                    66,666
  9  OOOOOO   OOOOOOOOOOOOOOOOOOOOOOOOOOOOOO                    66,666
  9  OOOOOO   OOOOOOOOOOOOOOOOOOOOOOOOOOOOOO                    66,666
```

☑ Create hyperlink

☑ Override browser's hyperlink appearance with DDS appearance

● Specify static URL

Choose a field &{PROMPT.ORDNBR.value}

URL et/DispArgs?Arg1= &{CUSTOMER.value}&Arg2= &{PROMPT.ORDNBR.value}

Target _blank

○ JavaScript hyperlink

○ Action hyperlink

□ Position cursor to the field □ Submit

○ Function key

Enter data ○ JavaScript function call

Figure 308. Save changes

10. Now that you have made changes to ORDENTD, it needs to be reconverted. In the WebFacing Perspective, right-click ORDENTD and select **Convert**. This needs to be done so the files can be generated to call the CustomerOrder application when the customer name hyperlink is clicked (see
Figure ***309***).

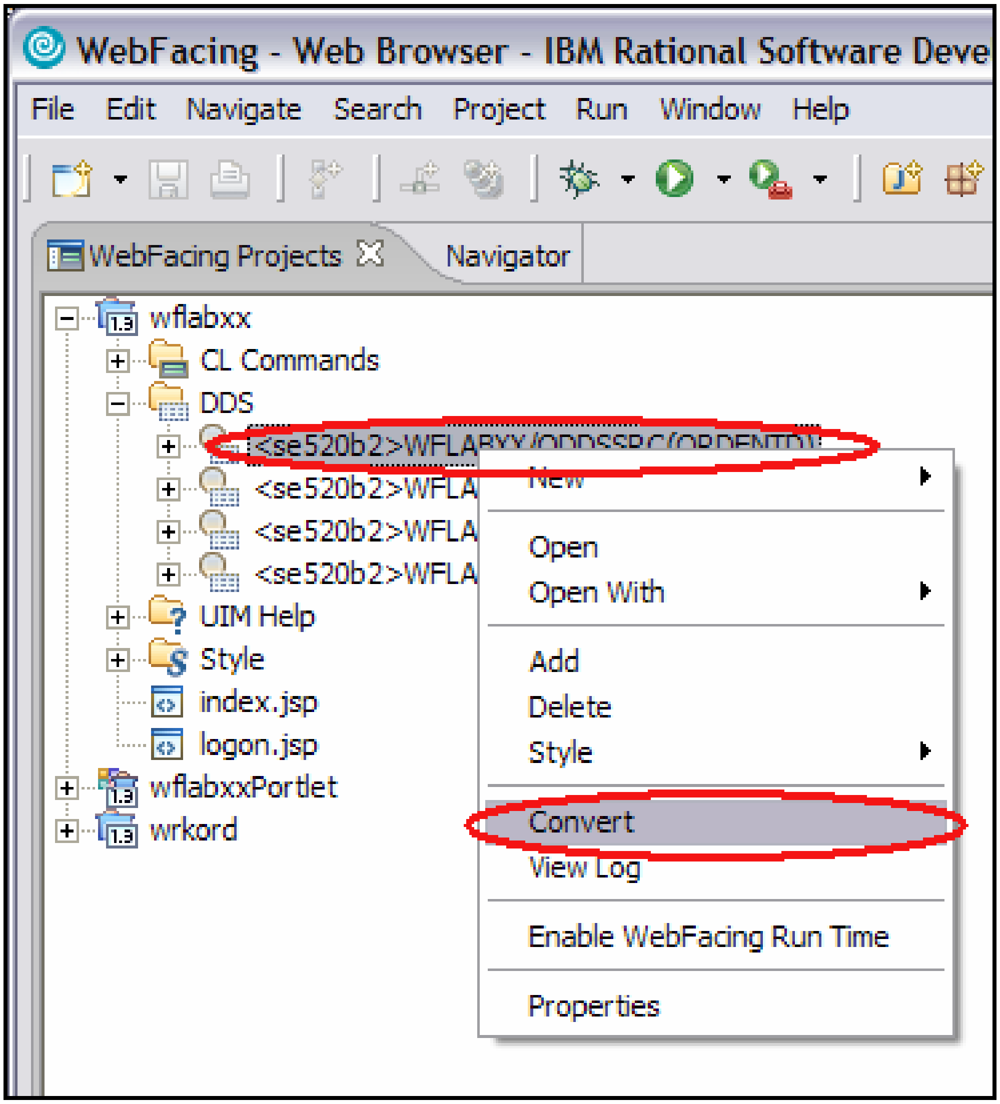
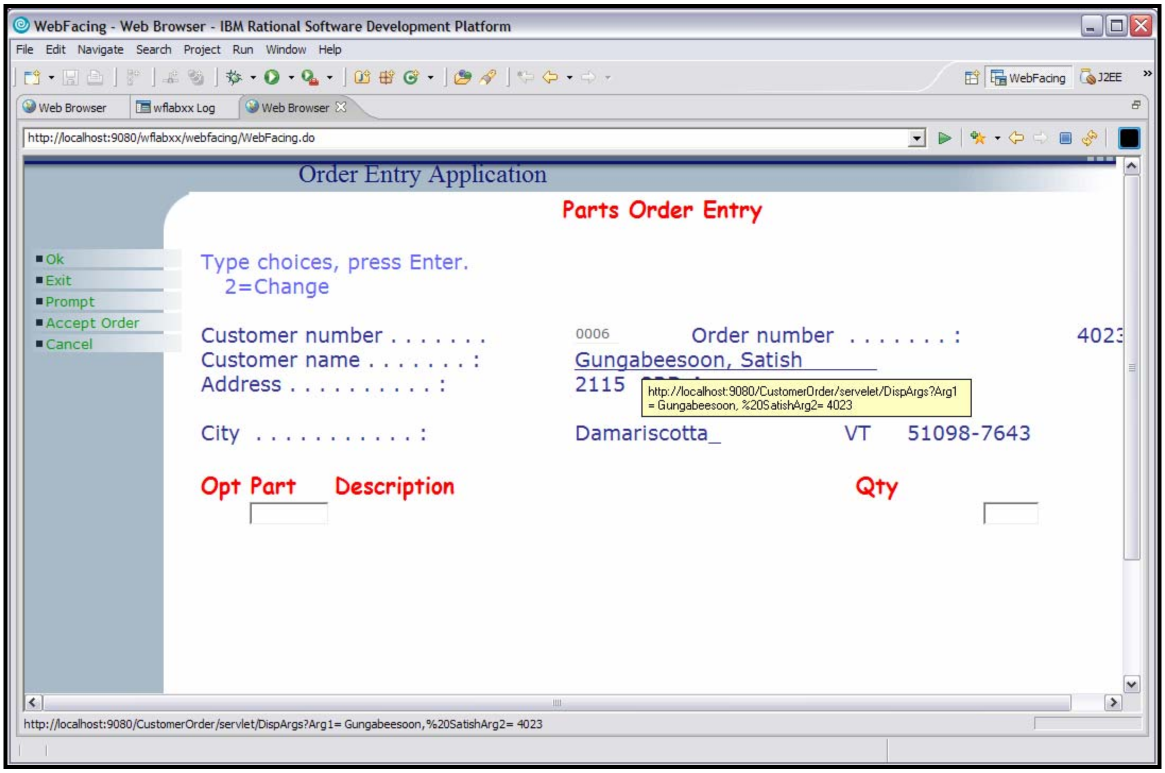


Figure 309. Changes made to ORDENTD need to be reconverted

Exercise 16.4: Test the integration of the Java application

Now that you have imported our Java application and modified our DDS source using Web Settings in CODE Designer, you need to test our WebFacing application.

1. In the WebFacing Perspective, right-click your WFLabxx project and select **Run on Server**. Launch the application, prompt for and select a customer. Run the WebFacing application and select a customer. Notice on the next screen, the Customer Name field is now a hot link. Placing a cursor over that hot link displays in the status window of the URL that is called upon clicking (see
Figure ***310***).



http://localhost:9080/wflabxx/webfacing/WebFacing.do

Order Entry Application

Parts Order Entry

Type choices, press Enter.
2=Change

Customer number . . . . 0006 Order number . . . . : 4023
Customer name . . . . : Gungabeesoon, Satish
Address . . . . : 2115

http://localhost:9080/CustomerOrder/servelet/DispArgs?Arg1 = Gungabeesoon, %20SatishArg2= 4023

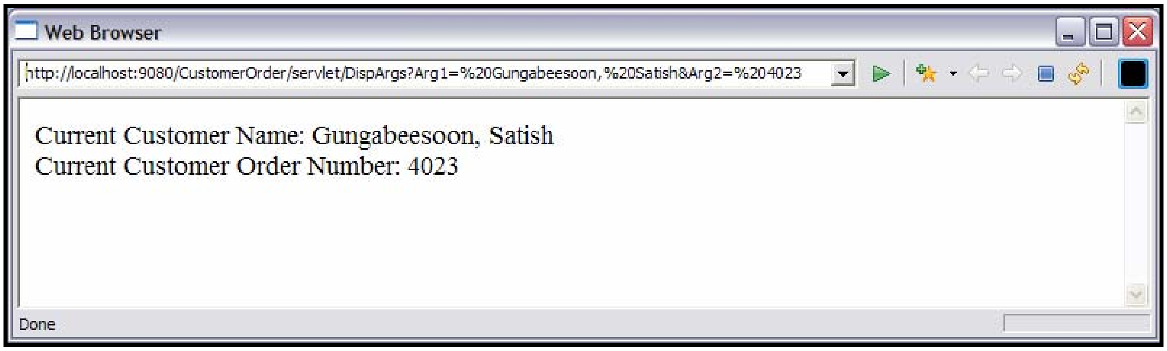
City . . . . : Damariscotta_ VT 51098-7643

Opt Part Description Qty

http://localhost:9080/CustomerOrder/servlet/DispArgs?Arg1= Gungabeesoon,%20SatishArg2= 4023

Figure 310. Testing the integration of the Java application

2. In this example, the CustomerOrder/serlet/DispArgs servlet is called and two arguments are passed, Arg1 and Arg2. Clicking the hot ink calls **DispArgs** and displays the results in a new window (see Figure 311).



http://localhost:9080/CustomerOrder/servlet/DispArgs?Arg1=%20Gungabeesoon,%20Satish&Arg2=%204023

Current Customer Name: Gungabeesoon, Satish
Current Customer Order Number: 4023

Figure 311. Results displayed

Recap

You have completed “Invoking a Java program from the IBM WebFacing Tool.” This simple example illustrates a powerful and simple way to provide new functionality in your WebFacing application. You now have the iformation to understand how to:

- Import an EAR file into WebSphere Development Suite Client
- Deploy an application to the same test server as your WebFacing project
- Change Web Settings to call another program

Invoking a System i program from a WebFaced Web page

In this chapter, you learn how to invoke a System i program from a hyperlink on a WebFaced Web page. The hyperlink is on a part name field. Clicking the link invokes an RPG program, passing it the part number value for this part name. The RPG service program retrieves the data for this part and display it on another Web page in a separate browser window.

To invoke the RPG service program, use the **Web Tools Web Interaction** feature of WebSphere Development Studio Client. This feature allows you to define input and output pages and then link fields on those pages to parameters of the System i host program. The wizard generates all of the code required to map input fields that are contained on the input page to the parameters that are used as input on the System i program. The wizard also maps the output field values from the System i program to the fields defined on the output page. As well, the code required to invoke the program is generated.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 17.1: Creating a new dynamic Web project
- Exercise 17.2: Defining the System i information
- Exercise 17.3: Running the Web Interaction wizard
- Exercise 17.4: Test the Web interaction
- Exercise 17.5: Creating a JavaScript hyperlink using the Web settings
- Exercise 17.6: Creating a new JavaScript function to invoke the Web interaction
- Exercise 17.7: Test the changed application

The exercises within each chapter must be completed in order. Start with “Exercise 17.1: Creating a new Dynamic Web Project” when you are ready to begin.

Length of time

This chapter takes approximately 45 minutes to complete.

Exercise 17.1: Creating a new Dynamic Web project

To begin, you create a new Web project that is used to invoke the System i program. You can use the existing WebFacing project, but by creating a separate project, you make this new function available to any other WebFaced project.

To create a new dynamic Web project:

1. Click **File > New > Project** on the workbench menu (see Figure ***312***).

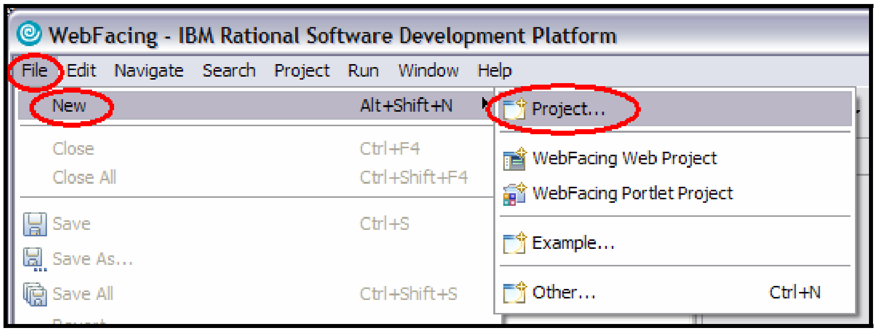


Figure 312. Creating a new Dynamic Web project

2. The New Project wizard dialog is displayed (see Figure ***313***):

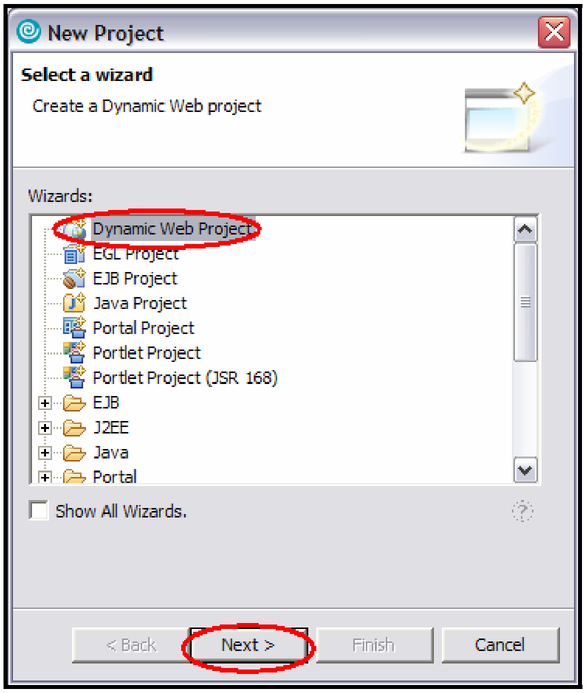
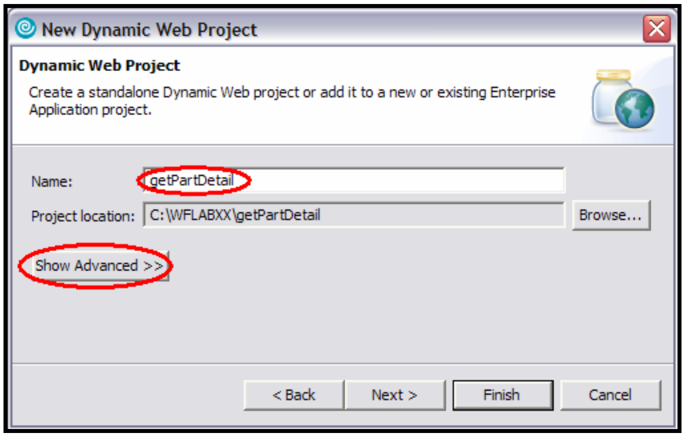


Figure 313. The New Project wizard dialog

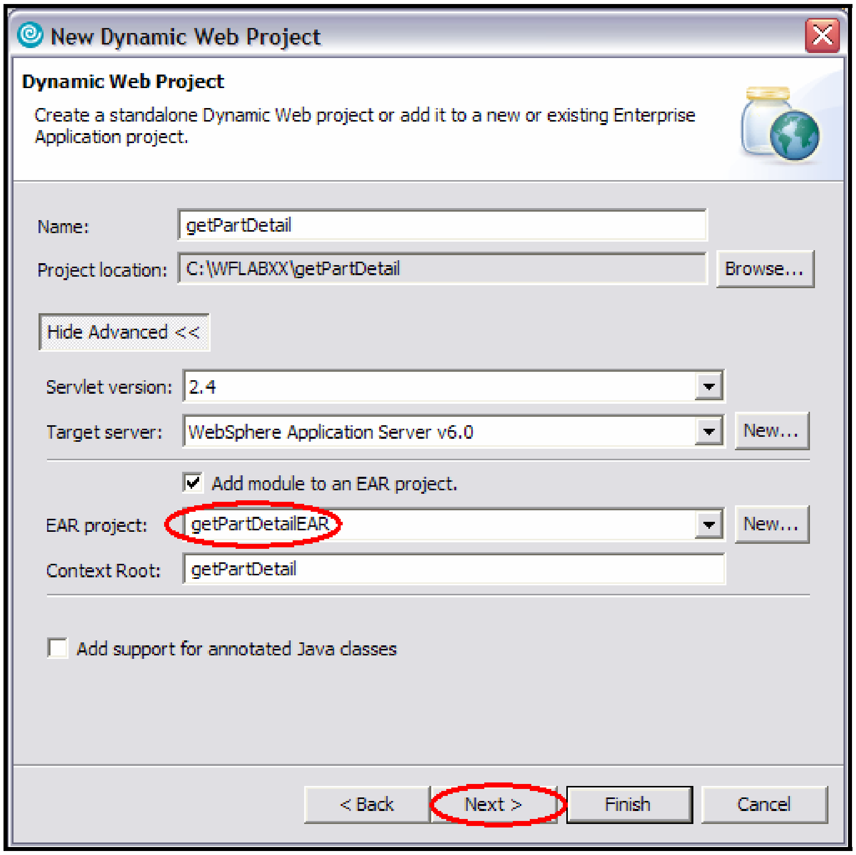
3. Select **Dynamic Web Project**.
4. Click **Next**.
5. The initial page of the Dynamic Web project wizard is displayed (see Figure ***314***.


New Dynamic Web Project
Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application project.
Show Advanced >>
< Back
Next >
Finish
Cancel

Name: getPartDetail
Project location: C:\WFLABXX\getPartDetail Browse...

Figure 314. The initial page of the Dynamic Web project wizard

6. In the **Project name** field, type `getPartDetail` (see Figure **315**).
7. Click **Show Advanced >>** to display additional fields.



New Dynamic Web Project

Dynamic Web Project

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

Name: getPartDetail

Project location: C:\WFLABXX\getPartDetail

Servlet version: 2.4

Target server: WebSphere Application Server v6.0

☑ Add module to an EAR project.

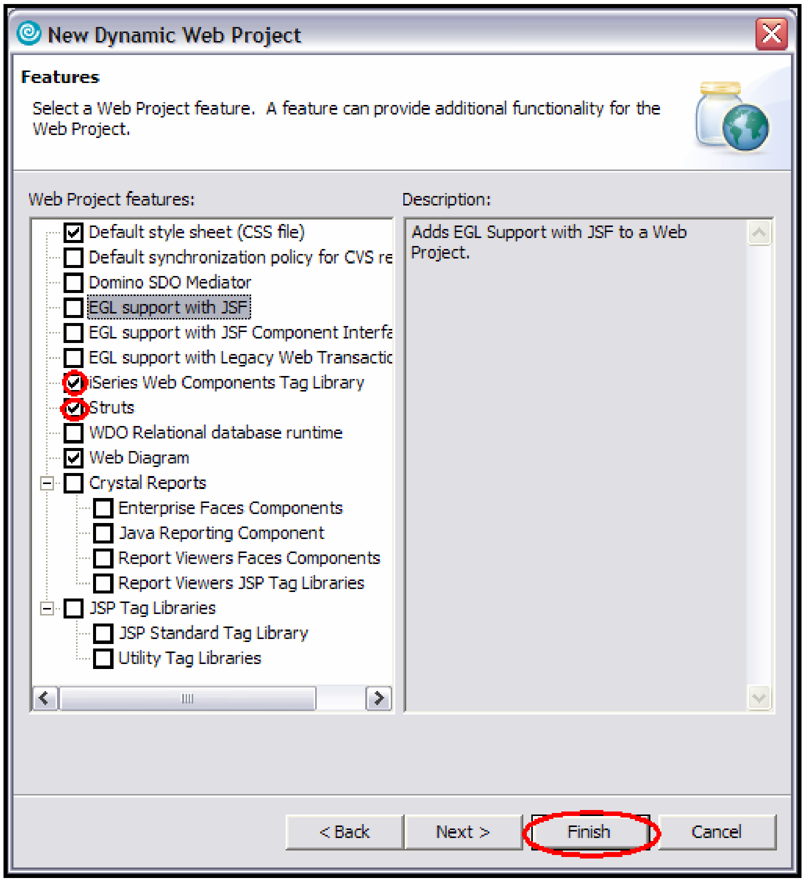
EAR project: getPartDetailEAR

Context Root: getPartDetail

□ Add support for annotated Java classes

Figure 315. In the Project name field, type getPartDetail.

8. In the **EAR project** field, you see that the value **getPartDetailEAR** has been filled in automatically for you.
9. Click **Next.**
10. The Features Page is displayed (see Figure ***316***).



New Dynamic Web Project

Features

Select a Web Project feature. A feature can provide additional functionality for the Web Project.

Web Project features:

- ☑ Default style sheet (CSS file)
- □ Default synchronization policy for CVS re
- □ Domino SDO Mediator
- □ EGL support with JSF
- □ EGL support with JSF Component Interfa
- □ EGL support with Legacy Web Transactic
- ☑ iSeries Web Components Tag Library
- ☑ Struts
- □ WDO Relational database runtime
- ☑ Web Diagram
- □ Crystal Reports
  - □ Enterprise Faces Components
  - □ Java Reporting Component
  - □ Report Viewers Faces Components
  - □ Report Viewers JSP Tag Libraries
- □ JSP Tag Libraries
  - □ JSP Standard Tag Library
  - □ Utility Tag Libraries

Description:

Adds EGL Support with JSF to a Web Project.

Figure 316. The Features Page

11. Some Features are already selected by default. In addition to these defaults, select the **iSeries Web Components Tag Library** and **Struts** check boxes.
12. Click **Finish**.

If you are prompted to change Perspectives, click **OK**. When the project has been created, you are back in the workbench where you see your new project in the Project Explorer view of the Web perspective (see
Figure ***317***).

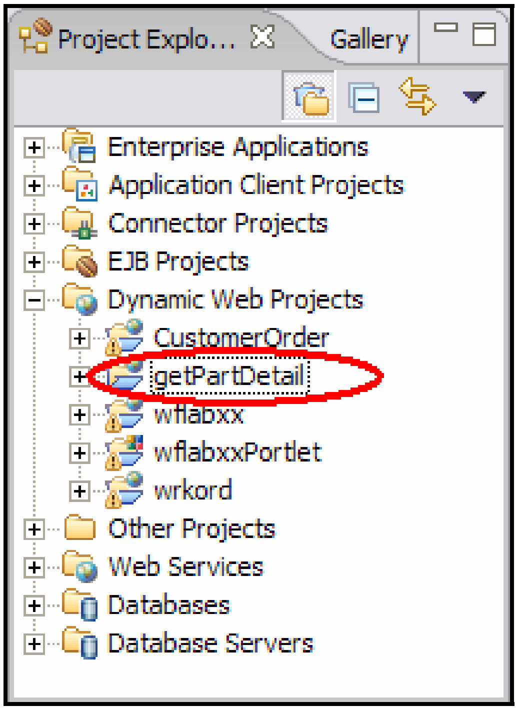


Figure 317. New project displayed in the Project Explorer view

Exercise 17.2: Defining System i configuration information

Because this project is used to invoke an RPG program, you need to define information about the particular System i model upon which the program runs, such as: server name and user profile.

To define System i configuration information:

1. Right-click the **getPartDetail** project in the Navigator view (see Figure *318*).

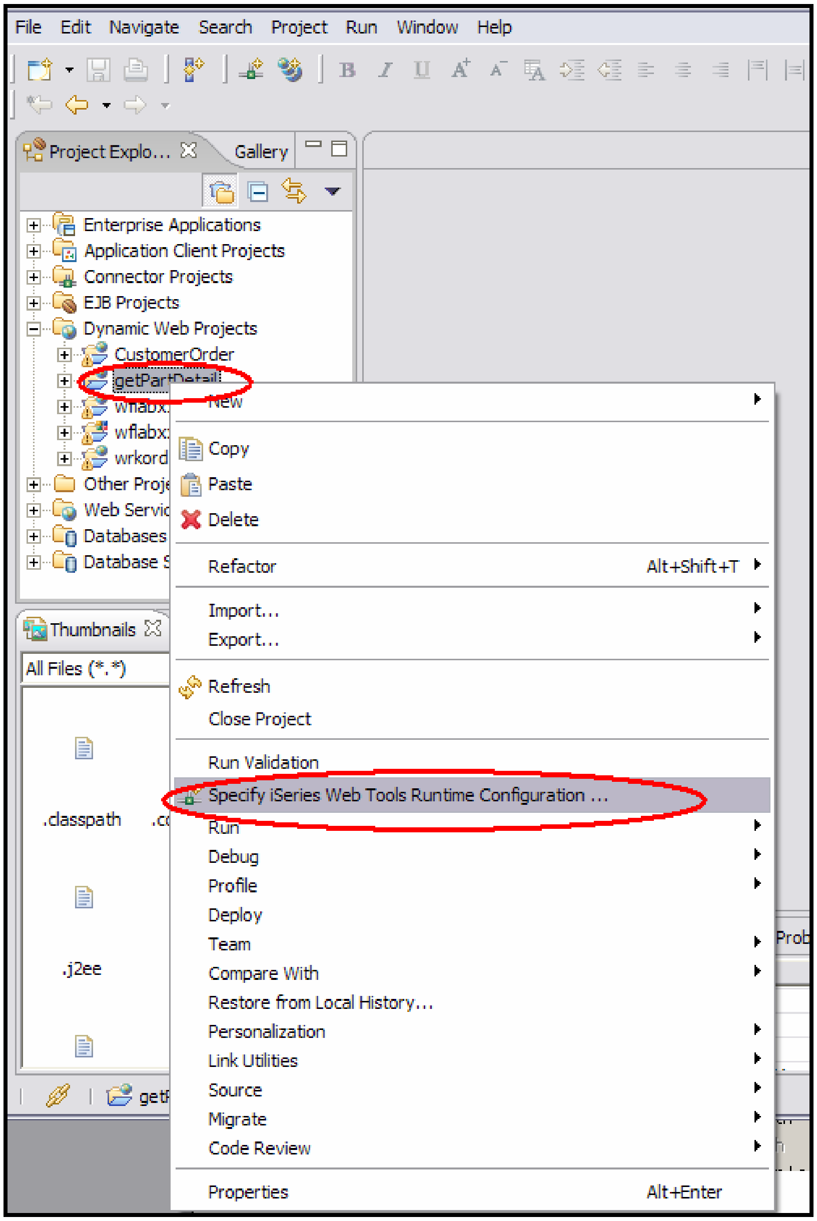
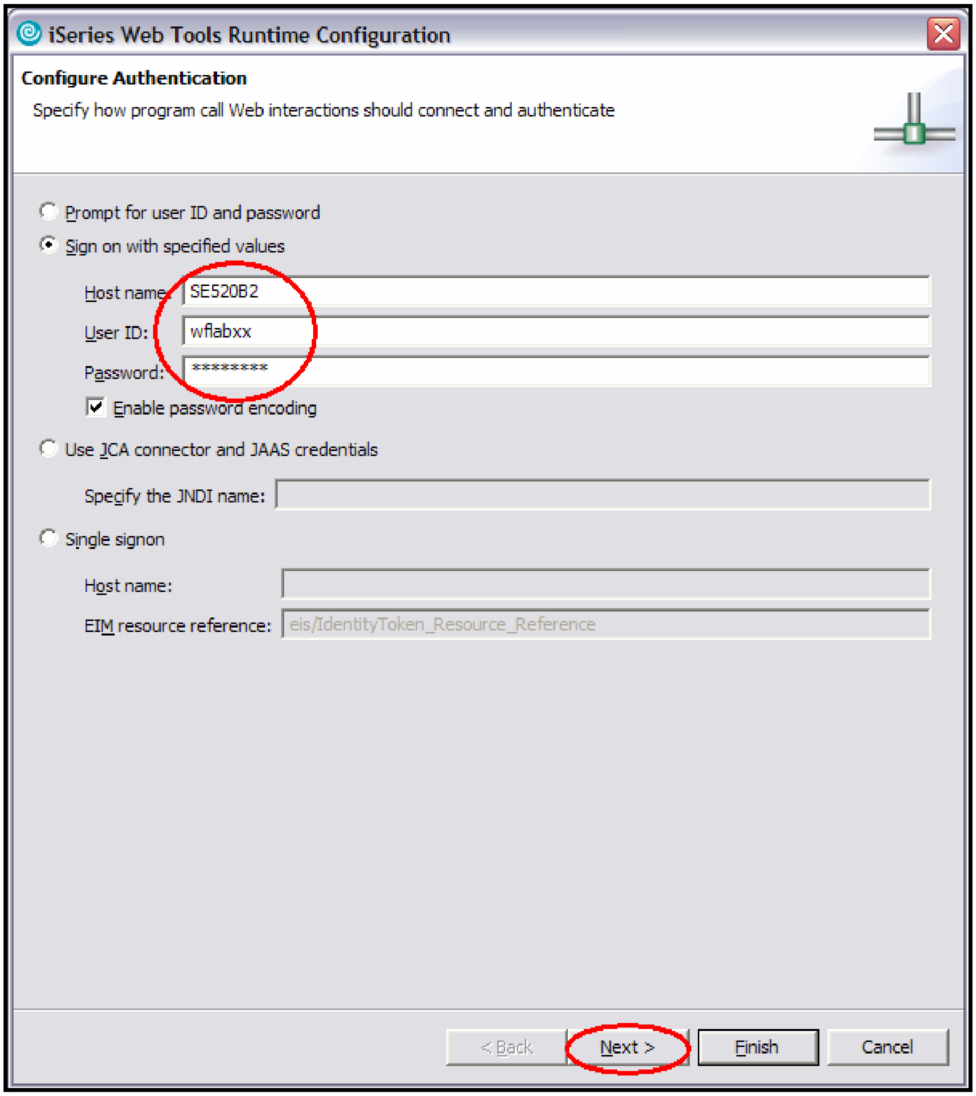


Figure 318. Defining System i configuration information

2. Click **Specify iSeries Web Tools Runtime Configuration** on the pop-up menu. The iSeries Web Tools Runtime Configuration wizard opens (see Figure **319**).



iSeries Web Tools Runtime Configuration

Configure Authentication

Specify how program call Web interactions should connect and authenticate

○ Prompt for user ID and password

● Sign on with specified values

Host name: SE520B2

User ID: wflabxx

Password: *********

☑ Enable password encoding

○ Use JCA connector and JAAS credentials

Specify the JNDI name:

○ Single signon

Host name:

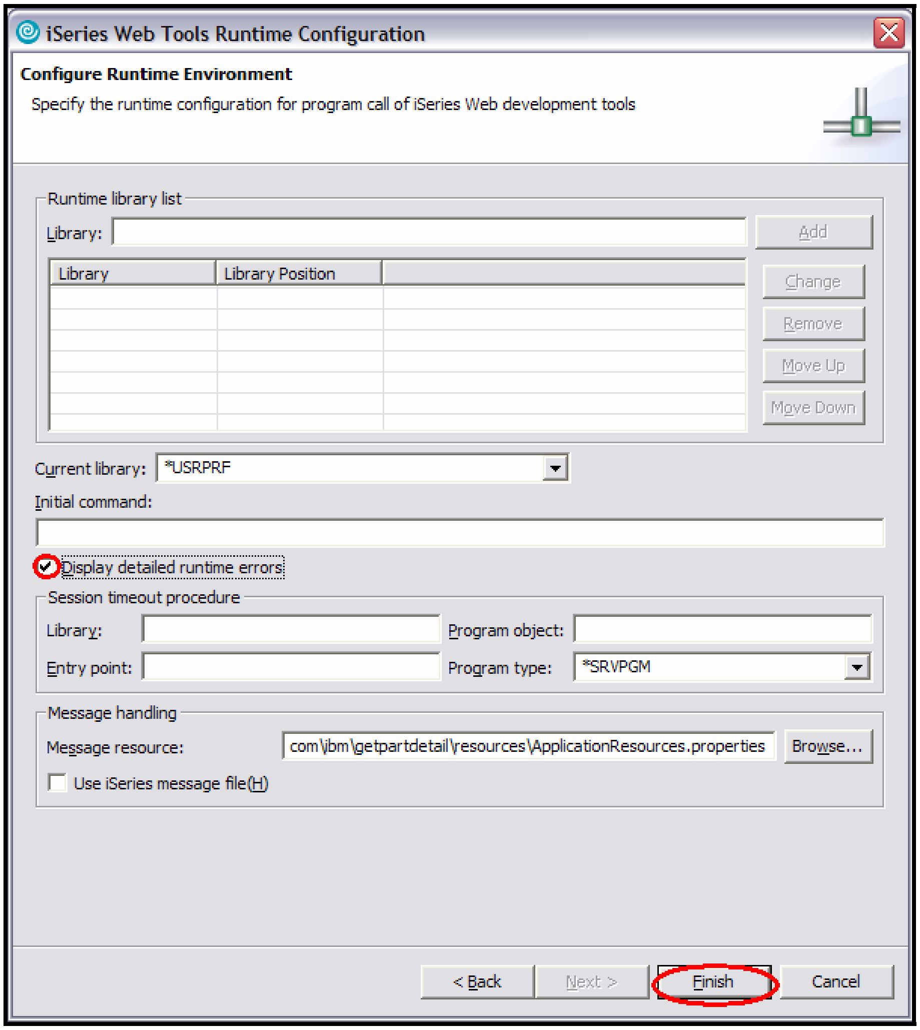
EIM resource reference: eis/IdentityToken_Resource_Reference

Figure 319. The iSeries Web Tools Runtime Configuration wizard

The **Configure Authentication** page of the wizard is used to indicate which System i model to use, as well as how the system is to authenticate the user.

3. In the **Host** name field, type the name of the host system, for example s400a.
4. In the **User ID** and **Password** fields, type the appropriate values.
5. Click **Next**.

6. The Configure Run Time page is displayed (see Figure ***320***).



iSeries Web Tools Runtime Configuration

Configure Runtime Environment

Specify the runtime configuration for program call of iSeries Web development tools

Runtime library list

Library:

Library	Library Position	

Current library: *USRPRF

Initial command:

☑ Display detailed runtime errors

Session timeout procedure

Library:

Program object:

Entry point:

Program type: *SRVPGM

Message handling

Message resource: com\ibm\getpartdetail\resources\ApplicationResources.properties

□ Use iSeries message file(H)

Figure 320. The Configure Run Time page

7. Select the check box, **Display detailed run-time errors**. This results in a detailed error page being displayed if a run-time error occurs.
8. Click **Finish** to define the runtime information for this project.

Exercise 17.3: Running the Web Interaction wizard

You now use the **Web Interaction Wizard** feature of WebSphere Development Studio Client. A Web Interaction is the flow from an input page to an action to be performed, then to an output page. In your case, the input page prompts for a part number, the action to be performed is an RPG service program procedure and the output page is a Web page with part detail. The wizard handles collecting data from the input page, and passing it as parameters to the service program procedure. When the procedure ends, the output parameters are mapped to elements on the output page, which is then sent to the browser.

For this chapter, you do not use an input page, even though one is created by the wizard for you. The input for this interaction comes from a value on a hyperlink in the page created by the IBM WebFacing Tool. However, you do use the input page to test the Web Interaction.

To run the Web Interaction wizard:

1. Select the **getPartDetail** project, then click the Web Interaction Wizard icon on the toolbar (see
   Figure ***321***):

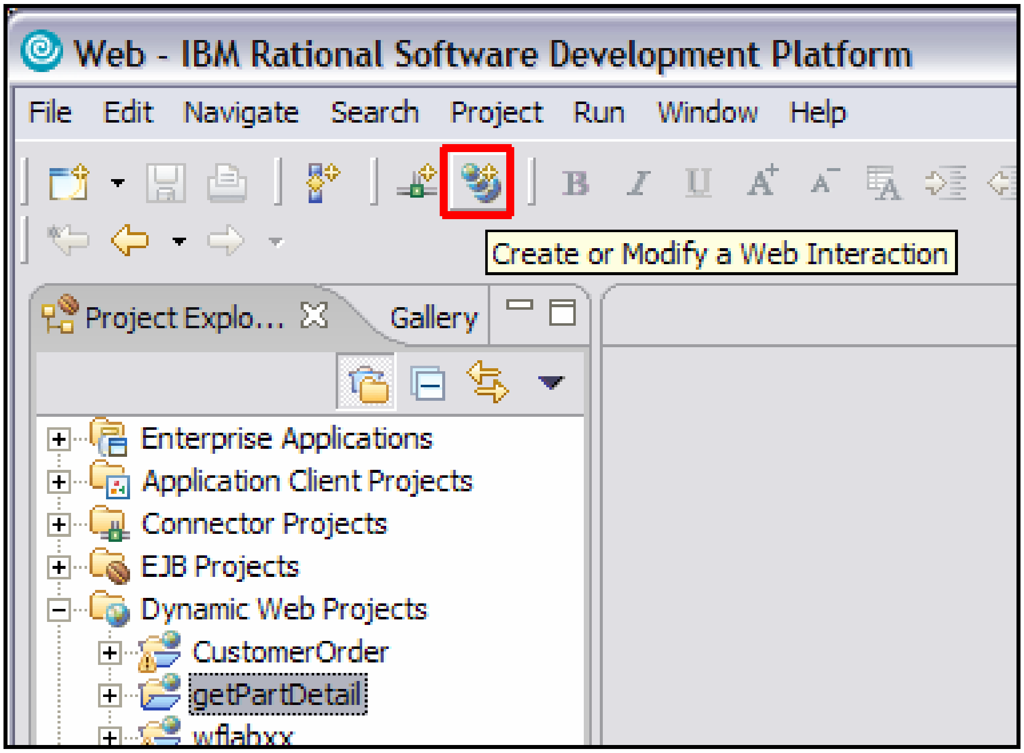
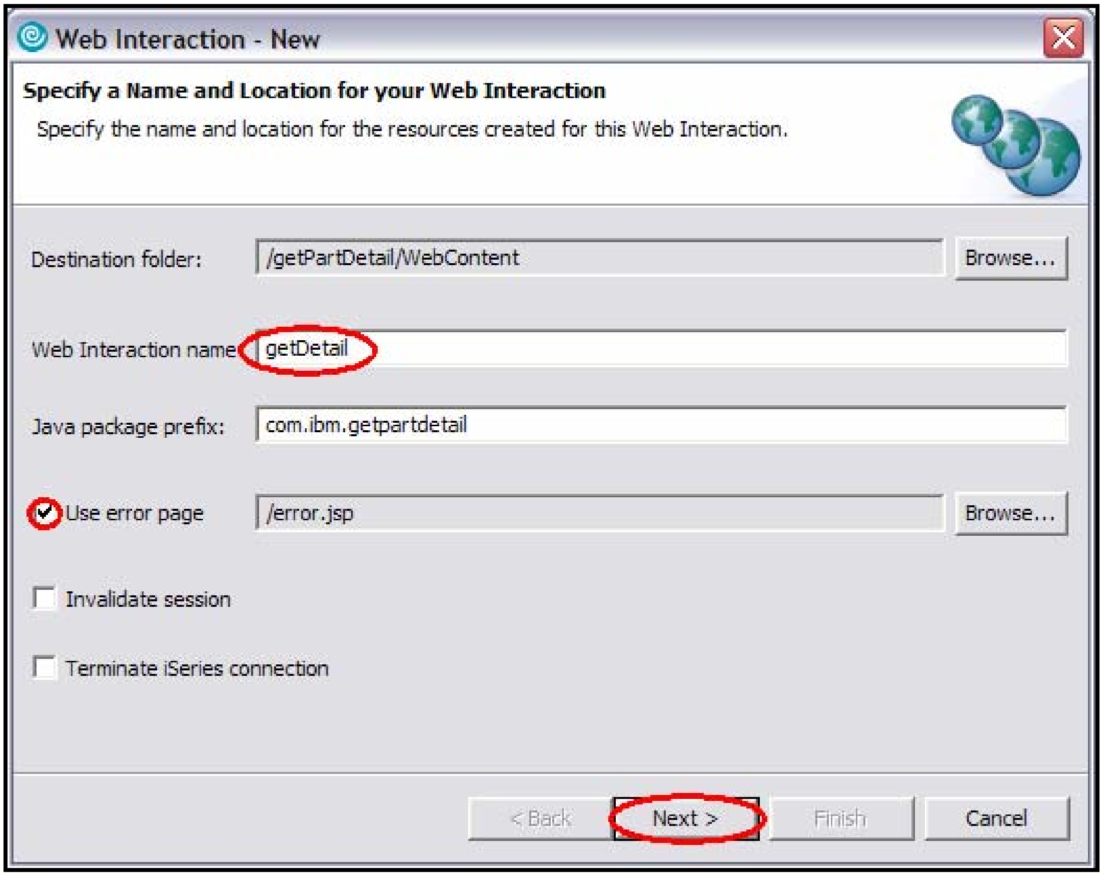


Figure 321. Select the getPartDetail project, then select the Web Interaction Wizard icon.

2. The first page of the Web Interaction Wizard is displayed (see Figure 322).



Specify a Name and Location for your Web Interaction

Specify the name and location for the resources created for this Web Interaction.

Destination folder: /getPartDetail/WebContent

Web Interaction name: getDetail

Java package prefix: com.ibm.getpartdetail

☑ Use error page: /error.jsp

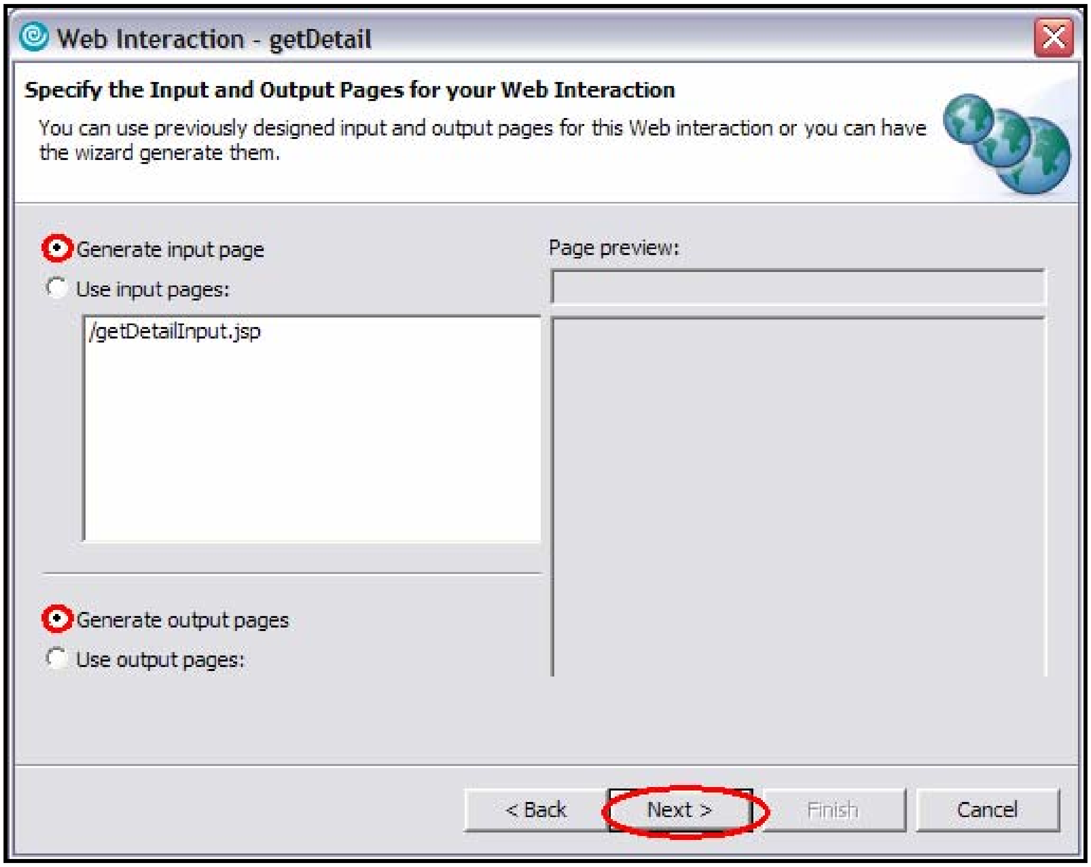
□ Invalidate session

□ Terminate iSeries connection

Figure 322. The first page of the Web Interaction Wizard

3. For the **Web Interaction name** field, type `getDetail`.
4. Ensure the **Use error page** box is checked.
5. Click **Next**.

The “Specify the Input and Output pages for your Web Interaction” screen is displayed (see Figure 323).



Web Interaction - getDetail

Specify the Input and Output Pages for your Web Interaction

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

● Generate input page

○ Use input pages:

/getDetailInput.jsp

Page preview:

● Generate output pages

○ Use output pages:

Figure 323. The Specify the Input and Output pages

6. Select the radio button labeled **Generate input page.**
7. Select the radio button labeled **Generate output page.**
8. Click **Next.**

Defining the System i procedure to invoke

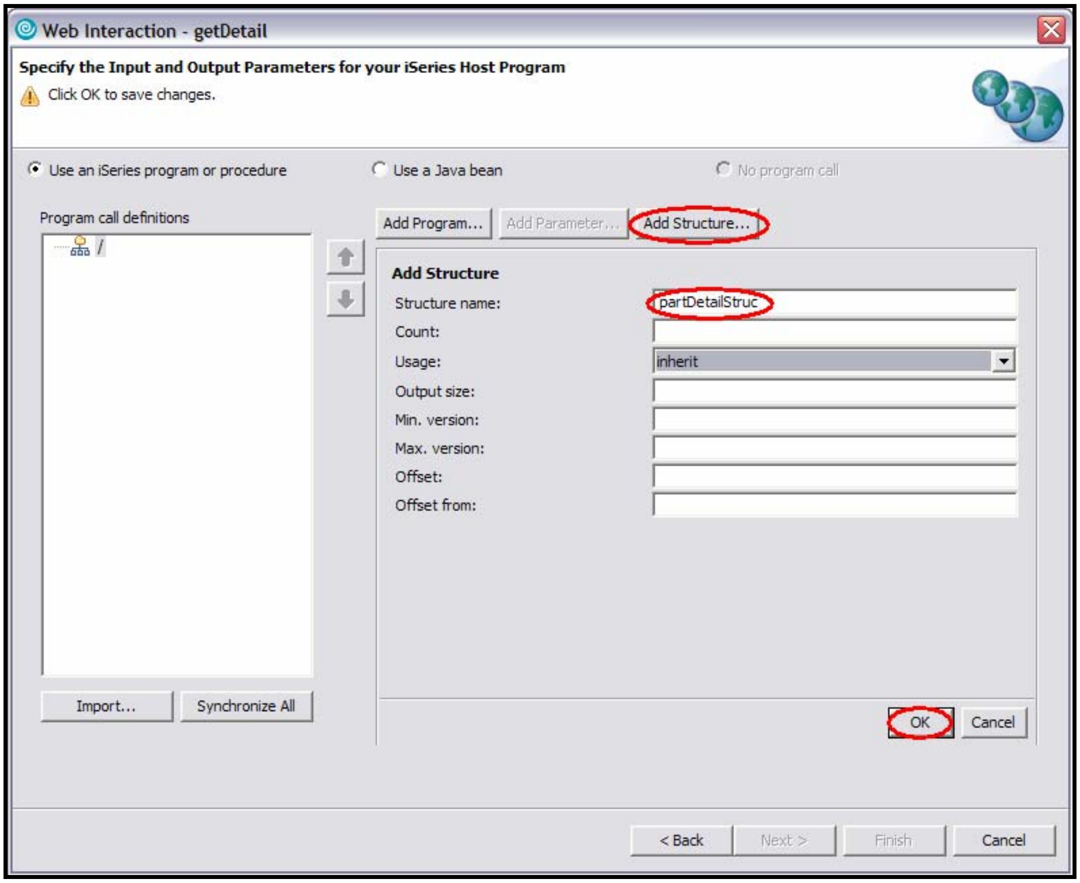
At this point, you have given this interaction a name, and indicated that you want the wizard to generate the input and the output pages. You must now define the information about the RPG procedure to be invoked including the name of the service program and procedure as well as define the parameters for the procedure.

This procedure has one input parameter (the part number) and one output parameter that is a structure of the various fields that provide part-related information.

You begin by first defining the return structure containing the part information.

To define the structure:

1. On the Specify Input and Output Parameters for your iSeries Host Program page of the wizard, click the **Add Structure** button. The Add Structure panel opens on the right side of the wizard (see
Figure ***324***).



Web Interaction - getDetail

Specify the Input and Output Parameters for your iSeries Host Program

Click OK to save changes.

● Use an iSeries program or procedure ○ Use a Java bean ○ No program call

Add Structure

Structure name: partDetailStruc

Count:

Usage: inherit

Output size:

Min. version:

Max. version:

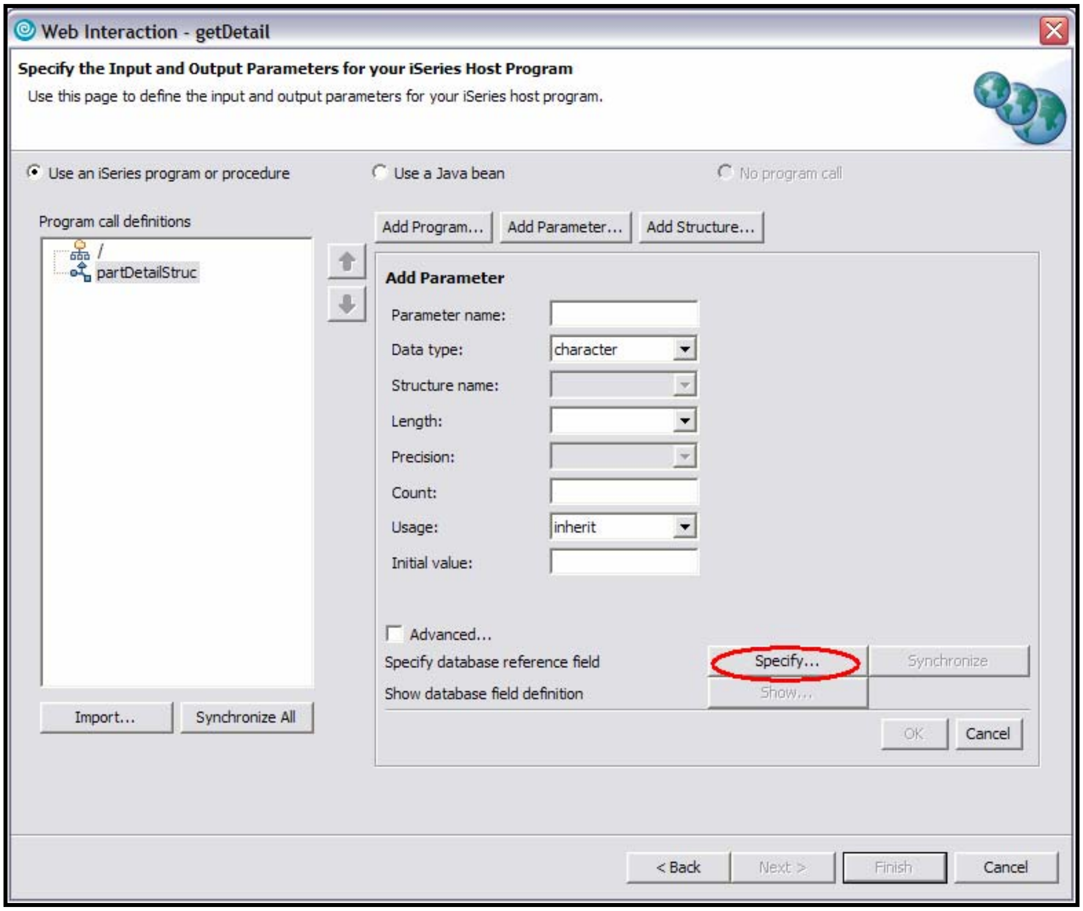
Offset:

Offset from:

Figure 324. The Add Structure panel

2. Type **partDetailStruc** in the **Structure name** field.
3. Click **OK** to add this structure to the Web Interaction.

4. The structure is added in the Program call definitions section on the left pane, and the Add Parameter pane is automatically displayed on the right side of the wizard page (see Figure 325).



Web Interaction - getDetail

Specify the Input and Output Parameters for your iSeries Host Program

Use this page to define the input and output parameters for your iSeries host program.

● Use an iSeries program or procedure ○ Use a Java bean ○ No program call

Program call definitions

- /
  - partDetailStruc

Add Parameter

Parameter name:

Data type: character

Structure name:

Length:

Precision:

Count:

Usage: inherit

Initial value:

□ Advanced...

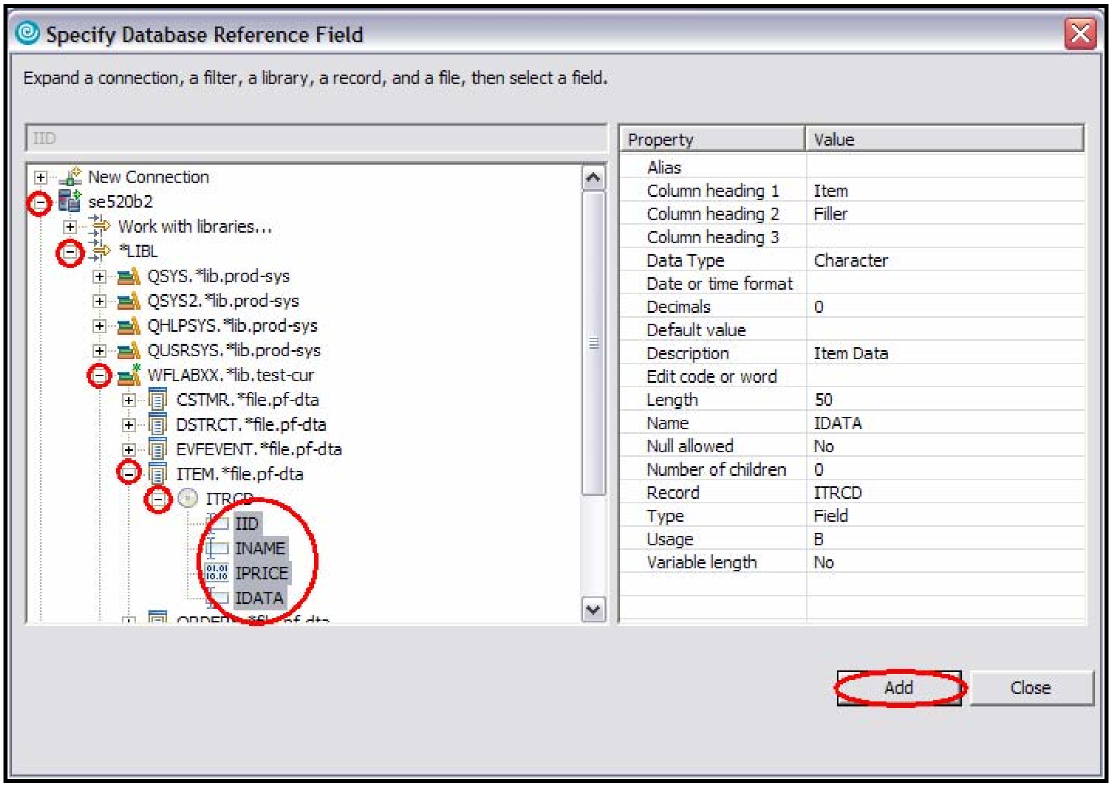
Specify database reference field

Show database field definition

Figure 325. The Add Parameter pane

5. At this point, you can click the **Add Parameter** button to add parameters to this structure. However, in this case, the parameters you need are defined in the two physical files, STOCK and ITEM. Therefore, you can use the Specify database reference field feature to define the structure parameters.
6. Click the **Specify** button.

7. The Specify Database Reference Field dialog is displayed (see Figure 326).



Specify Database Reference Field

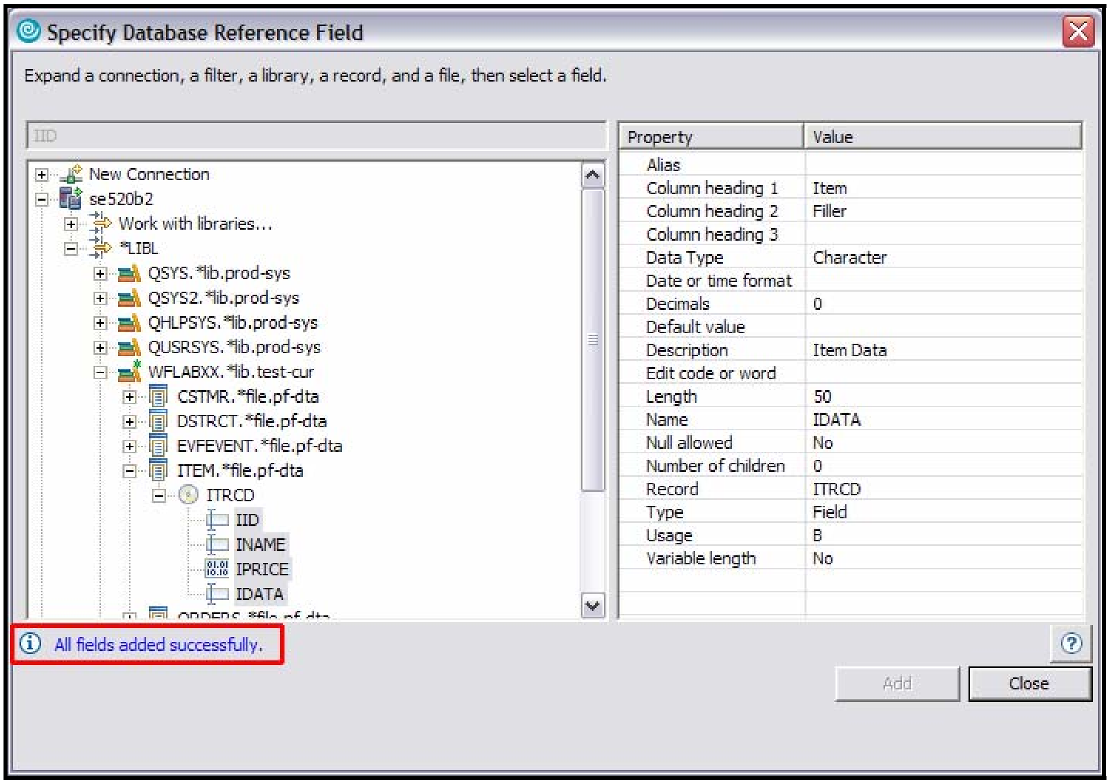
Expand a connection, a filter, a library, a record, and a file, then select a field.

Property	Value
Alias	
Column heading 1	Item
Column heading 2	Filler
Column heading 3	
Data Type	Character
Date or time format	
Decimals	0
Default value	
Description	Item Data
Edit code or word	
Length	50
Name	IDATA
Null allowed	No
Number of children	0
Record	ITRCD
Type	Field
Usage	B
Variable length	No

Figure 326. The Specify Database Reference Field

8. Expand the system connection name (**s400a**).
9. Expand the ***LIBL** list by clicking on the plus sign (+).
10. Expand the **WFLABXX** library.
11. Expand the **ITEM** file and then the **ITRCD** record format to see all of the fields in this format
12. You use all of the fields in this format. With the **Ctrl** key pressed, use the mouse to select the fields as shown in the figure and then press the **Add** button to add these fields to our structure.

13. A confirmation message appears in the dialog window, stating **All fields added successfully.** (see
Figure ***327***).



Specify Database Reference Field

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

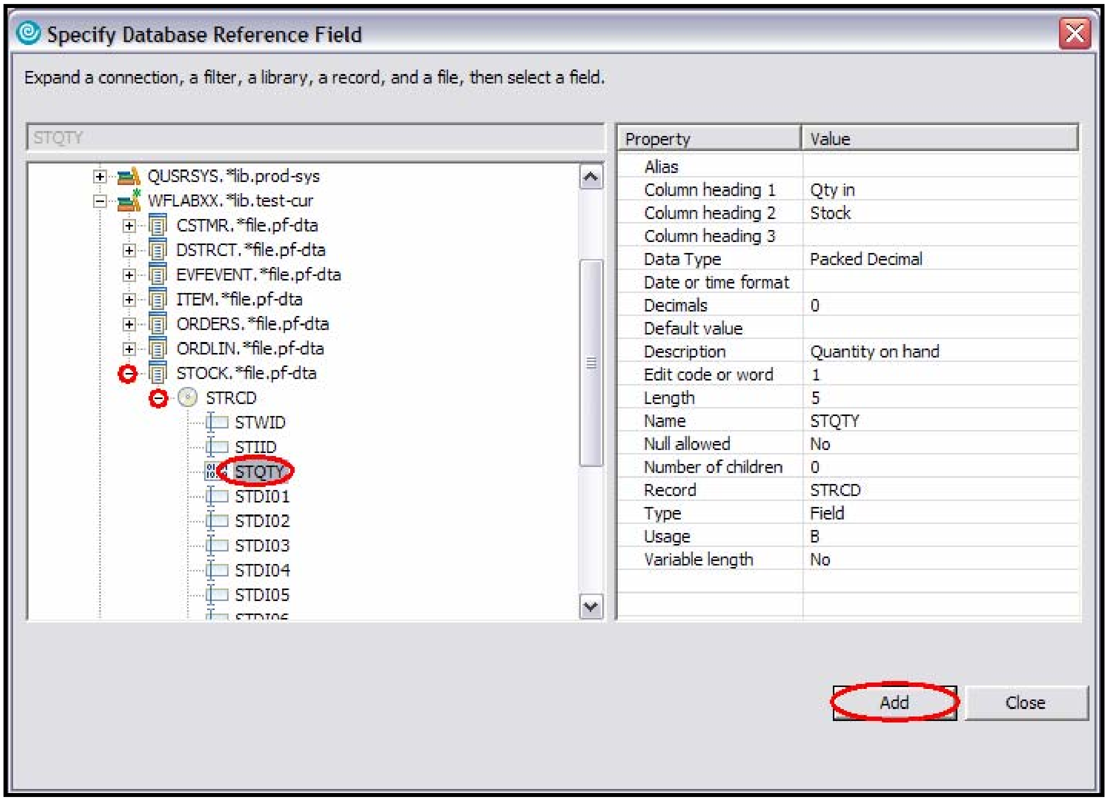
IID

Property	Value
Alias	
Column heading 1	Item
Column heading 2	Filler
Column heading 3	
Data Type	Character
Date or time format	
Decimals	0
Default value	
Description	Item Data
Edit code or word	
Length	50
Name	IDATA
Null allowed	No
Number of children	0
Record	ITRCD
Type	Field
Usage	B
Variable length	No

All fields added successfully.

Figure 327. All fields have added successfully.

14. You also need one field from the **STOCK** file (see Figure 328).



Specify Database Reference Field

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

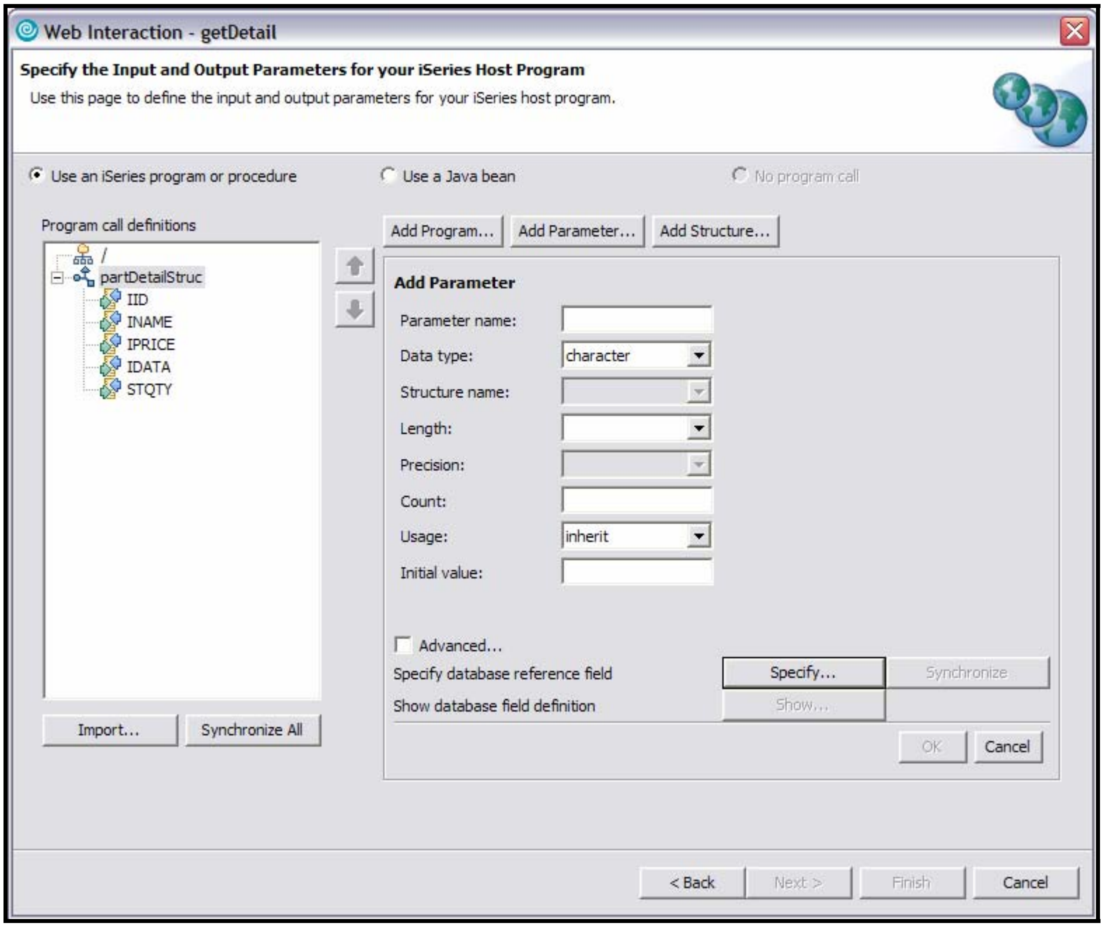
STQTY

```
QUSRSYS.*lib.prod-sys
WFLABXX.*lib.test-cur
  CSTMR.*file.pf-dta
  DSTRCT.*file.pf-dta
  EVFEVENT.*file.pf-dta
  ITEM.*file.pf-dta
  ORDERS.*file.pf-dta
  ORDLIN.*file.pf-dta
  STOCK.*file.pf-dta
    STRCD
      STWID
      STIID
      STQTY
      STDI01
      STDI02
      STDI03
      STDI04
      STDI05
```

Property	Value
Alias	
Column heading 1	Qty in
Column heading 2	Stock
Column heading 3	
Data Type	Packed Decimal
Date or time format	
Decimals	0
Default value	
Description	Quantity on hand
Edit code or word	1
Length	5
Name	STQTY
Null allowed	No
Number of children	0
Record	STRCD
Type	Field
Usage	B
Variable length	No

Figure 328. Add one field from the STOCK file

15. Expand the **STOCK** file and then the **STRCD** record format until you get to the field list.
16. Using the mouse, select the **STQTY** field, and click the **Add** button to add this field to our structure. A confirmation message stating **Field added successfully** is displayed.
17. Click the **Close** button to return to the Specify the Input and Output Parameters for your iSeries Host Program wizard page.
18. The wizard page now shows the completed structure in the left pane (see Figure ***329***).



Web Interaction - getDetail

Specify the Input and Output Parameters for your iSeries Host Program

Use this page to define the input and output parameters for your iSeries host program.

● Use an iSeries program or procedure ○ Use a Java bean ○ No program call

Program call definitions

- /
- partDetailStruc
  - IID
  - INAME
  - IPRICE
  - IDATA
  - STQTY

Add Parameter

Parameter name:	
Data type:	character
Structure name:	
Length:	
Precision:	
Count:	
Usage:	inherit
Initial value:	

□ Advanced...

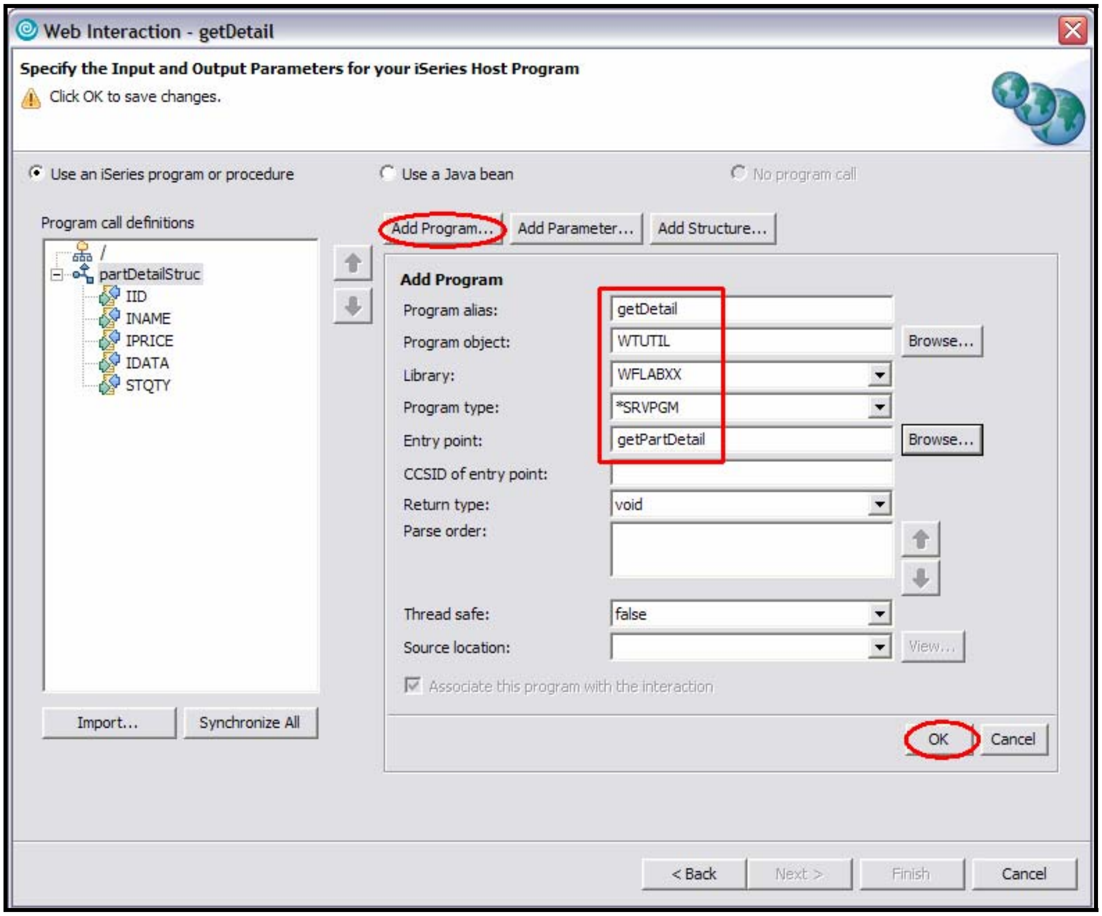
Specify database reference field

Show database field definition

Figure 329. The completed structure

Defining the procedure and parameters

Now, you can define the RPG procedure to be invoked and its parameters (see Figure ***330***).



Web Interaction - getDetail

Specify the Input and Output Parameters for your iSeries Host Program

Click OK to save changes.

● Use an iSeries program or procedure ○ Use a Java bean ○ No program call

Program call definitions

- /
  - partDetailStruc
    - IID
    - INAME
    - IPRICE
    - IDATA
    - STQTY

Add Program

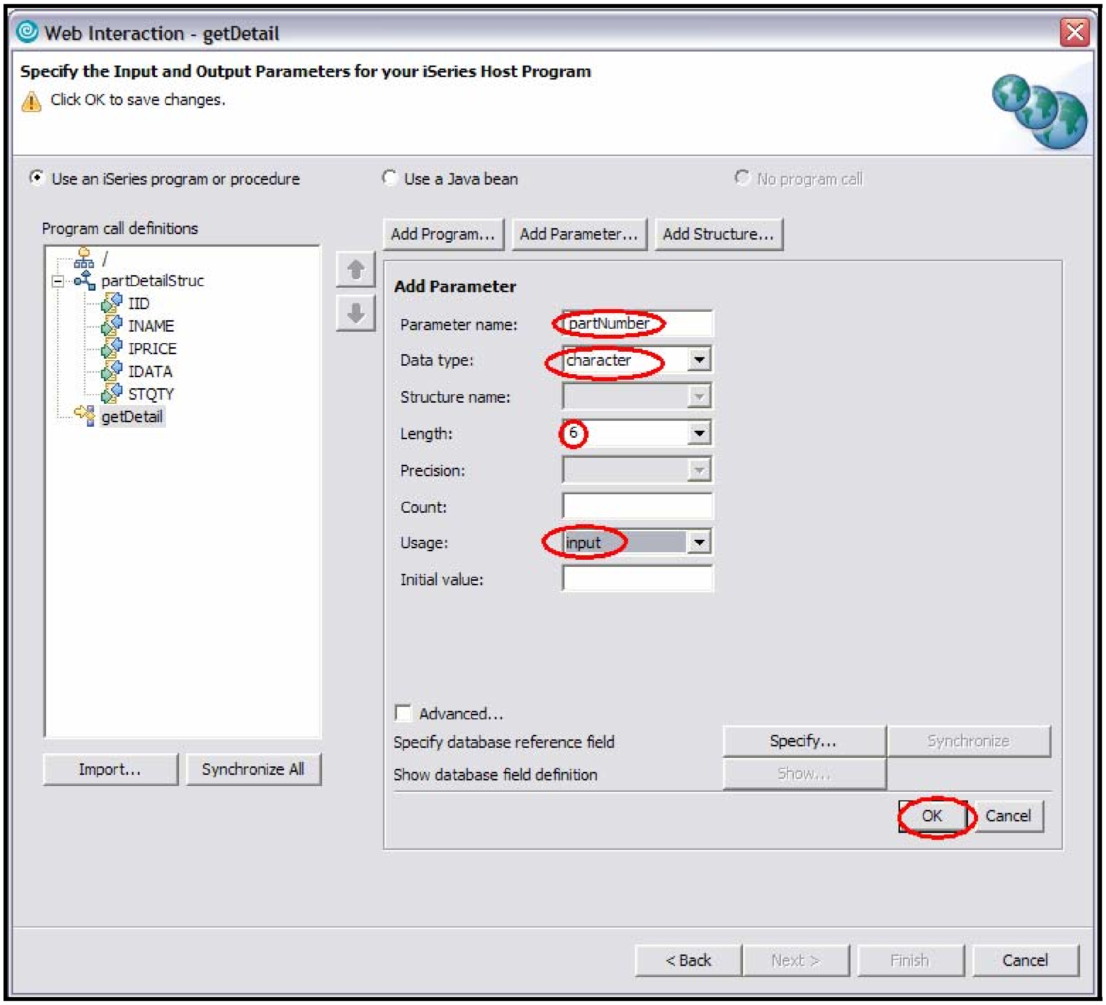
Program alias: getDetail
Program object: WTUTIL
Library: WFLABXX
Program type: *SRVPGM
Entry point: getPartDetail
CCSID of entry point:
Return type: void
Parse order:
Thread safe: false
Source location:
☑ Associate this program with the interaction

Figure 330. Defining the procedure and parameters

To add the program:

1. Click the **Add Program** button to define a program.
2. In the **Program alias** field, type `getDetail`.
3. In the **Program object** field, type `WTUTIL`.
4. In the **Library** field, type `WFLABXX`.
5. In the **Program type** list, select ***SRVPGM**.
6. In the **Entry point** field, type `getPartDetail`.
7. Click **OK** to add this program definition.

The right pane switches automatically to the **Add Parameter** view (see Figure ***331***). You now add parameters to the procedure you just defined.



Web Interaction - getDetail

Specify the Input and Output Parameters for your iSeries Host Program

Click OK to save changes.

● Use an iSeries program or procedure ○ Use a Java bean ○ No program call

Program call definitions

- /
- partDetailStruc
  - IID
  - INAME
  - IPRICE
  - IDATA
  - STQTY
- getDetail

Add Parameter

Parameter name: partNumber

Data type: character

Structure name:

Length: 6

Precision:

Count:

Usage: input

Initial value:

□ Advanced...

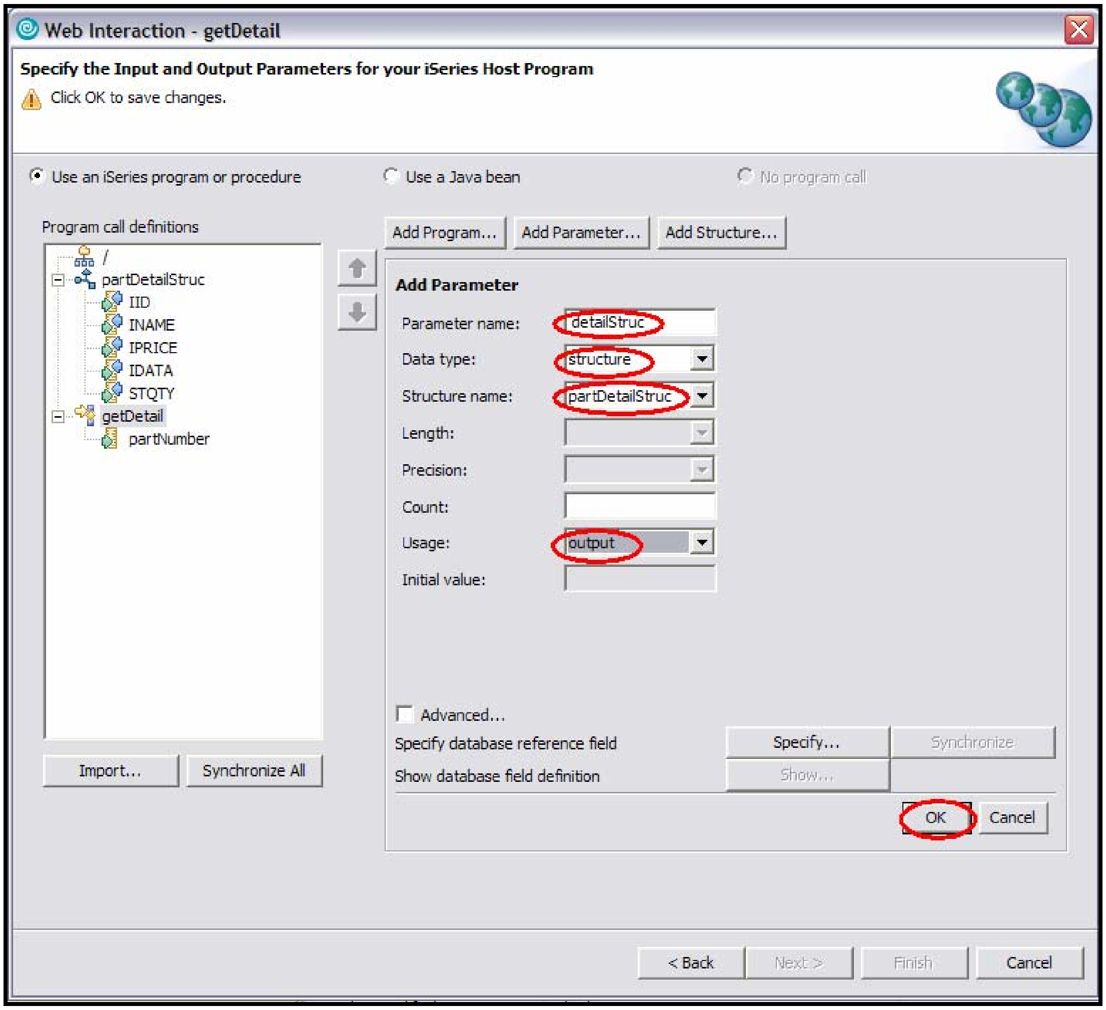
Specify database reference field

Show database field definition

Figure 331. The right pane switches automatically to the Add Parameter view.

8. In the **Parameter name** field, type `partNumber`. This is the part number passed from the hyperlink.
9. In the **Data type** list, select **character**.
10. In the **Length** field, type 6.
11. In the **Usage** list, select **input**.
12. Click **OK** to add this parameter to the program.

You now add the structure you previously defined as the second parameter to our procedure (see Figure 332).



Web Interaction - getDetail

Specify the Input and Output Parameters for your iSeries Host Program

Click OK to save changes.

● Use an iSeries program or procedure ○ Use a Java bean ○ No program call

Add Parameter

Parameter name: detailStruc

Data type: structure

Structure name: partDetailStruc

Length:

Precision:

Count:

Usage: output

Initial value:

□ Advanced...

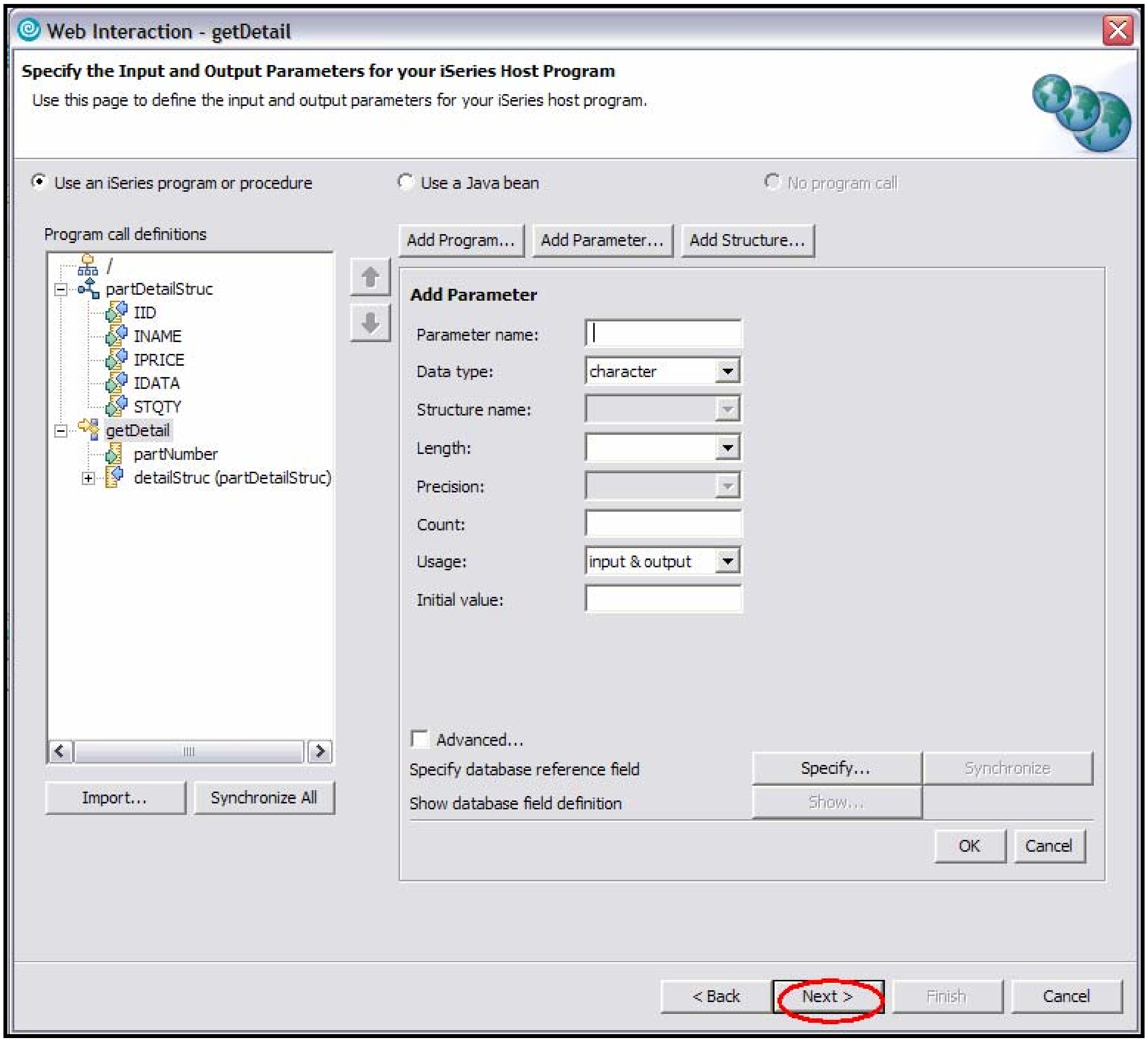
Specify database reference field

Show database field definition

Figure 332. Add parameters that you previously defined

13. In the **Parameter name** field, type `detailStruc`.
14. From the **Data type** list, select **structure**.
15. Ensure that **partDetailStruc** is selected in the **Structure name** field.
16. From the **Usage** list, select **output**.
17. Click **OK** to add this structure as the second parameter to the program call definition.

The completed program definition is displayed in the left pane in the Program call definitions section (see Figure 333).



Web Interaction - getDetail

Specify the Input and Output Parameters for your iSeries Host Program

Use this page to define the input and output parameters for your iSeries host program.

● Use an iSeries program or procedure ○ Use a Java bean ○ No program call

Program call definitions

```
/
⊟ partDetailStruc
    IID
    INAME
    IPRICE
    IDATA
    STQTY
⊟ getDetail
    partNumber
  ⊞ detailStruc (partDetailStruc)
```

Add Parameter

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

□ Advanced...

Specify database reference field

Show database field definition

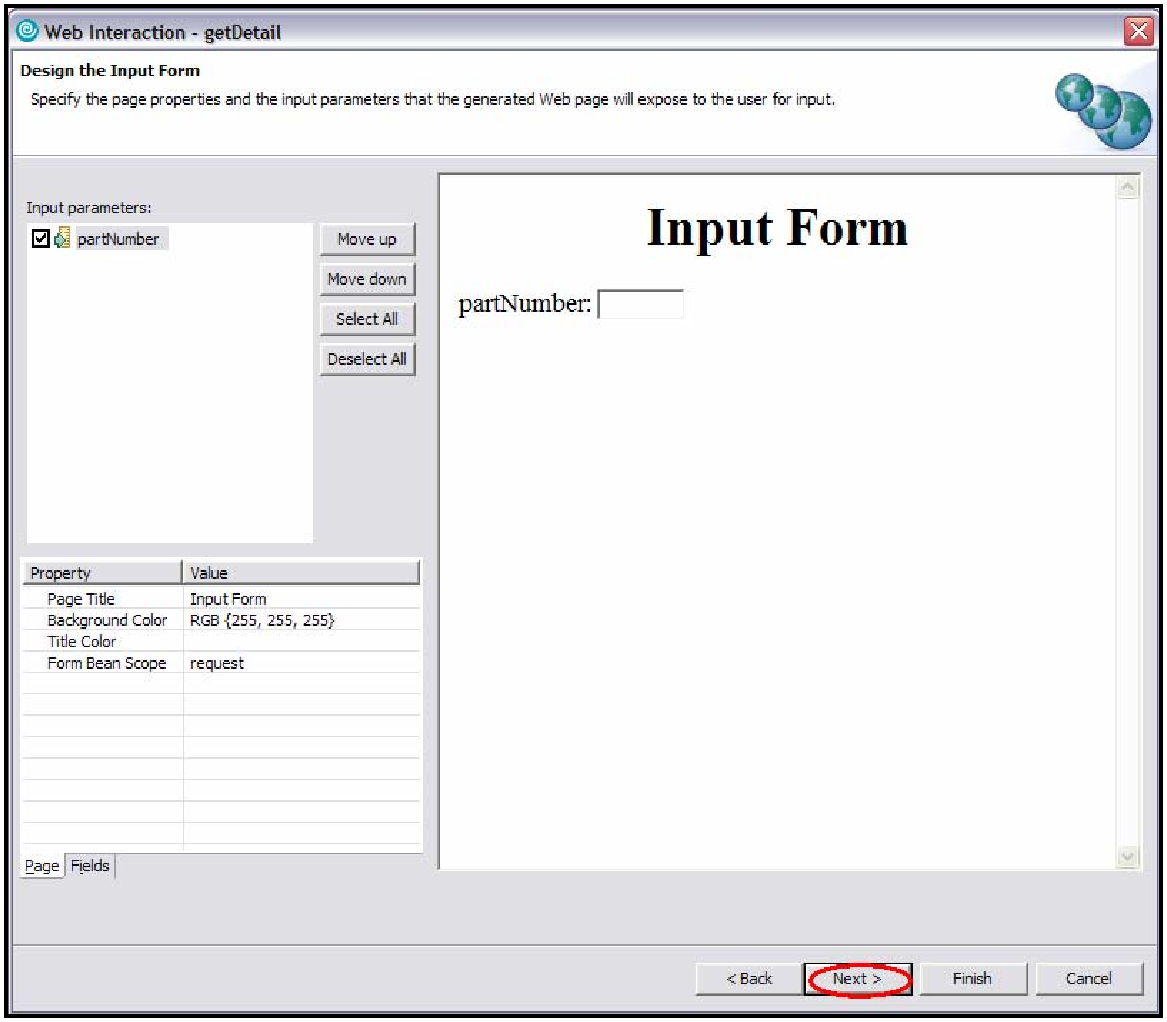
Figure 333. The completed program definition

18. Click **Next** to define the Input and Output pages for this Web interaction.

Defining the Input and Output pages

So far, you have defined the service program procedure to invoke as well as the input and output parameters. In this section, you define the content of the Web pages to be generated by the wizard.

1. Because, as mentioned earlier, you do not use the input page, except for testing. Click **Next** on the Design the Input Form page of the wizard (see
2. Figure ***334***).



Web Interaction - getDetail

Design the Input Form

Specify the page properties and the input parameters that the generated Web page will expose to the user for input.

Input parameters:

☑ partNumber

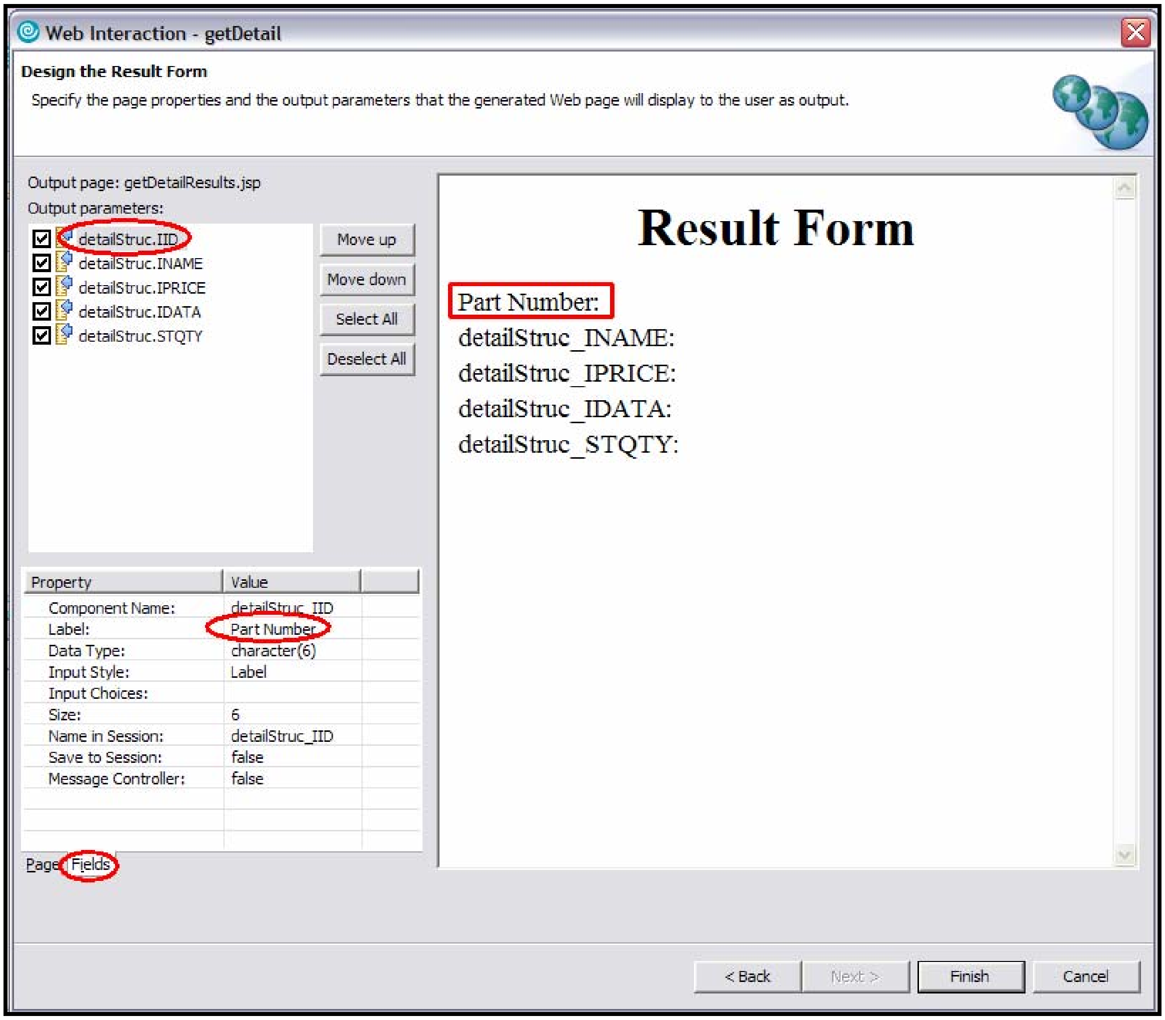
Property	Value
Page Title	Input Form
Background Color	RGB {255, 255, 255}
Title Color	
Form Bean Scope	request

Input Form

partNumber:

*Figure 334. Click **Next** on the Design the Input Form page of the wizard.*

The Design the Result Form page is displayed (see Figure 335).



Web Interaction - getDetail

Design the Result Form

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

Output page: getDetailResults.jsp

Output parameters:

- ☑ detailStruc.IID
- ☑ detailStruc.INAME
- ☑ detailStruc.IPRICE
- ☑ detailStruc.IDATA
- ☑ detailStruc.STQTY

Property	Value
Component Name:	detailStruc_IID
Label:	Part Number
Data Type:	character(6)
Input Style:	Label
Input Choices:	
Size:	6
Name in Session:	detailStruc_IID
Save to Session:	false
Message Controller:	false

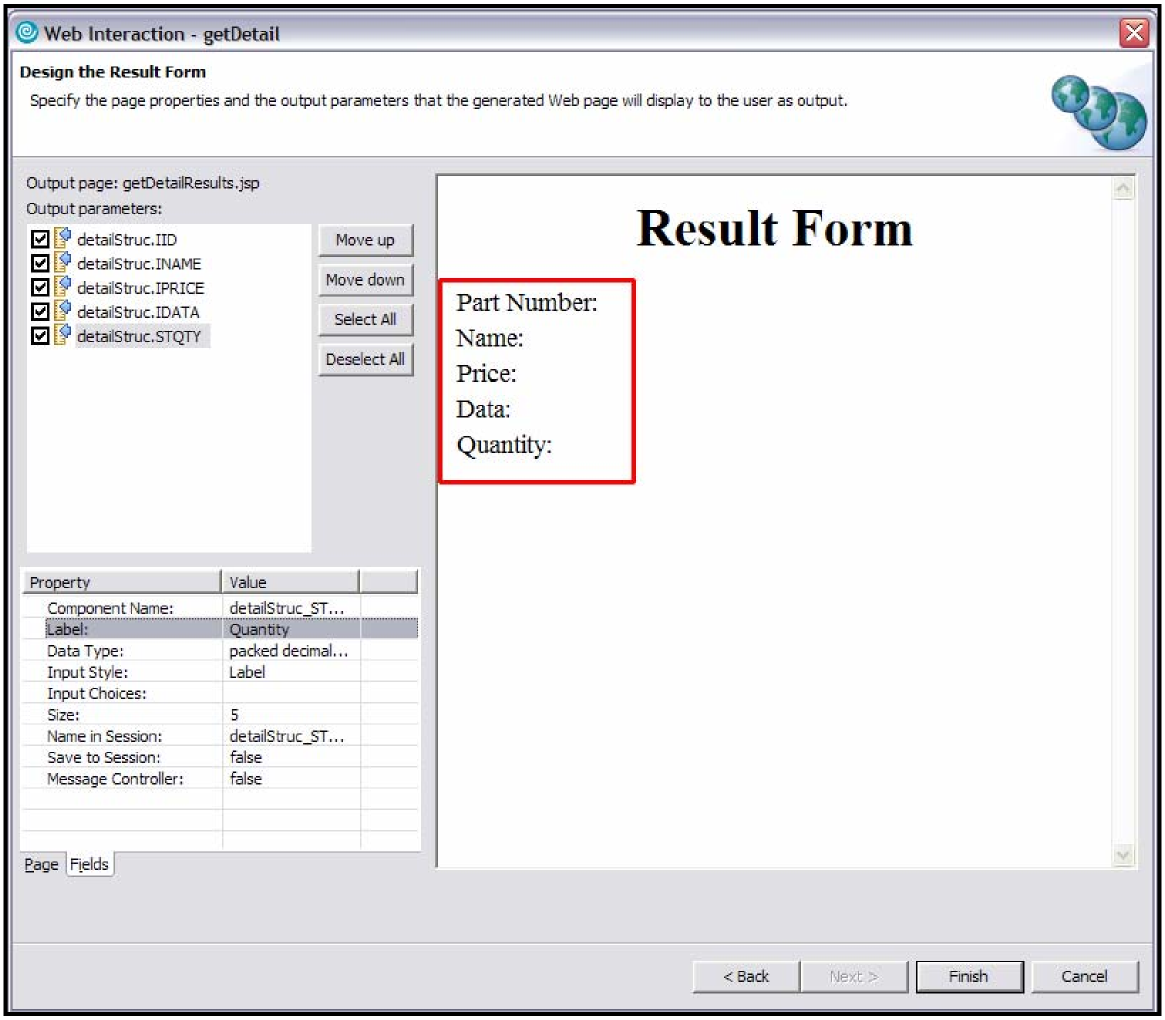
Result Form

Part Number:
detailStruc_INAME:
detailStruc_IPRICE:
detailStruc_IDATA:
detailStruc_STQTY:

Figure 335. The Design the Result Form

3. In the **Output parameters** section, select **detailStruc.IID**.
4. Click the **Fields** tab.
5. Click in the **Value** column for the **Label** property, and type **Part number**.
6. Press **Enter**.
7. Note that on the Result Form page preview pane, the label on the page has changed from **IID** to **Part number**.

8. Using the preceding steps as a guide, change the label for each of the fields to a more meaningful value. The completed form looks similar to the following, Figure 336:



Web Interaction - getDetail

Design the Result Form

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

Output page: getDetailResults.jsp

Output parameters:

☑ detailStruc.IID
☑ detailStruc.INAME
☑ detailStruc.IPRICE
☑ detailStruc.IDATA
☑ detailStruc.STQTY

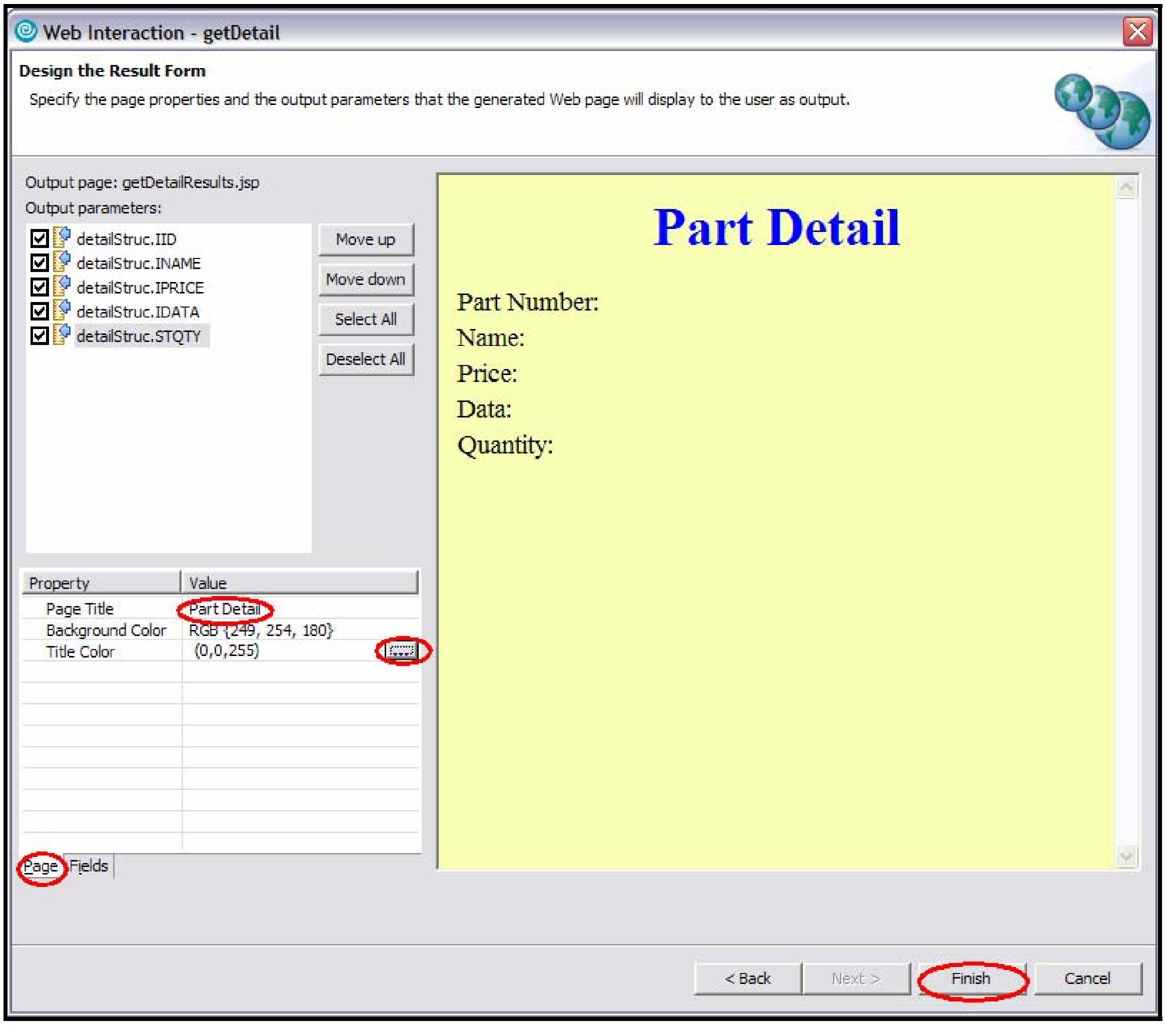
Property	Value
Component Name:	detailStruc_ST...
Label:	Quantity
Data Type:	packed decimal...
Input Style:	Label
Input Choices:	
Size:	5
Name in Session:	detailStruc_ST...
Save to Session:	false
Message Controller:	false

Result Form

Part Number:
Name:
Price:
Data:
Quantity:

Figure 336. Completed form

1. To complete the form, change the page heading and its background color (see Figure 337).



Web Interaction - getDetail

Design the Result Form

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

Output page: getDetailResults.jsp

Output parameters:

- ☑ detailStruc.IID
- ☑ detailStruc.INAME
- ☑ detailStruc.IPRICE
- ☑ detailStruc.IDATA
- ☑ detailStruc.STQTY

Property	Value
Page Title	Part Detail
Background Color	RGB {249, 254, 180}
Title Color	(0,0,255)

Part Detail

Part Number:
Name:
Price:
Data:
Quantity:

Figure 337. Change page heading and background color

Below is a more detailed list of tasks to complete this Web interaction wizard.

1. Click the **Page** tab.
2. Click in the **Value** column for the **Page Title** property and type a new title such as `Part Detail`.
3. Click the **Value** tab of the **Background Color** property and click the color selection button.
4. When the **Color** dialog is displayed, select a color of your choice for the background color and click **OK.**
5. Click the **Value** column of the **Title Color** property.
6. When the **Color** dialog is displayed, select a color of your choice for the title and click **OK**.
7. Click **Finish** to complete the Web Interaction Wizard.

Exercise 17.4: Test the Web interaction

At this point, you can test the Web interaction to verify it works before you move on to creating the hyperlink in the WebFaced application to invoke this interaction. To verify the interaction (see Figure 338):

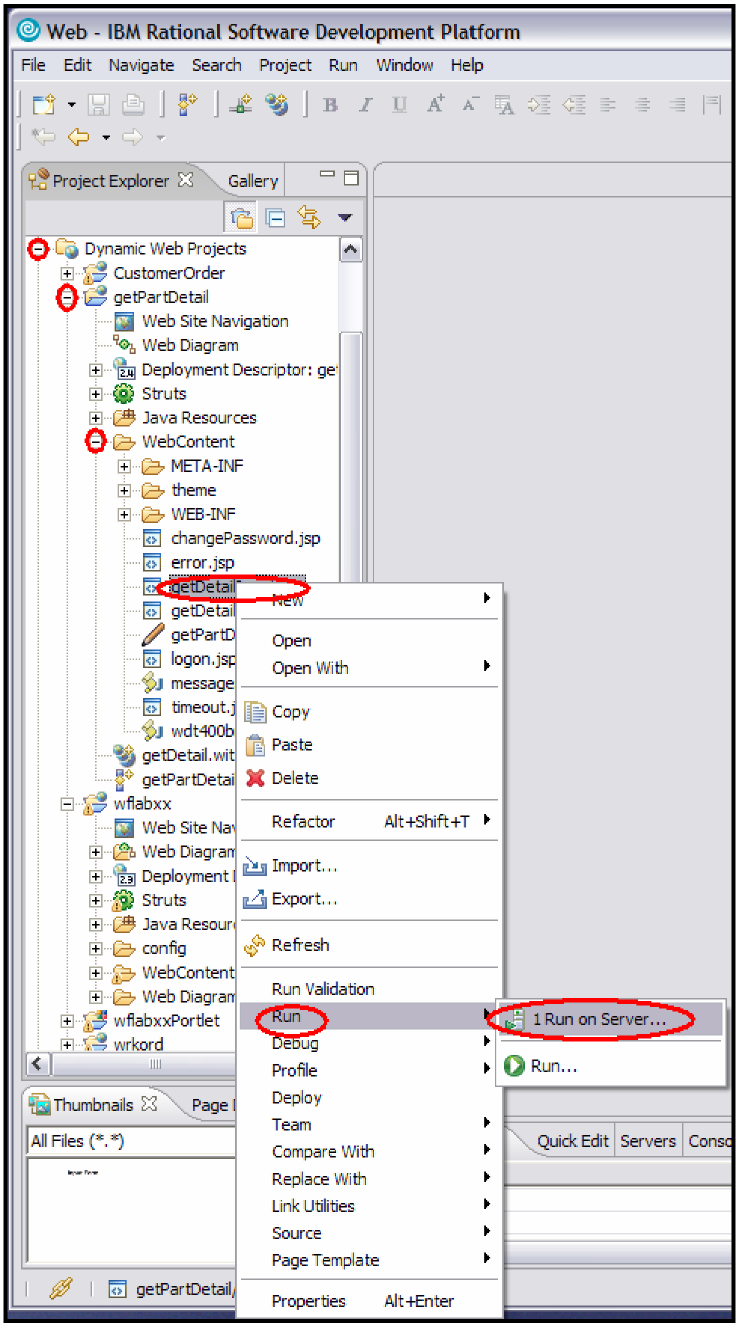
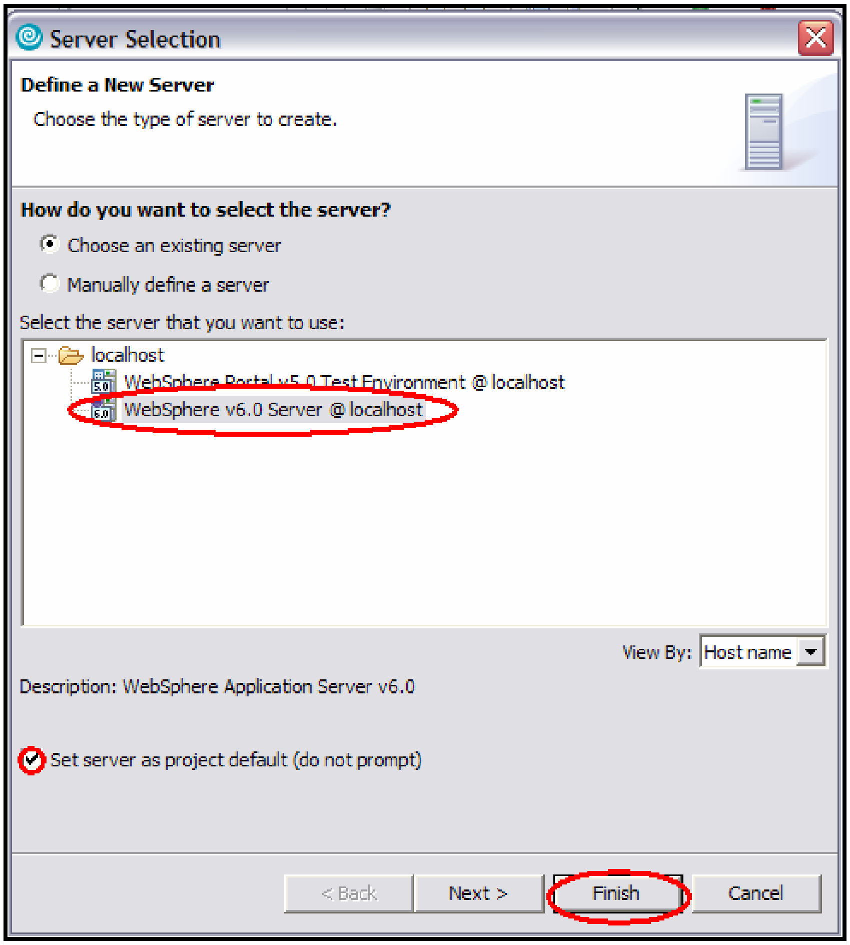


Figure 338. Test the Web interaction

From the Project Explorer view of the project, expand the project as follows:

1. Expand the **Dynamic Web Projects > getPartDetail > WebContent** folder.
2. Right-click the **getDetailInput.jsp** page and select **Run > Run on Server** from the pop-up menu.

   The Server selection dialog is displayed (see Figure **339**).


Server Selection

Define a New Server

Choose the type of server to create.

How do you want to select the server?

● Choose an existing server

○ Manually define a server

Select the server that you want to use:

- localhost
  - WebSphere Portal v5.0 Test Environment @ localhost
  - WebSphere v6.0 Server @ localhost

View By: Host name

Description: WebSphere Application Server v6.0

☑ Set server as project default (do not prompt)

Figure 339. The Server selection dialog

3. Ensure the **WebSphere v6.0 Server @ localhost** server is selected if there is more than one server.
4. Select the **Set server as project default (do not prompt)** check box.
5. Click **Finish**.

6. The input page is displayed, prompting for a part number (see Figure 340).

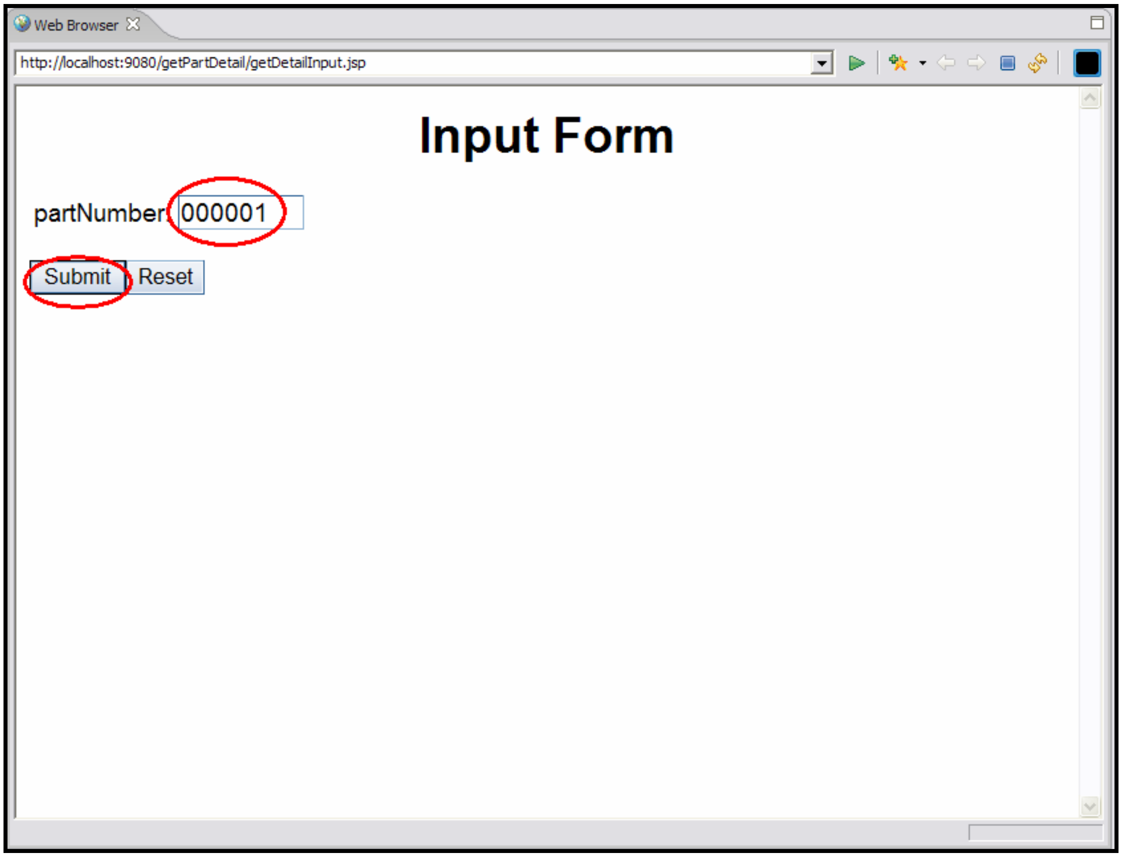
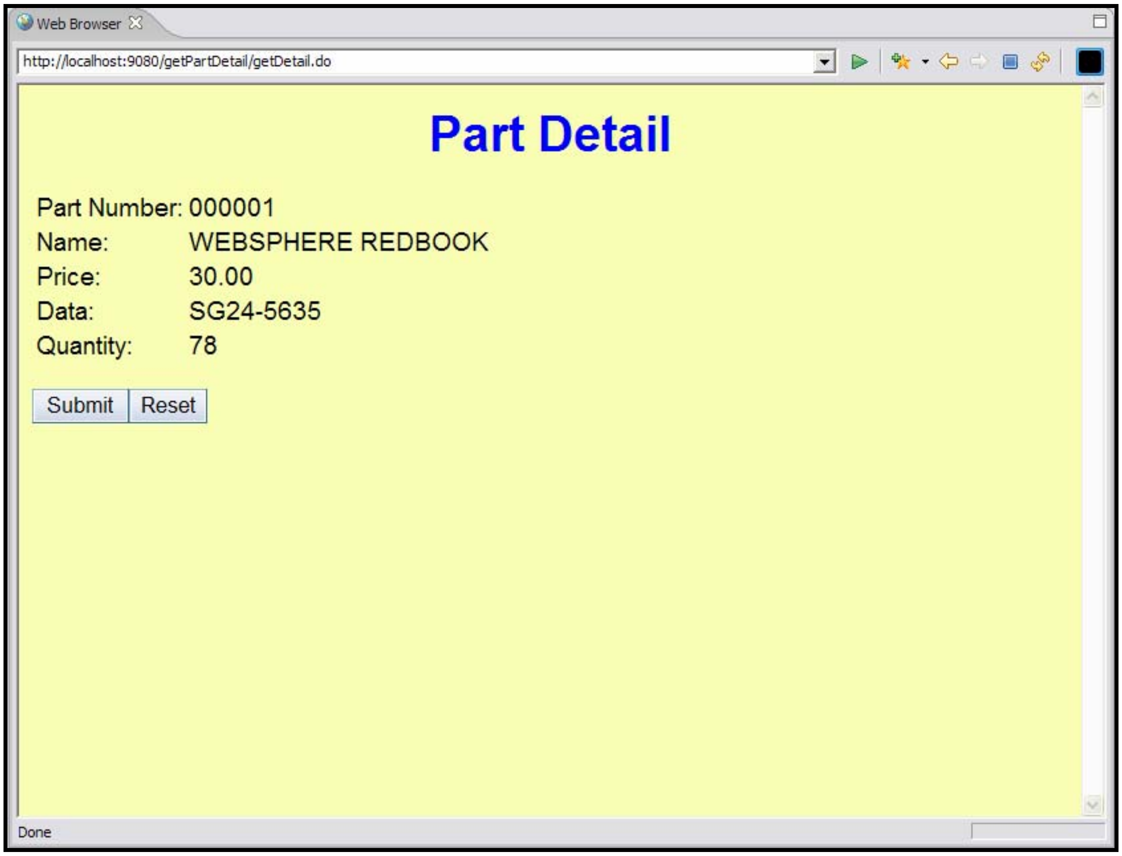


Figure 340. Input page form

7. In the **partNumber** field, type a value of 000001.
8. Click the **Submit** button.
9. The output page is displayed showing the part detail (see Figure ***341***):



Part Detail

Part Number: 000001
Name: WEBSPHERE REDBOOK
Price: 30.00
Data: SG24-5635
Quantity: 78

Figure 341. Output page showing Part Detail

Note the Web Interaction URL

Before you leave this section, you need to take note of the URL to invoke this Web Interaction. This URL is used later in the JavaScript code that you create.

Each Web Interaction is stored in the project with the name it was given when created, with a **.wit** extension.

To review the Web Interaction URL:

1. Expand the project and locate the Web Interaction file, **getDetail.wit** (see Figure 342).

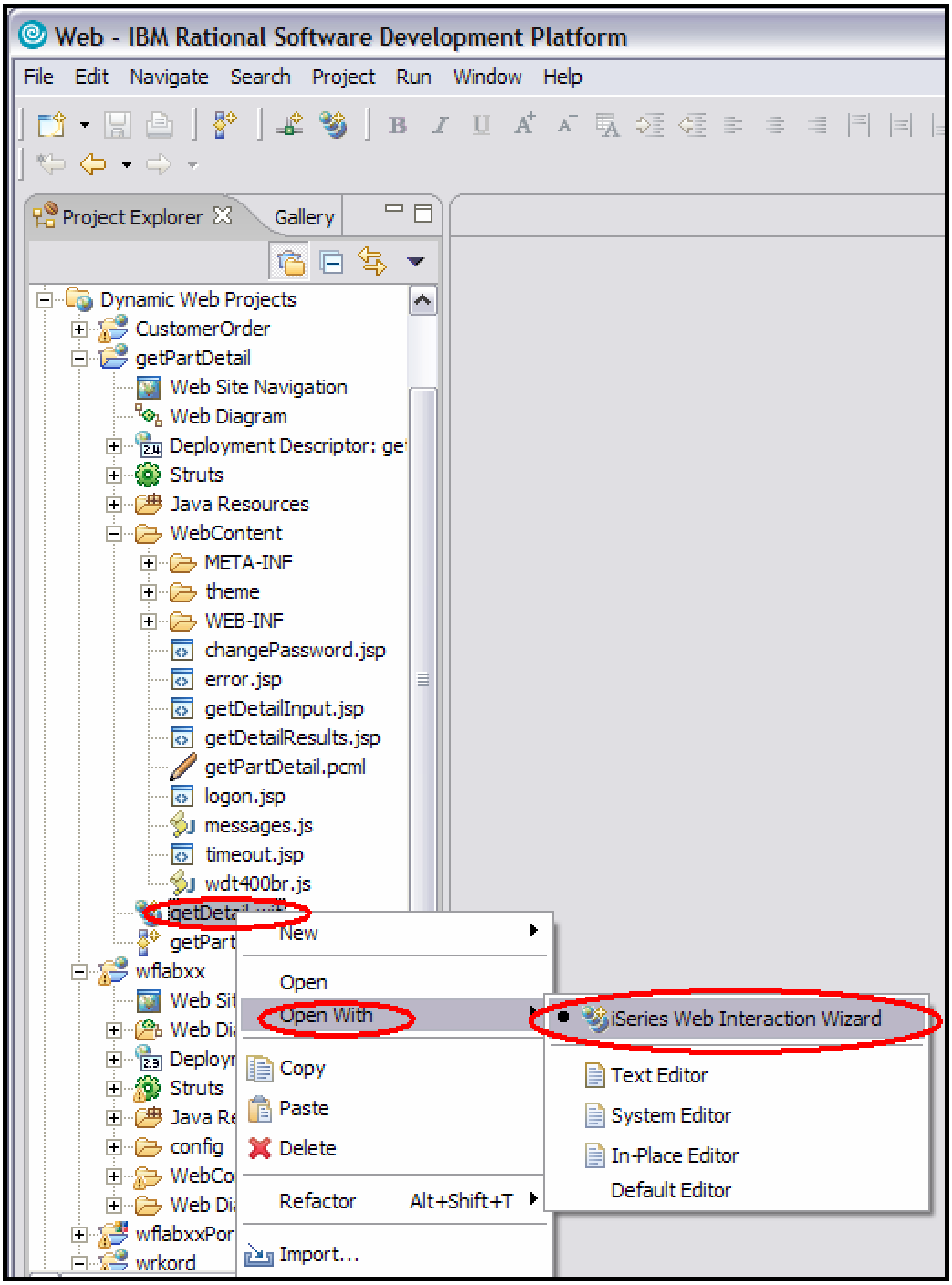
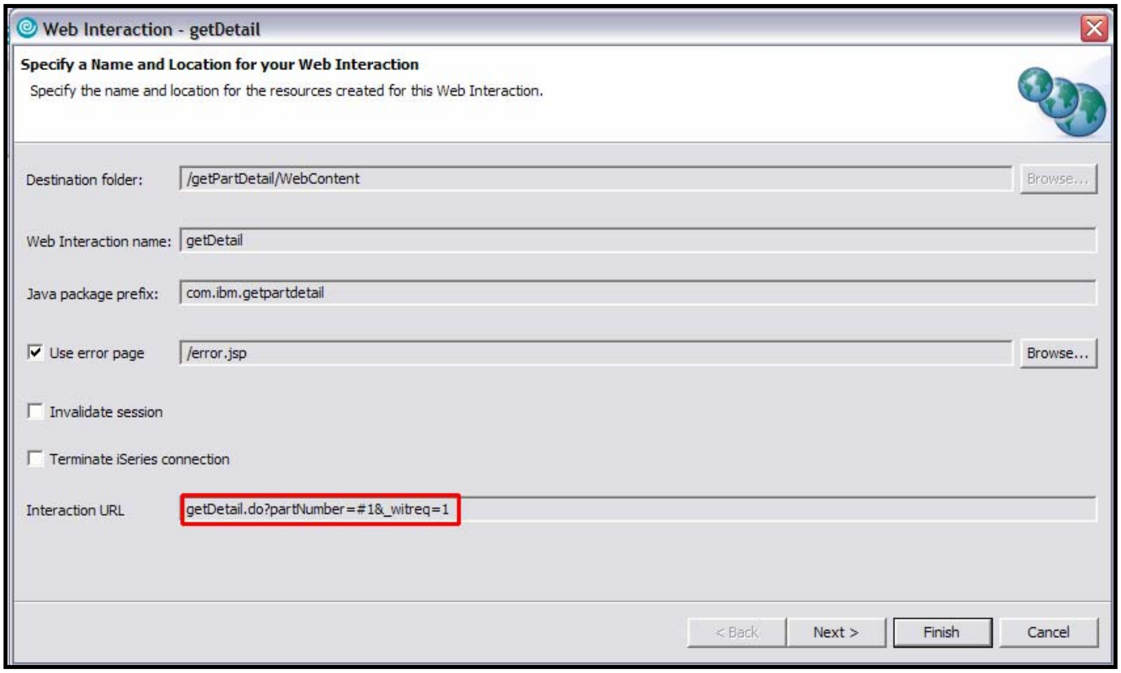


Figure 342. Expand the project and locate the Web interaction file, getDetail.wit.

2. Right-click the Interaction file and select **Open With > Web Interaction Wizard**.
3. The Web Interaction Wizard is displayed (see Figure 343).



Web Interaction - getDetail

Specify a Name and Location for your Web Interaction

Specify the name and location for the resources created for this Web Interaction.

Destination folder: /getPartDetail/WebContent

Web Interaction name: getDetail

Java package prefix: com.ibm.getpartdetail

☑ Use error page /error.jsp

□ Invalidate session

□ Terminate iSeries connection

Interaction URL getDetail.do?partNumber=#1&_witreq=1

Figure 343. The Web Interaction wizard

4. Note the value of the **Interaction URL** field. In our example, the value is **getDetail.do?partNumber=#1&_witreq=1**.
5. The **getDetail.do** portion is the name of the generated action.
6. The **partNumber=#1** portion describes the part number parameter.
7. The **_witreq=1** portion is required when invoking an Interaction from another Web page.

   When invoking an Interaction from another project, you must precede it with the context root of the Interaction. In our case, the context root is **/getPartDetail**.

   Therefore, the URL required to invoke this Web Interaction from another Web project is:

   /getPartDetail/getDetail.do?_witreq=1&partNumber=#3

   In a URL, the question mark(**?**) indicates the beginning of the parameters. The ampersand (**&**) is used to separate parameters. **#3** is replaced with the actual value of the part number at run-time. The order of the parameters is not important.

   Make note of this URL as you will use it later in this chapter.

8. Click **Cancel** to close the **Web Interaction** dialog window.

Exercise 17.5: Creating a JavaScript hyperlink using the Web Settings

Now you return to the WebFacing project and use the Web Settings feature to create the hyperlink to invoke the Web Interaction in the Web Tools project you just created. You use the Web Settings to create a hyperlink on the Part Description output field of the Select Part window.

To create the hyperlink (see
Figure ***344***):

1. In the **WebFacing Projects** view, expand the **wflabxx** project and its **DDS** folder.
2. Right-click the **SLTPARTD** source member.
3. Click **Open With > CODE Designer** on the pop-up menu.

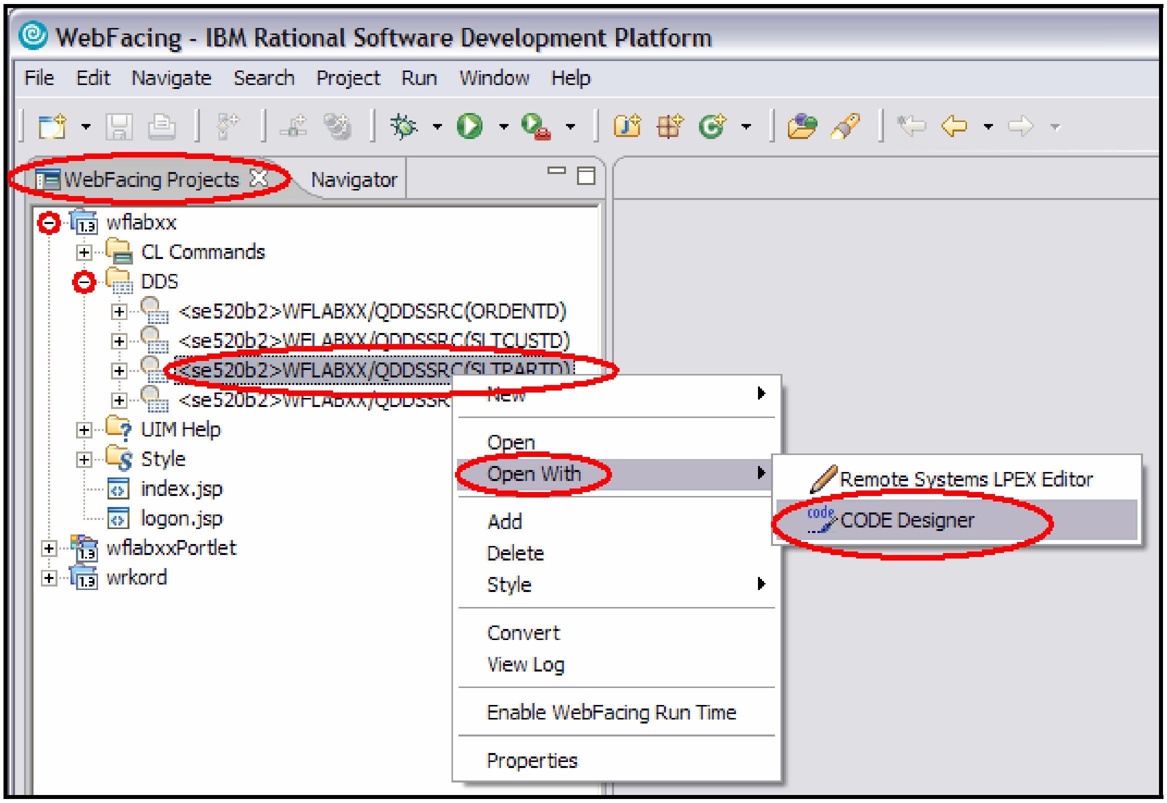
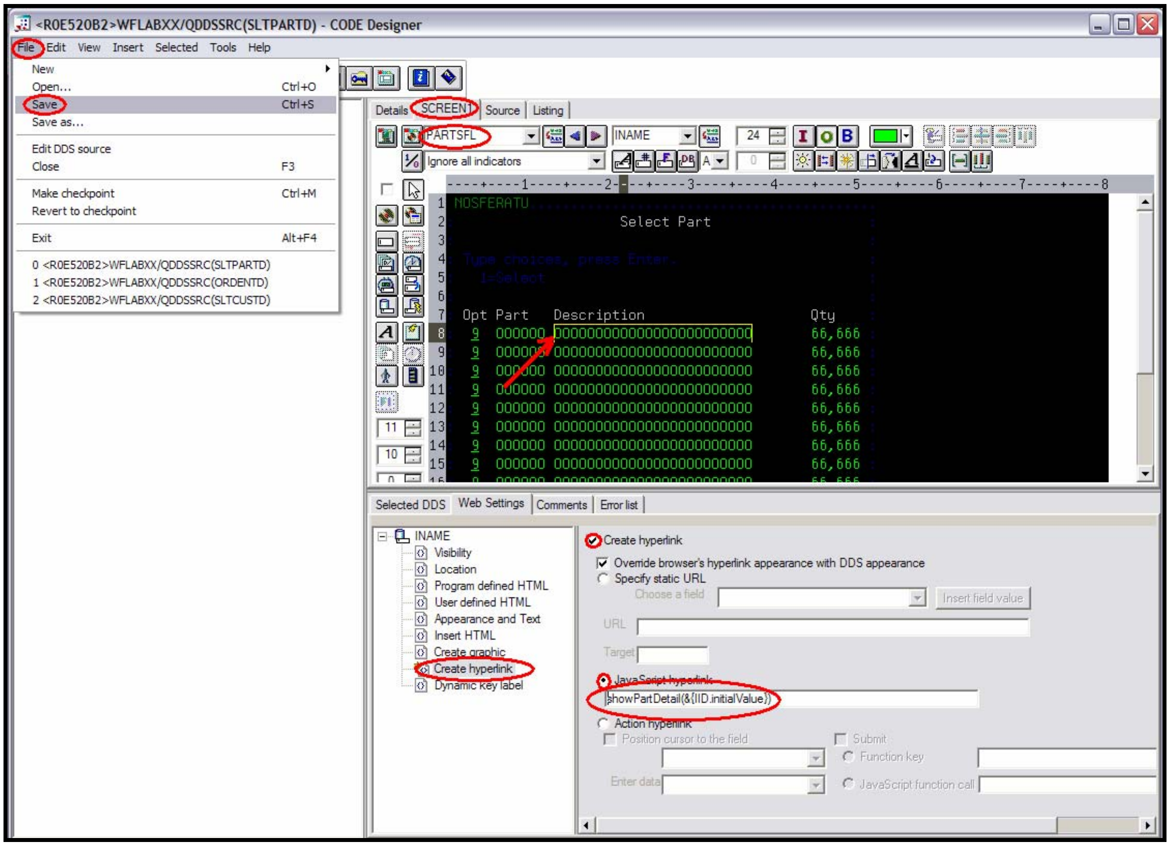

WebFacing - IBM Rational Software Development Platform
File Edit Navigate Search Project Run Window Help
WebFacing Projects
Navigator
wflabxx
CL Commands
DDS
<se520b2>WFLABXX/QDDSSRC(ORDENTD)
<se520b2>WFLABXX/QDDSSRC(SLTCUSTD)
<se520b2>WFLABXX/QDDSSRC(SLTPARTD)
<se520b2>WFLABXX/QDDSSR
UIM Help
Style
index.jsp
logon.jsp
wflabxxPortlet
wrkord
New
Open
Open With
Add
Delete
Style
Convert
View Log
Enable WebFacing Run Time
Properties
Remote Systems LPEX Editor
CODE Designer

Figure 344. Creating a JavaScript hyperlink using the Web Settings

The DDS member is loaded and CODE Designer opens (see Figure 345).



```
 1 NOSFERATU
 2                        Select Part
 3
 4  Type choices, press Enter.
 5    1=Select
 6
 7  Opt Part   Description                        Qty
 8   9  000000 000000000000000000000000000      66,666
 9   9  000000 000000000000000000000000000      66,666
10   9  000000 000000000000000000000000000      66,666
11   9  000000 000000000000000000000000000      66,666
12   9  000000 000000000000000000000000000      66,666
13   9  000000 000000000000000000000000000      66,666
14   9  000000 000000000000000000000000000      66,666
15   9  000000 000000000000000000000000000      66,666
```

☑ Create hyperlink
☑ Override browser's hyperlink appearance with DDS appearance
○ Specify static URL
● JavaScript hyperlink
showPartDetail(&{IID.initialValue})

Figure 345. The DDS member is loaded and CODE Designer opens.

1. Click the **SCREEN1** tab.
2. Select **PARTSFL** from the dropdown list.
3. Click the first row of the **Description** field and select the **Web Settings** tab.
4. In the left pane, select the **Create hyperlink** option.
5. In the right pane, select the **Create hyperlink** check box.
6. In the right pane, select the **JavaScript hyperlink** radio button.
7. In the JavaScript hyperlink data entry field, type the following text:
   `showPartDetail(&{IID.initialValue})`
8. Save the PARTSFL DDS changes by selecting **File > Save**.
9. Close the CODE editor.

With this Web Setting, a hyperlink is created around the part Description output field in the part subfile. Clicking this hyperlink invokes the JavaScript function showPartDetail. The value passed to this JavaScript function is the value of the part number field (**IID**). Note that the values in this field are case sensitive and must be typed as shown.

Now, you must create the JavaScript function in the WebFaced project. This JavaScript function, in turn, invokes the Web Interaction you created earlier.

Exercise 17.6: Creating a JavaScript function to invoke the Web Interaction

In this part of the exercise, you create the JavaScript function to invoke the Web Interaction. Each WebFacing project has a folder named **usr** in the **WebContent** folder. Any JavaScript in this folder is sent to the browser each time a WebFaced page is to be displayed. You create a JavaScript file in this folder, and type the JavaScript function into the file.

To create the JavaScript file (see
Figure ***346***):

1. Switch to the **Navigator** view in the WebFacing project.
2. Expand the **WebContent** folder until you get to the **usr** folder.

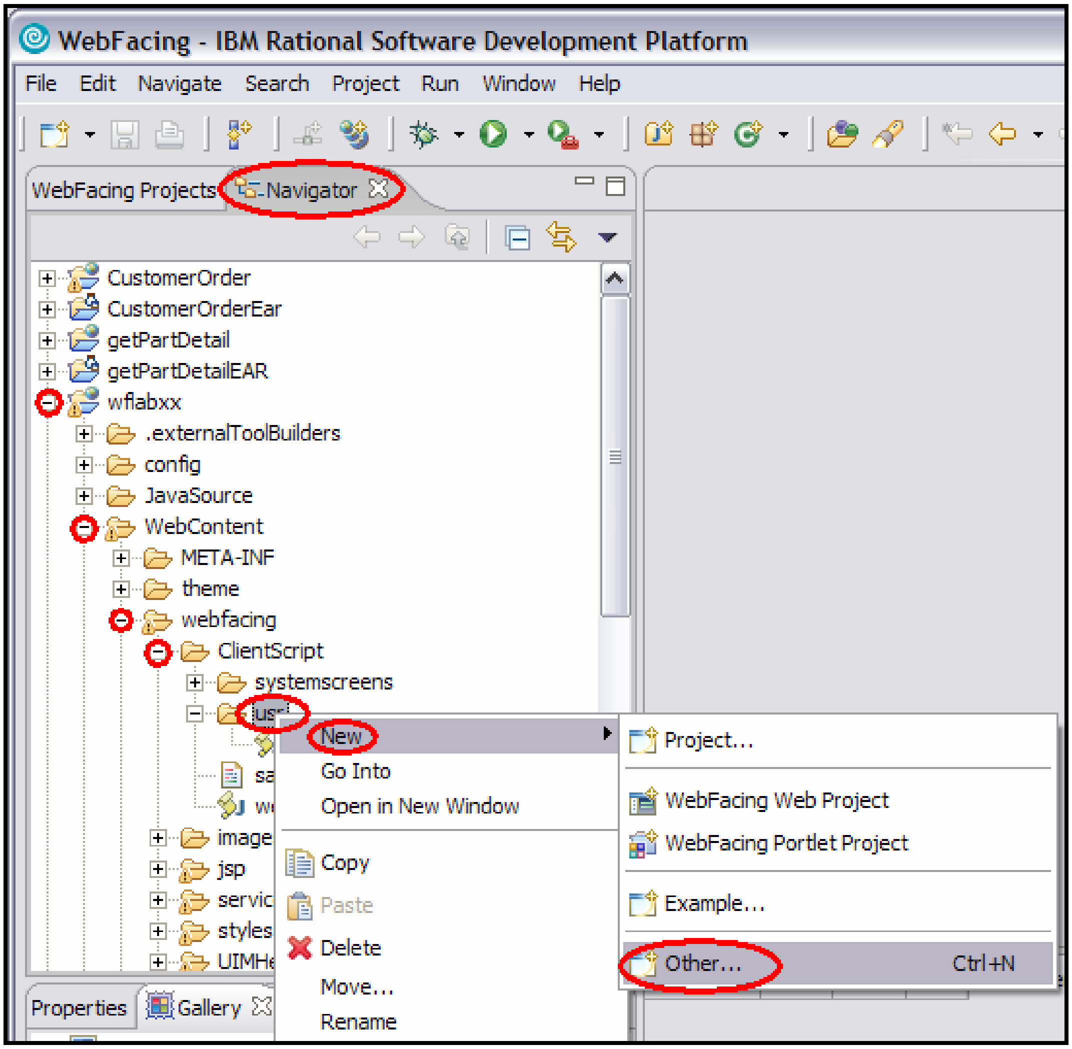


Figure 346. Creating a JavaScript function to invoke the Web Interaction

3. Right-click the **usr** folder, and from the pop-up menu, select **New > Other** (see Figure ***347***).

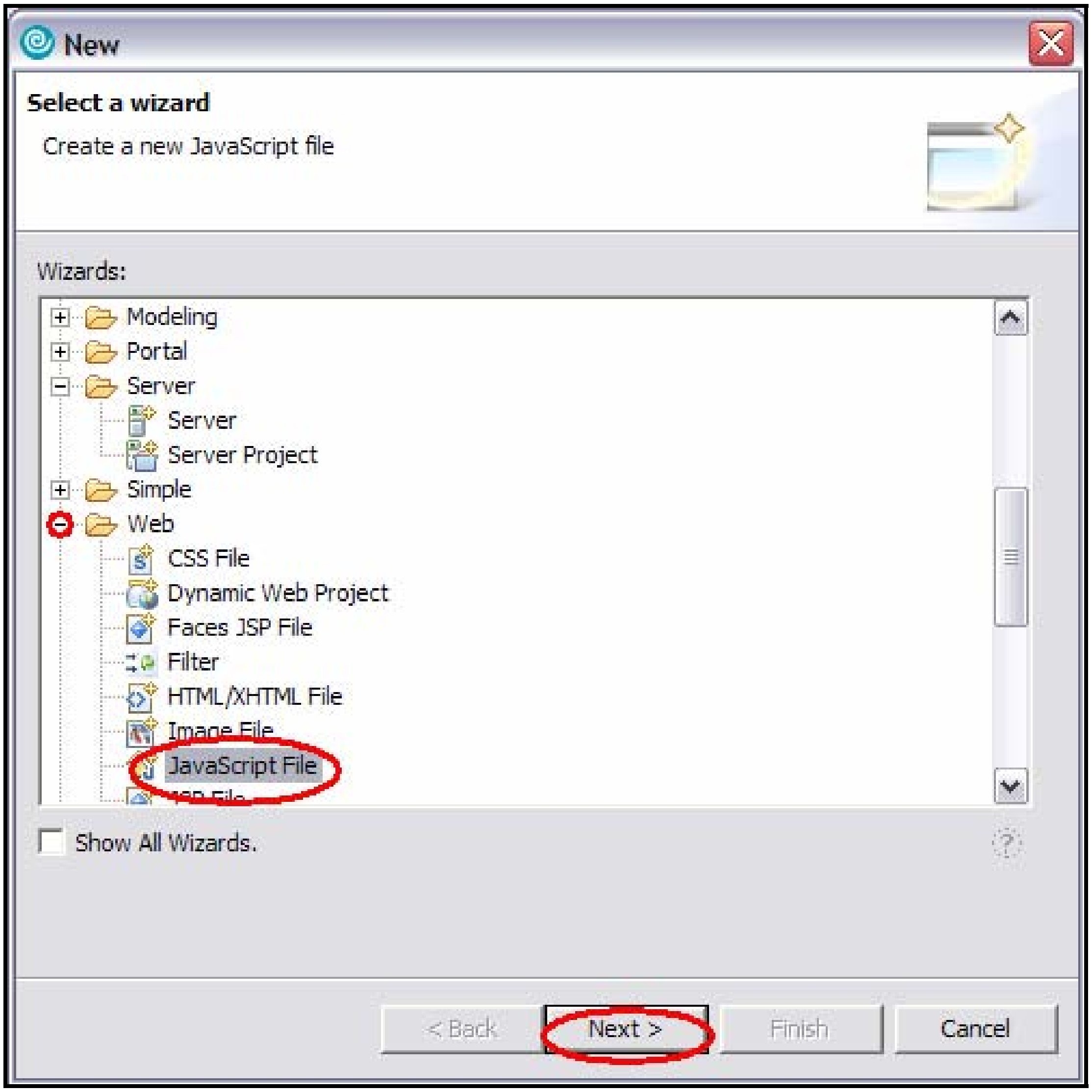
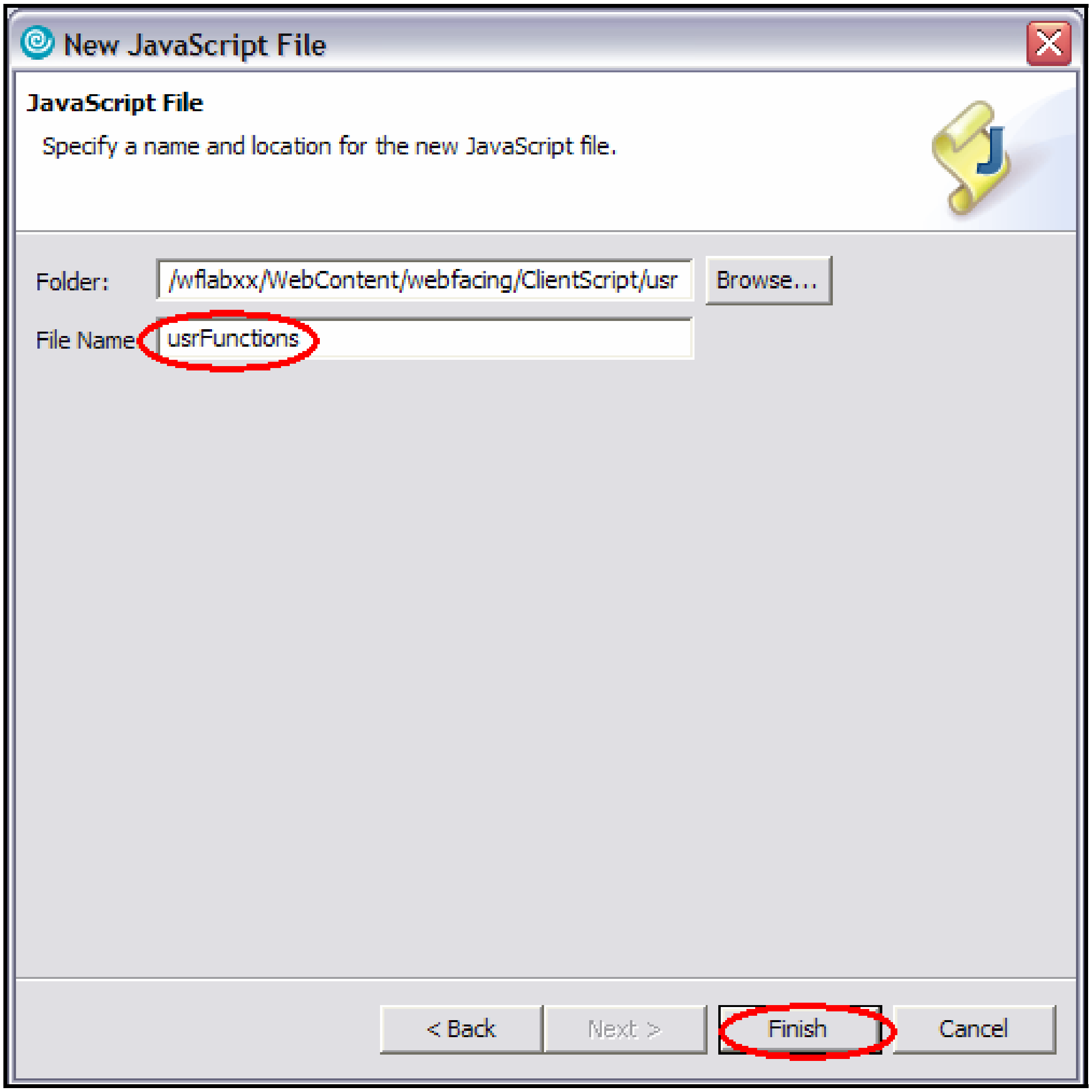


Figure 347. Create a new JavaScript file

4. On the **Select** page, select **Web** in the left pane and **JavaScript File** in the right pane.
5. Click **Next**.

6. The New JavaScript File page is displayed (see Figure ***348***).



Specify a name and location for the new JavaScript file.

Folder: /wflabxx/WebContent/webfacing/ClientScript/usr

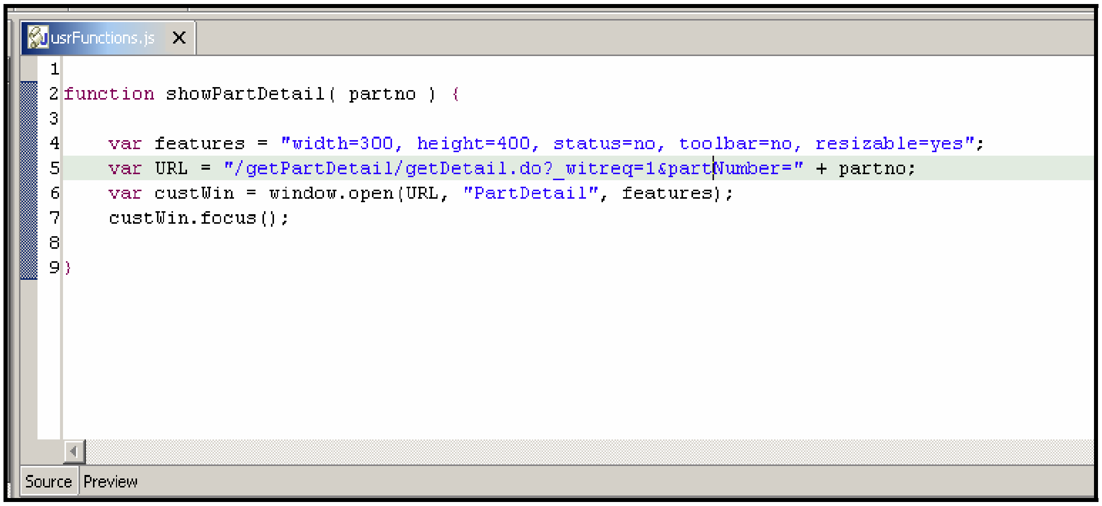
File Name: usrFunctions

Figure 348. The New JavaScript File page

7. On the **New JavaScript File** page, type a name for the new JavaScript file, such as **usrFunctions**.
   Note: The JavaScript extension of **.js** is added to the File Name after it is saved.
8. Click **Finish**.

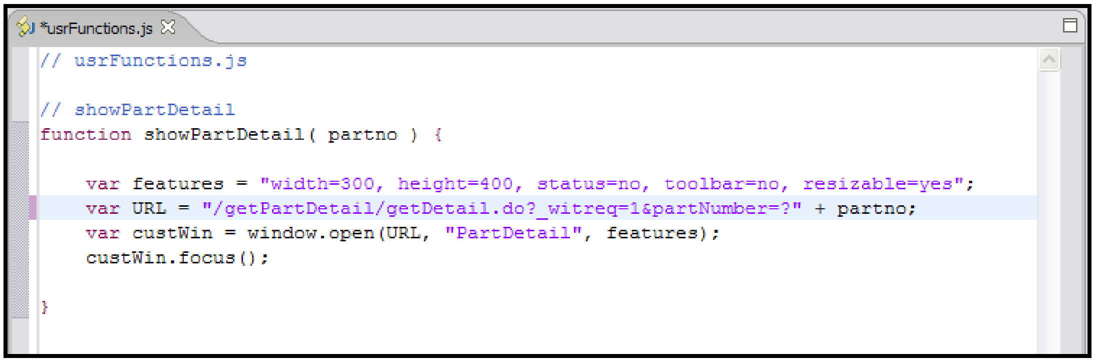
9. When the new JavaScript file opens in the editor, delete the default generated text. Create a new JavaScript function called showPartDetail. The text of this new function is (see
   Figure ***349*** and
   Figure ***350***):

```
// usrFunctions.js
// showPartDetail
function showPartDetail( partno ) {
      var features = "width=300, height=400, status=no, toolbar=no, 
resizable=yes";
      var URL = “/getPartDetail/getDetail.do?_witreq=1&partNumber=” + 
partno;
      var custWin = window.open(URL, "PartDetail", features);
      custWin.focus();
}
```



```

function showPartDetail( partno ) {

    var features = "width=300, height=400, status=no, toolbar=no, resizable=yes";
    var URL = "/getPartDetail/getDetail.do?_witreq=1&partNumber=" + partno;
    var custWin = window.open(URL, "PartDetail", features);
    custWin.focus();

}
```

Figure 349. Create a new JavaScript function called showPartDetail.



```
// usrFunctions.js

// showPartDetail
function showPartDetail( partno ) {

    var features = "width=300, height=400, status=no, toolbar=no, resizable=yes";
    var URL = "/getPartDetail/getDetail.do?_witreq=1&partNumber=?" + partno;
    var custWin = window.open(URL, "PartDetail", features);
    custWin.focus();

}
```

Figure 350. New function text

Here is an overview of this JavaScript function:

- **Line 2** declares the function and indicates it is receiving a single parameter.
- **Line 4** defines a variable that determines the features of the pop-up window including its size and the fact it has no toolbar.
- **Line 5** defines the URL that is loaded into the pop-up window. Notice that this is the URL you noted earlier in the Web interaction section of this chapter. Also, note at the end of the URL, you are appending the value of the passed in part number.
- **Line 6** creates the pop-up window, which causes it to display the results of the Web Interaction.
- **Line 7** is used to give the window focus in case it does not appear in the foreground.

10. Press **Ctrl+s** to save the new JavaScript file.

Exercise 17.7: Testing the changed application

Before you can test the new function, you must convert the WebFacing project to pick up the new Web Settings you just added

To convert the project (see
Figure ***351***):

1. Select the WebFacing Projects tab to switch to the WebFacing Project view.
2. The WebFacing Project view is displayed.

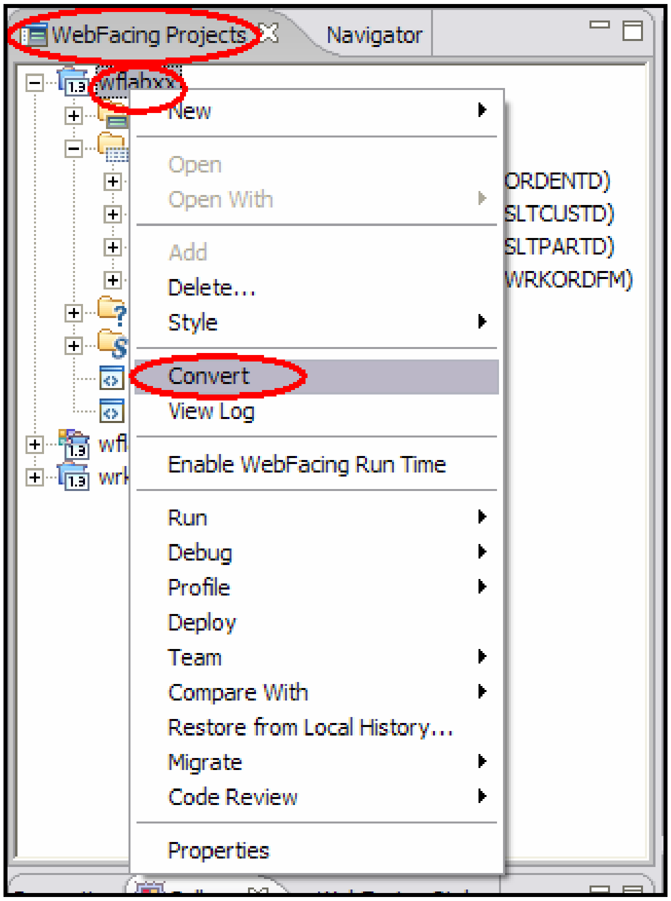


Figure 351. Testing the changed application

3. Right-click the project and select Convert from the pop-up menu.
4. Wait until the conversion is complete, and the conversion log displays.

Restart the test server

In addition, whenever changes are made to the JavaScript, you must re-start the application in the test environment server.
Note: This is not required on the production server.

To re-start the application (see Figure ***352***):

1. Switch to the **Servers** view.

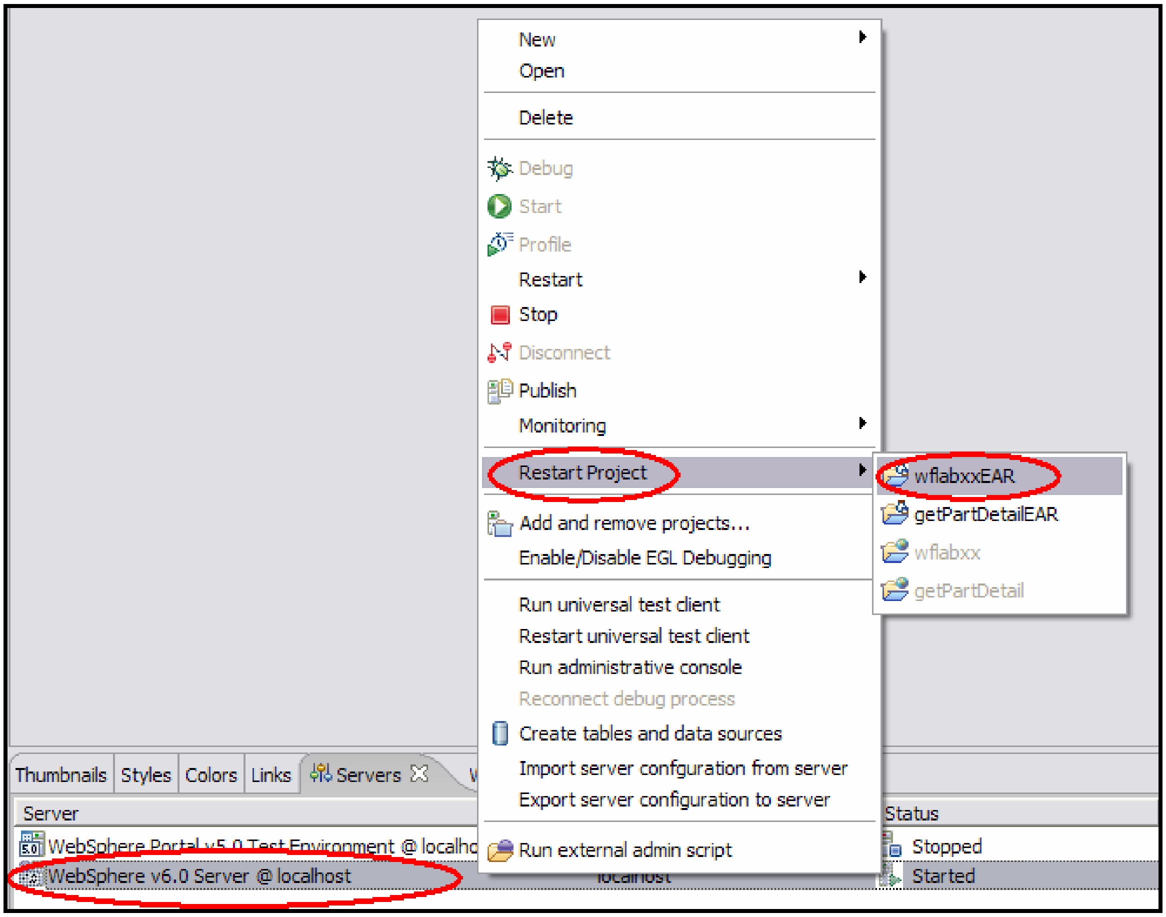
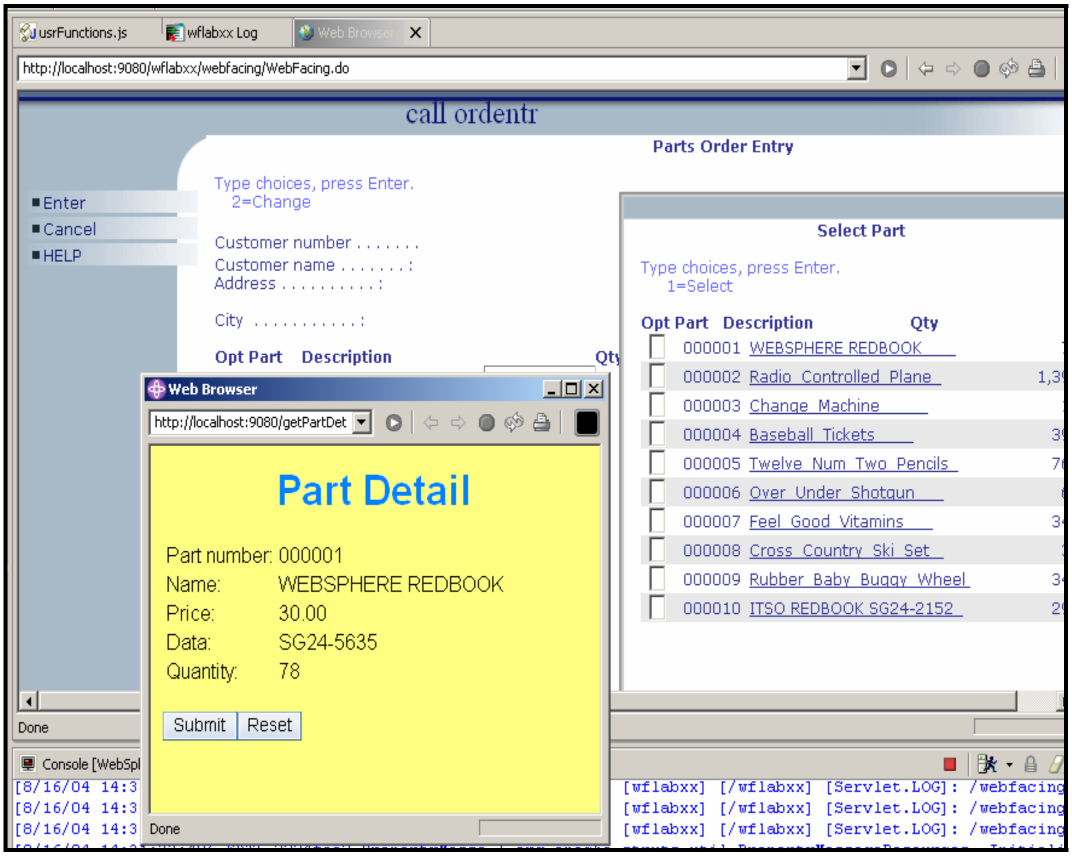

New
Open
Delete
Debug
Start
Profile
Restart
Stop
Disconnect
Publish
Monitoring
Restart Project
Add and remove projects...
Enable/Disable EGL Debugging
Run universal test client
Restart universal test client
Run administrative console
Reconnect debug process
Create tables and data sources
Import server confguration from server
Export server configuration to server
Run external admin script
wflabxxEAR
getPartDetailEAR
wflabxx
getPartDetail
Thumbnails
Styles
Colors
Links
Servers
Server
Status
WebSphere Portal v5.0 Test Environment @ localho
Stopped
WebSphere v6.0 Server @ localhost
localhost
Started

Figure 352. Switch to the Servers view.

2. Right-click the **WebSphere v6.0 Server @ localhost** test environment server and select **Restart Project**.
3. From the cascading menu, select your WebFaced project, **wflabxxEAR**.
4. Wait until the message appears in the Console view indicating your application has started.
5. In the WebFacing Projects view, right-click the **wflabxx** project and select **Run on Server**.

6. As you have done before, select a customer number to get to the Parts Order Entry screen (see Figure 353).



Parts Order Entry

Type choices, press Enter.
2=Change

Customer number . . . .
Customer name . . . . :
Address . . . . :

City . . . . :

Opt Part Description Qty

Select Part

Type choices, press Enter.
1=Select

Opt	Part	Description	Qty
□	000001	WEBSPHERE REDBOOK	
□	000002	Radio Controlled Plane	1,3
□	000003	Change Machine	
□	000004	Baseball Tickets	3
□	000005	Twelve Num Two Pencils	7
□	000006	Over Under Shotgun	
□	000007	Feel Good Vitamins	3
□	000008	Cross Country Ski Set	
□	000009	Rubber Baby Buggy Wheel	3
□	000010	ITSO REDBOOK SG24-2152	2

Part Detail

Part number: 000001
Name: WEBSPHERE REDBOOK
Price: 30.00
Data: SG24-5635
Quantity: 78

Submit Reset

```
[8/16/04 14:3 [wflabxx] [/wflabxx] [Servlet.LOG]: /webfacing
[8/16/04 14:3 [wflabxx] [/wflabxx] [Servlet.LOG]: /webfacing
[8/16/04 14:3 [wflabxx] [/wflabxx] [Servlet.LOG]: /webfacing
```

Figure 353. Select a customer number to get to Parts Order Entry screen

7. Press **F4** to open the **Select Part** window.
8. Note that the part Description output fields of the subfile are now hyperlinks.
9. Click one of the hyperlinks.
10. The part detail, as retrieved by the Web Interaction, is displayed in a pop-up window.

Note: When the pop-up window appears, it is larger than the size that was defined in the JavaScript. This is due to the behavior of the built-in Web browser. To see how the pop-up appears to an end user:

1. Start a new instance of Internet Explorer.
2. Copy the URL address from the location field of the internal browser.
3. Paste the URL into the location field of the new browser window.
4. Press **Enter** to invoke the WebFaced application.

Now, when you run the application, the pop-up window appeara in the size defined by the JavaScript function.

Recap

You have completed “Invoking a System i program from a WebFaced application.” You now have the information to understand how to:

- Use the Web Interaction wizard to invoke a System i program
- Test the Web interaction
- Add a JavaScript hyperlink using Web Settings
- Add a JavaScript function to a WebFaced application
- Restart a Web project
- Test the application by clicking on one of the hyperlinks

Adding a pop-up calendar

In this chapter, you learn how to incorporate a pop-up calendar in a WebFaced page. Pop-up calendars are very useful in a graphical interface in that they allow the user simply to select a date value and have it inserted into the target field. This usually results in fewer typing errors by the user as well as making the application easier to use.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 18.1: Reviewing the sample 5250 application
- Exercise 18.2: Importing the calendar JavaScript and style sheet
- Exercise 18.3: Adding the Web Settings to invoke the calendar pop-up
- Exercise 18.4: Convert the project with the new Web Settings
- Exercise 18.5: Test the application with the JavaScript calendar.

The exercises within each chapter must be completed in order. Start with "Exercise 18.1: Reviewing the sample 5250 application."

Length of time

This chapter takes approximately 30 minutes to complete.

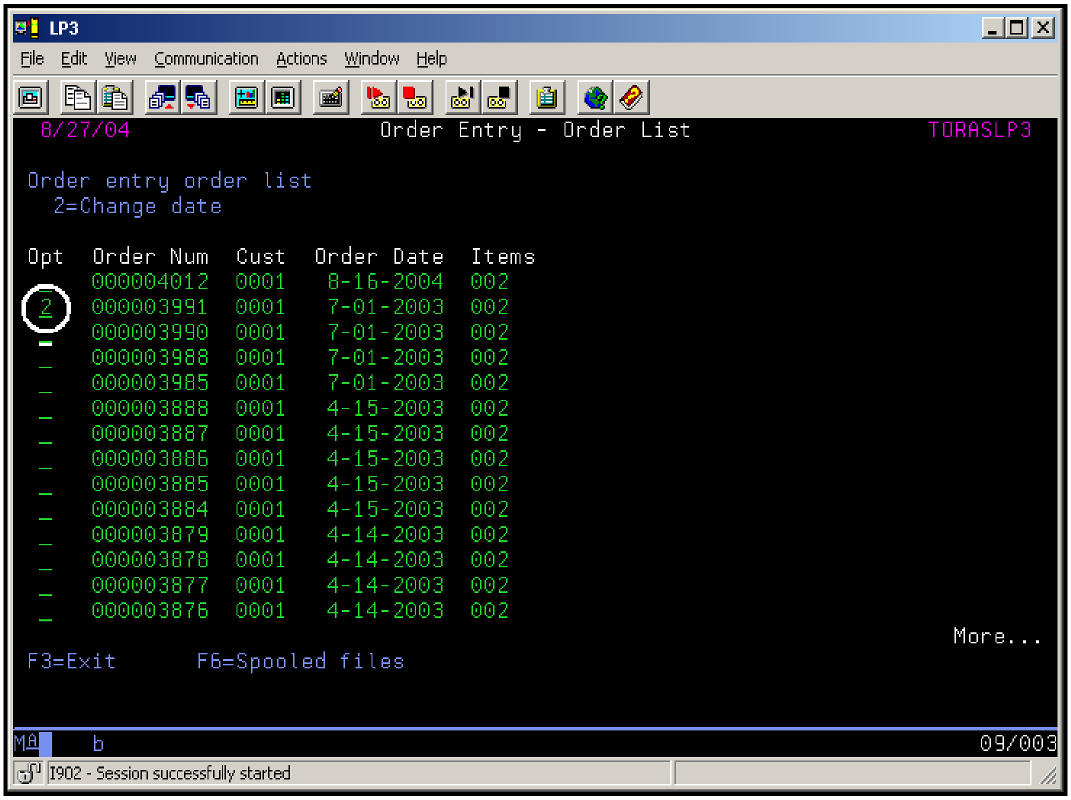
Exercise 18.1: Reviewing the 5250 sample application

You need a 5250 emulator on your workstation to start this sample application. The display for this application uses a subfile to display the orders created by the Order Entry application. Selecting option **2** on an order displays a window that allows you to change the order date for an order.

To run the 5250 application:

1. Start a 5250 emulation session.
2. In the **User ID** field, type your User ID.
3. In the **Password** field, type your password.
4. On the command line of the 5250 screen, invoke the sample application: CALL WRKORD.

   The application starts and displays the Order List screen (see Figure ***354***).

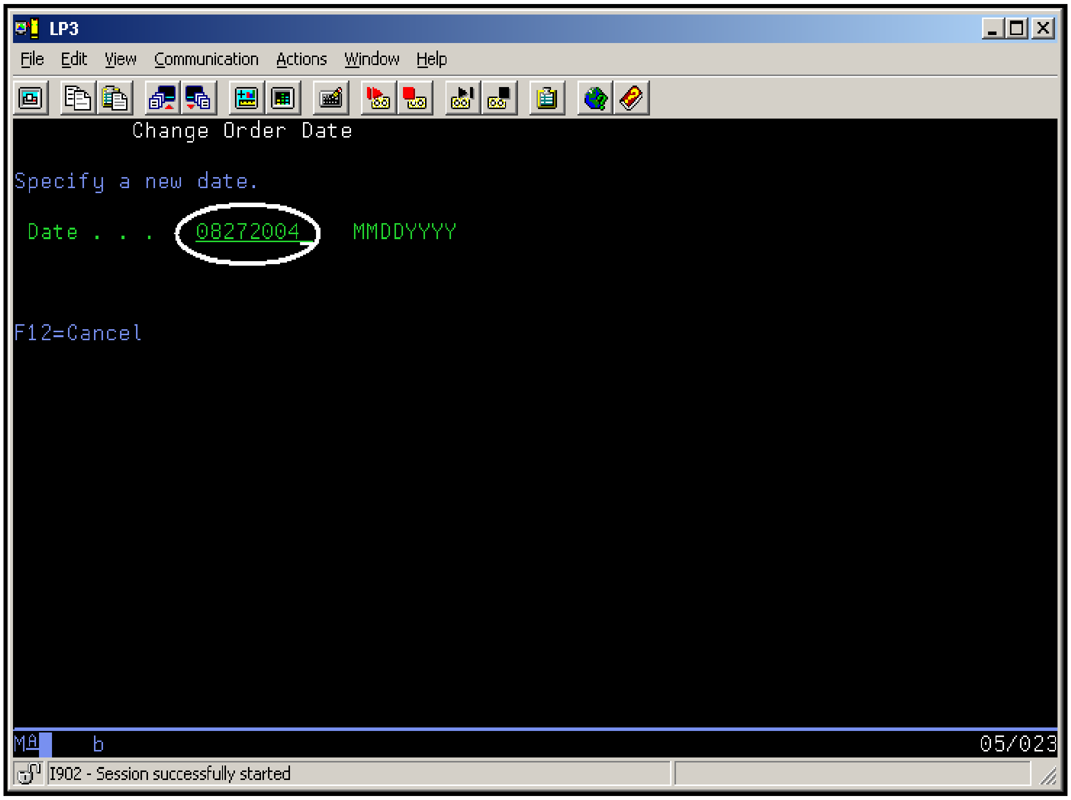


```
 8/27/04                     Order Entry - Order List                  TORASLP3

Order entry order list
  2=Change date

Opt  Order Num  Cust   Order Date  Items
     000004012  0001    8-16-2004  002
 2   000003991  0001    7-01-2003  002
 _   000003990  0001    7-01-2003  002
 _   000003988  0001    7-01-2003  002
 _   000003985  0001    7-01-2003  002
 _   000003888  0001   4-15-2003  002
 _   000003887  0001   4-15-2003  002
 _   000003886  0001   4-15-2003  002
 _   000003885  0001   4-15-2003  002
 _   000003884  0001   4-15-2003  002
 _   000003879  0001   4-14-2003  002
 _   000003878  0001   4-14-2003  002
 _   000003877  0001   4-14-2003  002
 _   000003876  0001   4-14-2003  002
                                                                       More...
F3=Exit       F6=Spooled files

MA      b                                                              09/003
I902 - Session successfully started
```

Figure 354. The Order List screen

5. You can use the page keys to scroll through the orders.
6. In the option field for one of the orders, type a 2, and press **Enter**.

7. Another screen is displayed prompting for a new date for this order (see Figure 355).



```
          Change Order Date

Specify a new date.

 Date . . .   08272004_   MMDDYYYY


F12=Cancel
```

I902 - Session successfully started

Figure 355. Another screen is displayed prompting for a new date for this order.

8. Type a new date in the **Date** field and press **Enter**.
9. The date is updated in the selected subfile record.
10. Press **F3=Exit** to end the sample application.

In the next exercise, you import some JavaScript files as well as a style sheet that displays a pop-up calendar from which the user can select a date on the Change Order Date screen on the refaced version of this sample application.

Exercise 18.2: Importing the calendar JavaScript

In this exercise, you import the JavaScript files and style sheet required to display a pop-up calendar.

There are many JavaScript calendars available on the Internet. For this exercise, you use one that was downloaded from: http://www.hotscripts.com/JavaScript. On that page, click **Scripts and Programs**, then **Calendars**, which takes you to a page with many JavaScript calendars. The one used in this exercise is **JS calendar version 9.6**.

To import the JavaScript files (see Figure 356):

1. Switch to the **Navigator** view for the **wrkord** project.

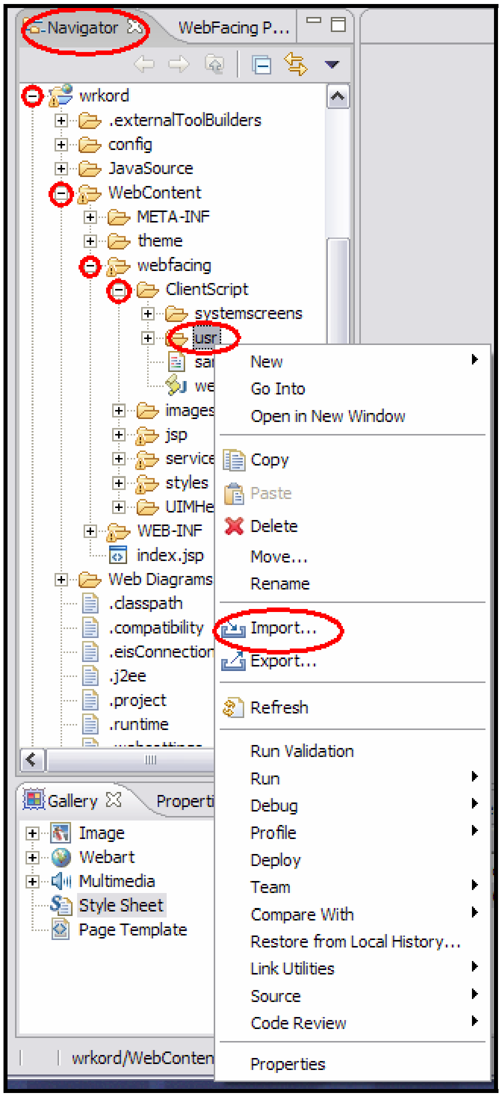
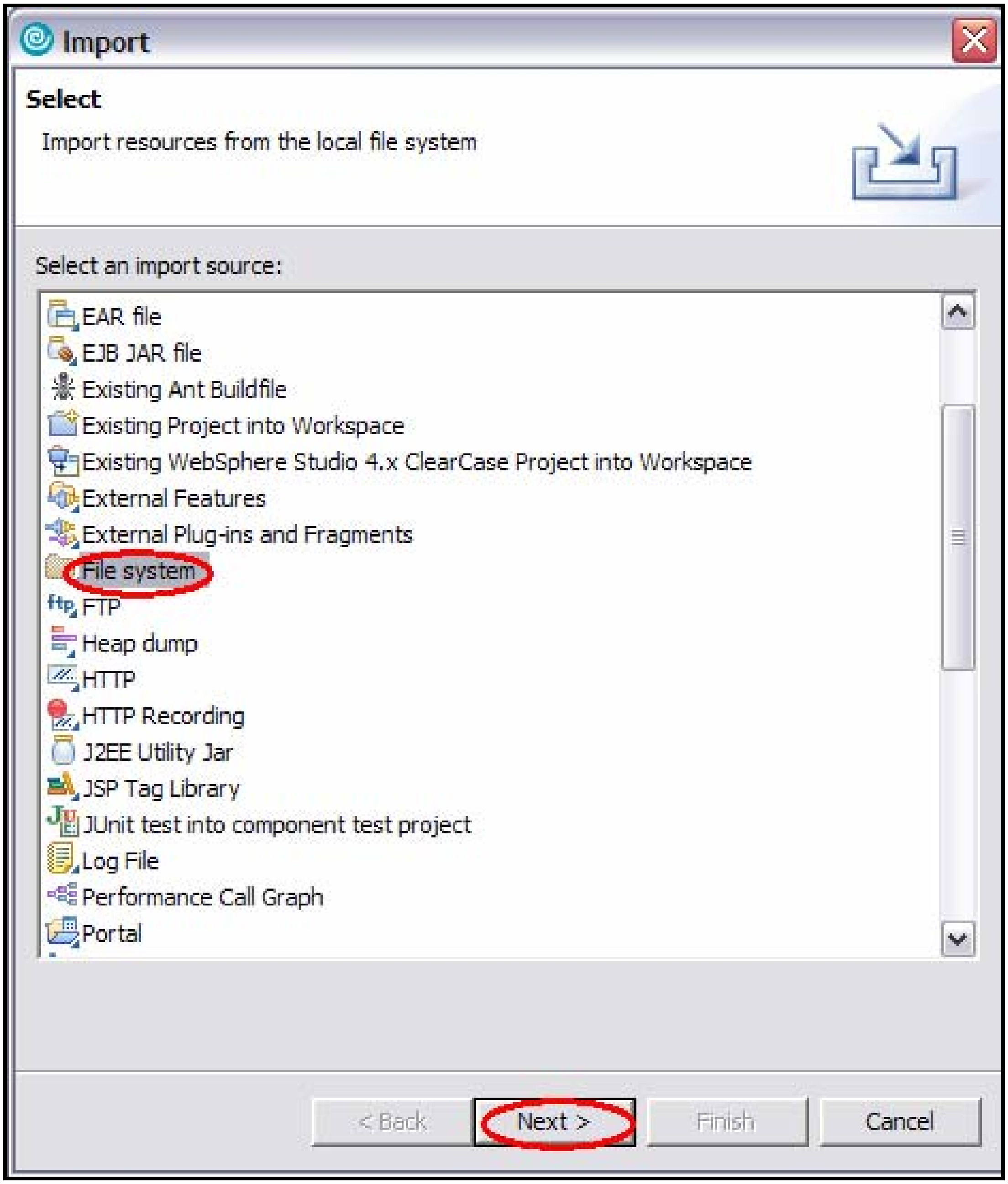


Figure 356. Switch to the Navigator view for the wrkord project.

2. Expand **WebContent > webfacing > ClientScript > usr**.
3. Select the **usr** folder.
4. Right-click the **usr** folder and from the pop-up menu, select **Import**.
5. The Select page of the Import wizard is displayed (see

Figure ***357***).



Import

Select

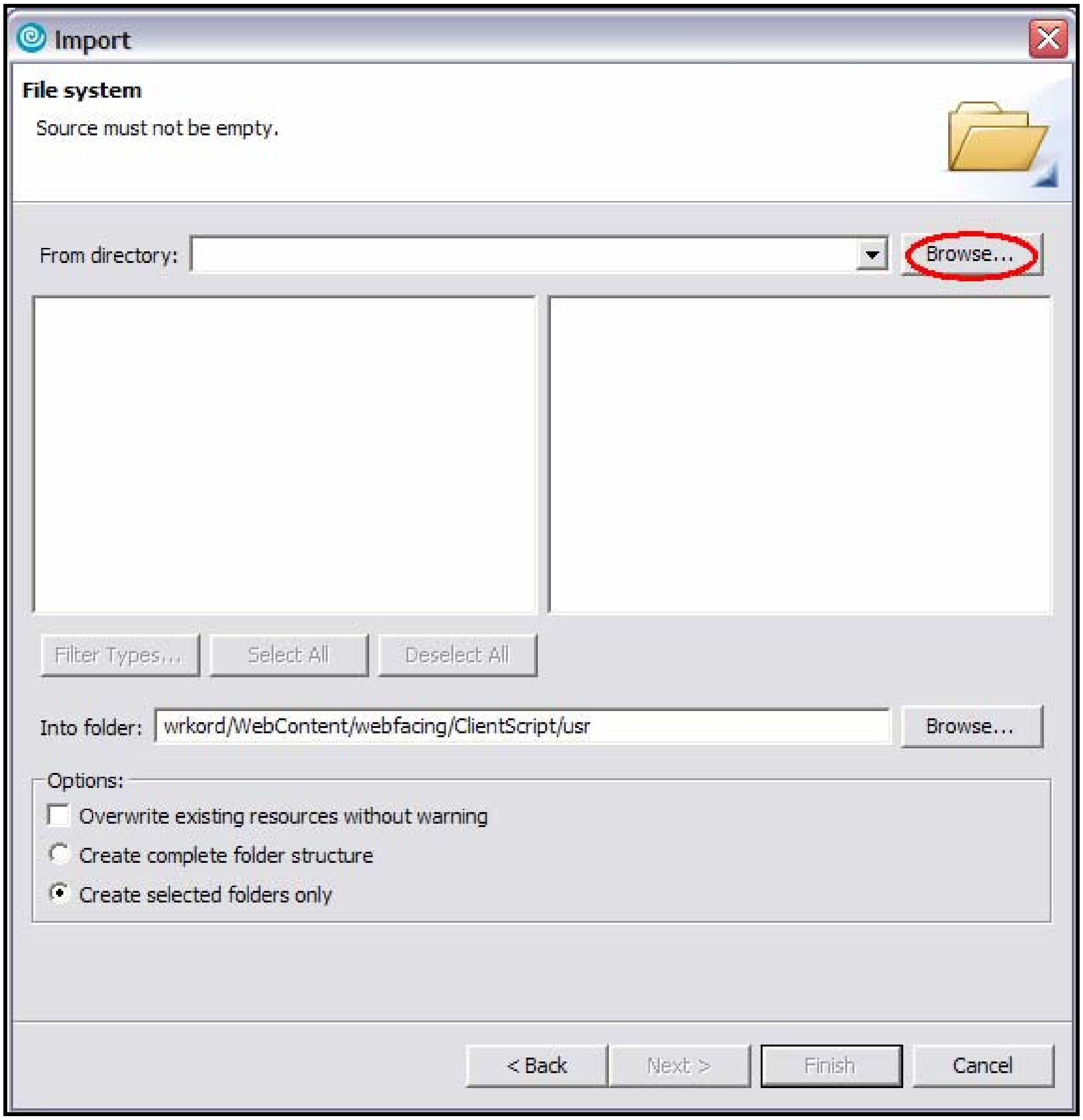
Import resources from the local file system

Select an import source:

Figure 357. The Select page

6. Select **File system** and click **Next**.

7. The File system page is displayed (see Figure ***358***).



Import

File system

Source must not be empty.

From directory:

Into folder: wrkord/WebContent/webfacing/ClientScript/usr

Options:

□ Overwrite existing resources without warning

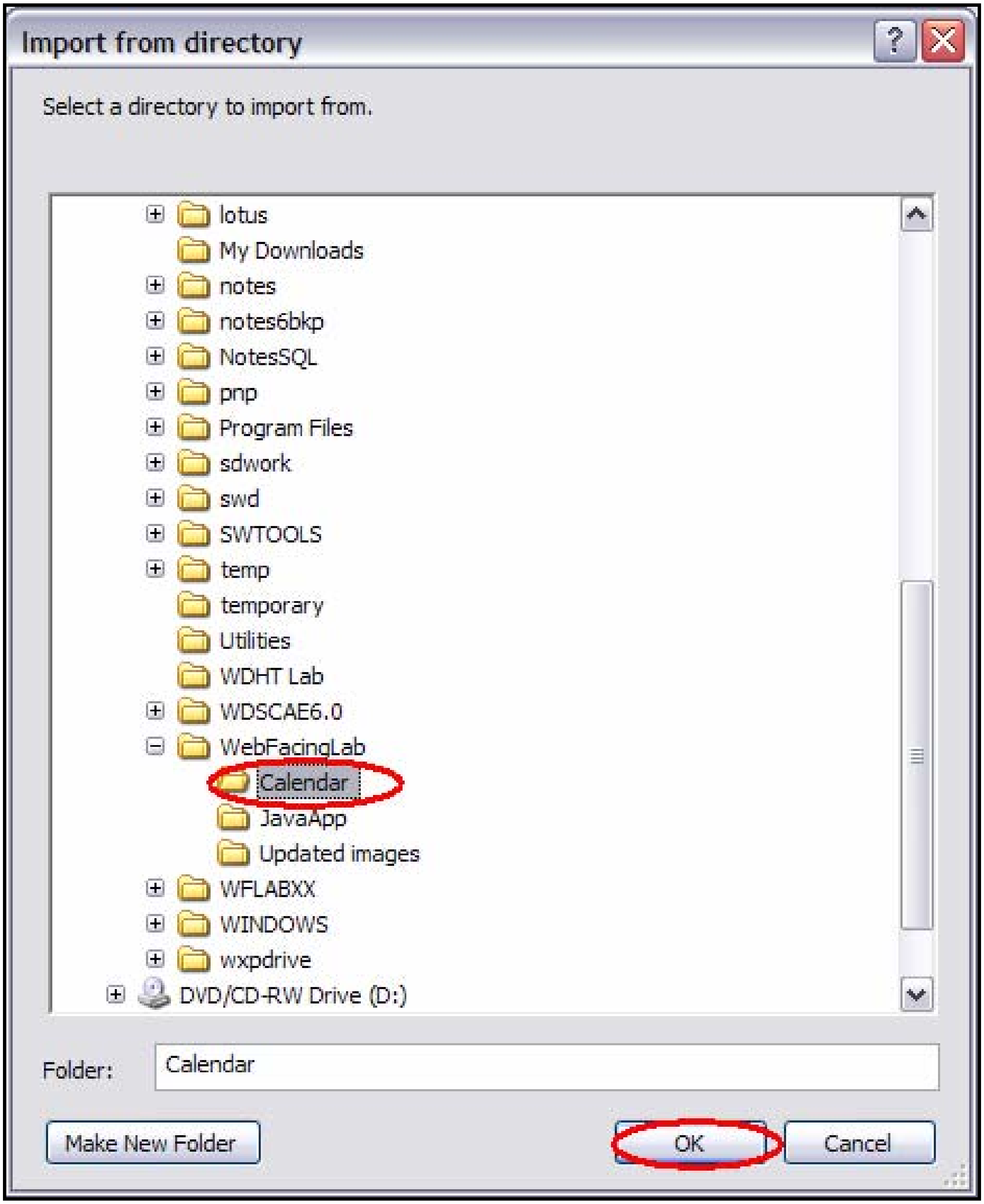
○ Create complete folder structure

● Create selected folders only

Figure 358. The File System page

8. Click the **Browse** button next to the **From directory:** field.

The Import from directory dialog opens (see Figure 359).



Import from directory

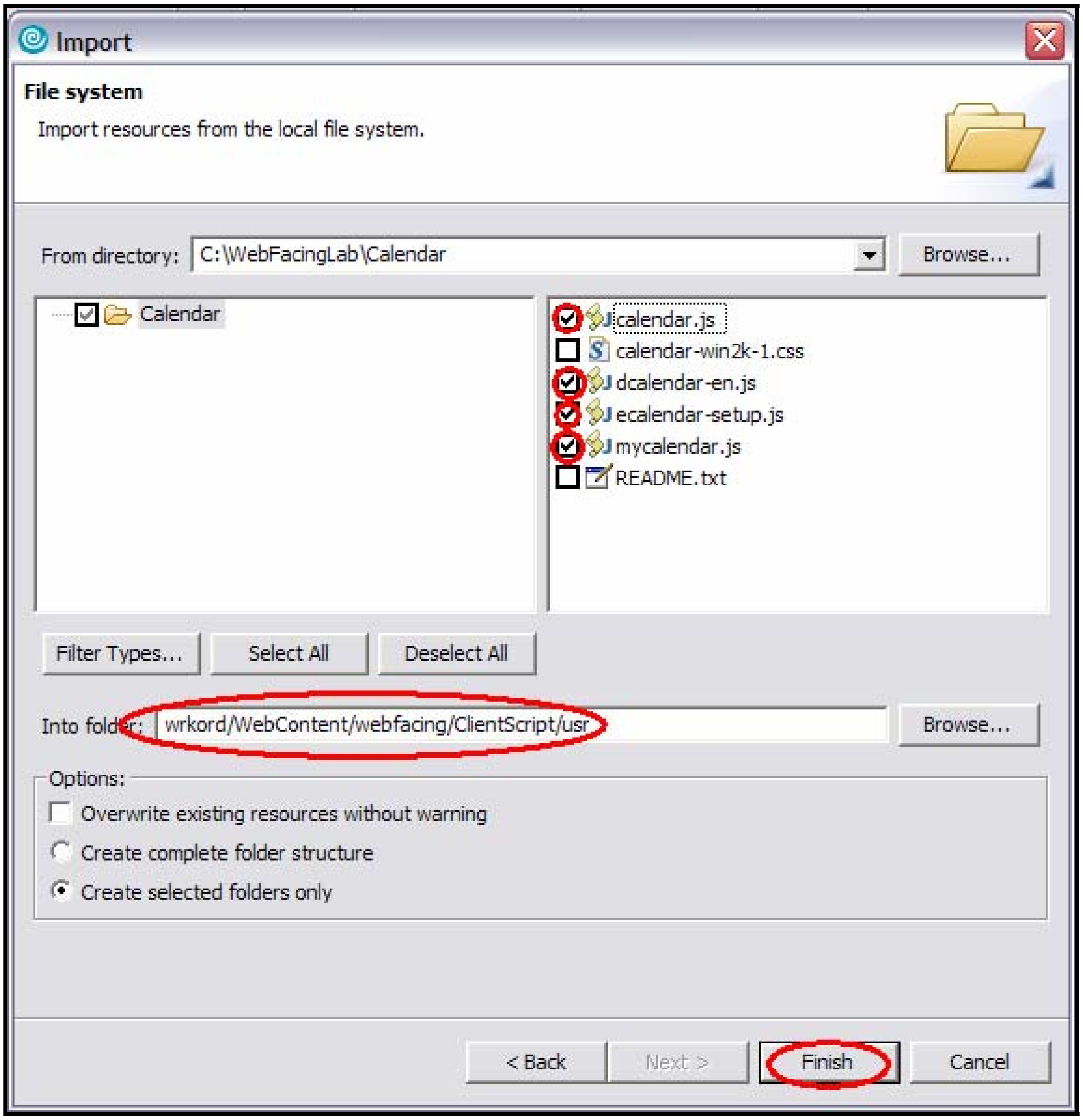
Select a directory to import from.

Folder: Calendar

Figure 359. The Import from directory dialog

9. On the **Import from directory** dialog, expand down to the **C:\WebFacingLab\Calendar** folder.
10. Click the **Calendar** folder, and click **OK**.

11. The File system dialog is displayed with the selected folder (see Figure 360).



Import

File system

Import resources from the local file system.

From directory: C:\WebFacingLab\Calendar

☑ Calendar

- ☑ calendar.js
- □ calendar-win2k-1.css
- ☑ dcalendar-en.js
- ☑ ecalendar-setup.js
- ☑ mycalendar.js
- □ README.txt

Into folder: wrkord/WebContent/webfacing/ClientScript/usr

Options:

- □ Overwrite existing resources without warning
- ○ Create complete folder structure
- ● Create selected folders only

Figure 360. The File system dialog

12. In the right pane, check the boxes for the files **calendar.js**, **dcalendar-en.js**, **ecalendar-setup.js** and **mycalendar.js**. Make sure the other check boxes are un-checked.
13. Ensure the **Into folder** field contains the path to the **usr** folder in the **wrkord** project.
14. Click **Finish** to import the files.
15. The Navigator view now shows the imported files in the **usr** folder (see Figure ***361***).

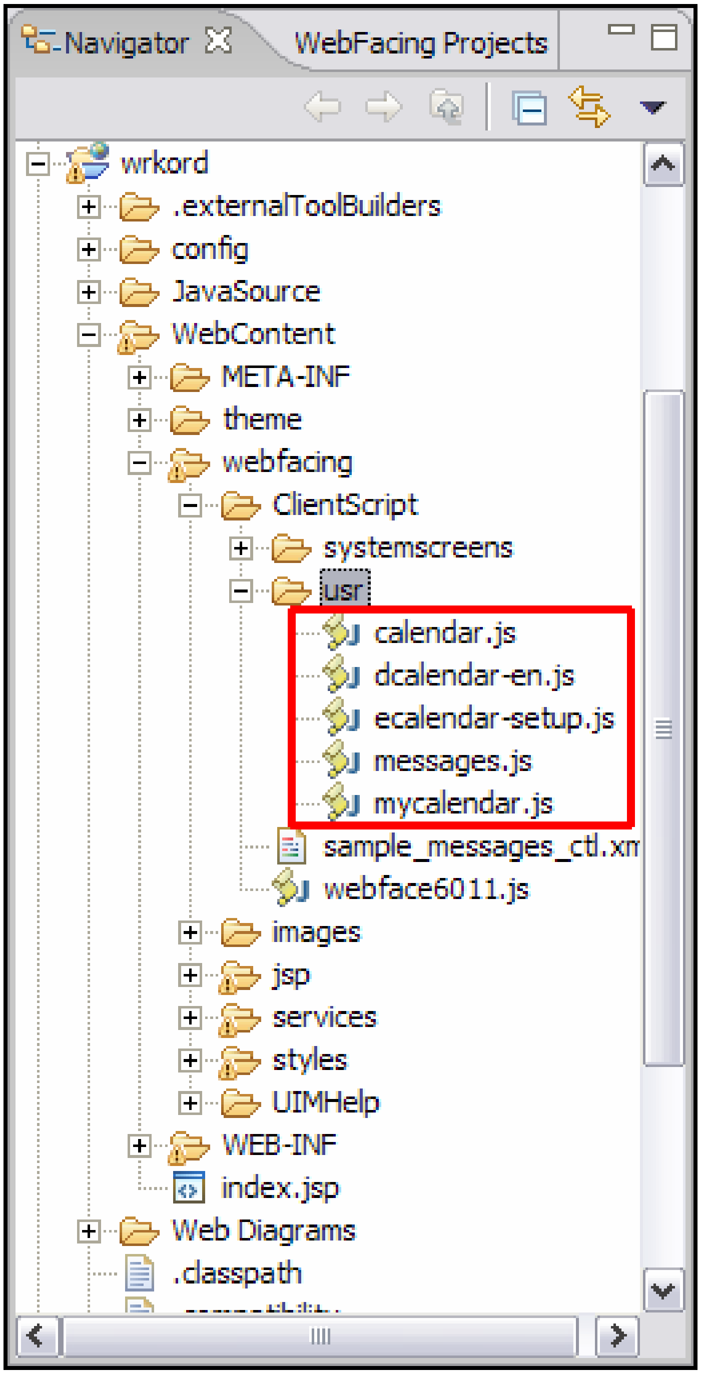


Figure 361. The Navigator view shows the imported files in the ***usr*** *folder.*

Importing the calendar style sheet

You need to import one more file into our project. This one is a Style Sheet that defines the look of the calendar. This style sheet must be imported into the **styles** folder of the project. In addition, a reference to the style sheet must be added to the PageBuilder.jsp JavaServer page.

To import the style sheet, begin by performing the following steps while referring to Figure **362**:

1. Ensure you are in the **Navigator** view.
2. Expand the **WebContent > webfacing > styles > chrome** folders.
3. Right-click the **chrome** folder, and select **Import** from the pop-up menu.

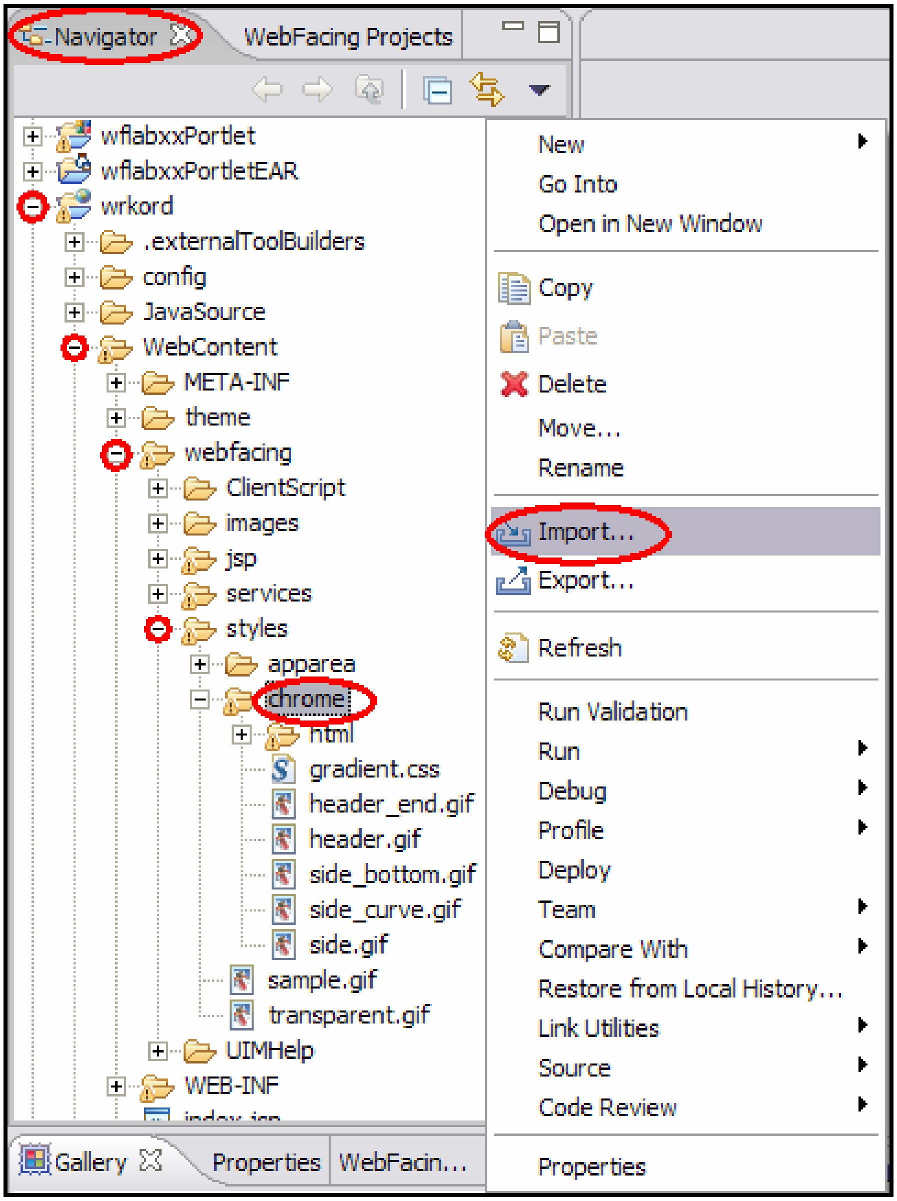
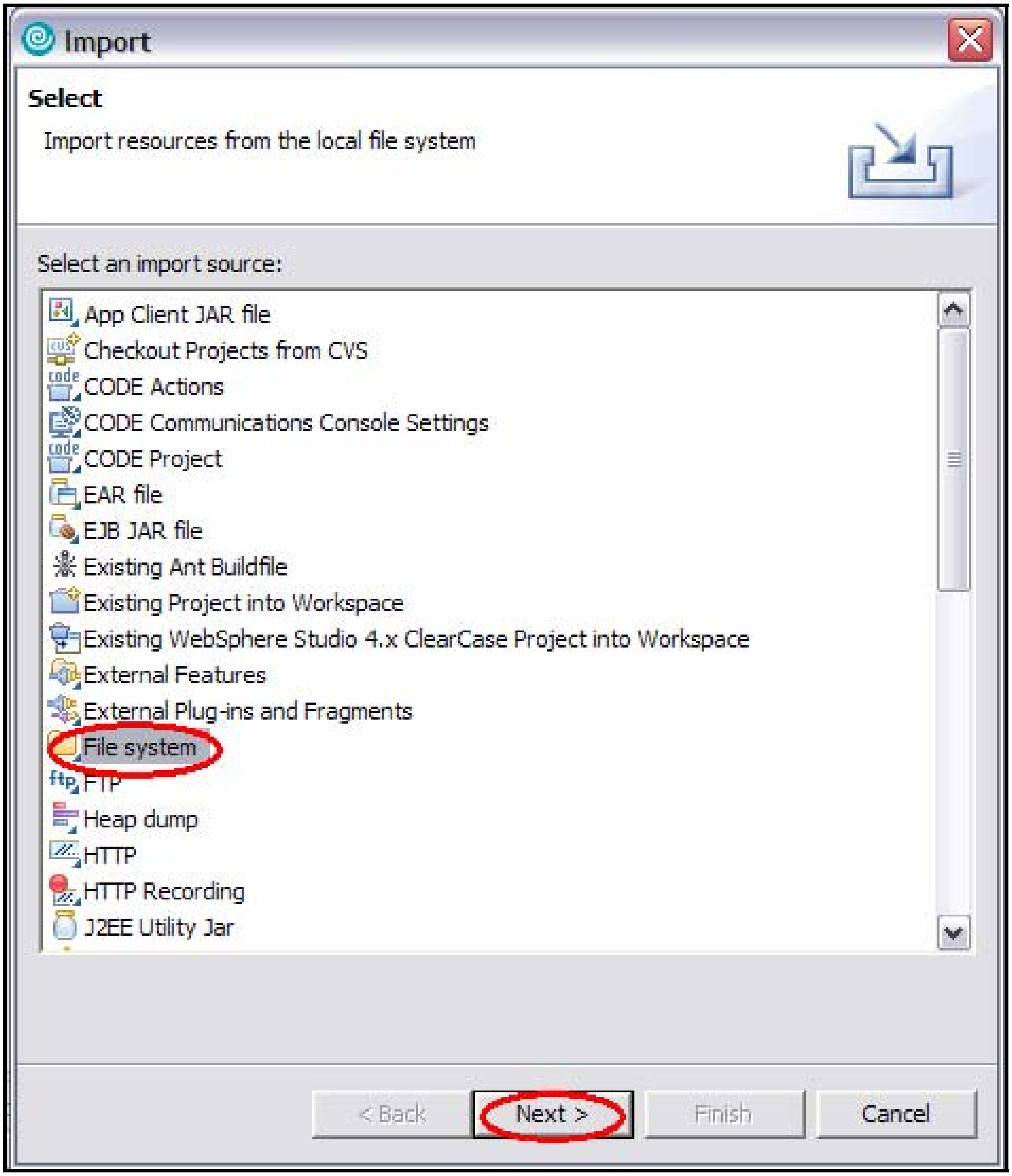


Figure 362. Importing the calendar style sheet

4. The Select page is displayed (see Figure ***363***).



Select

Import resources from the local file system

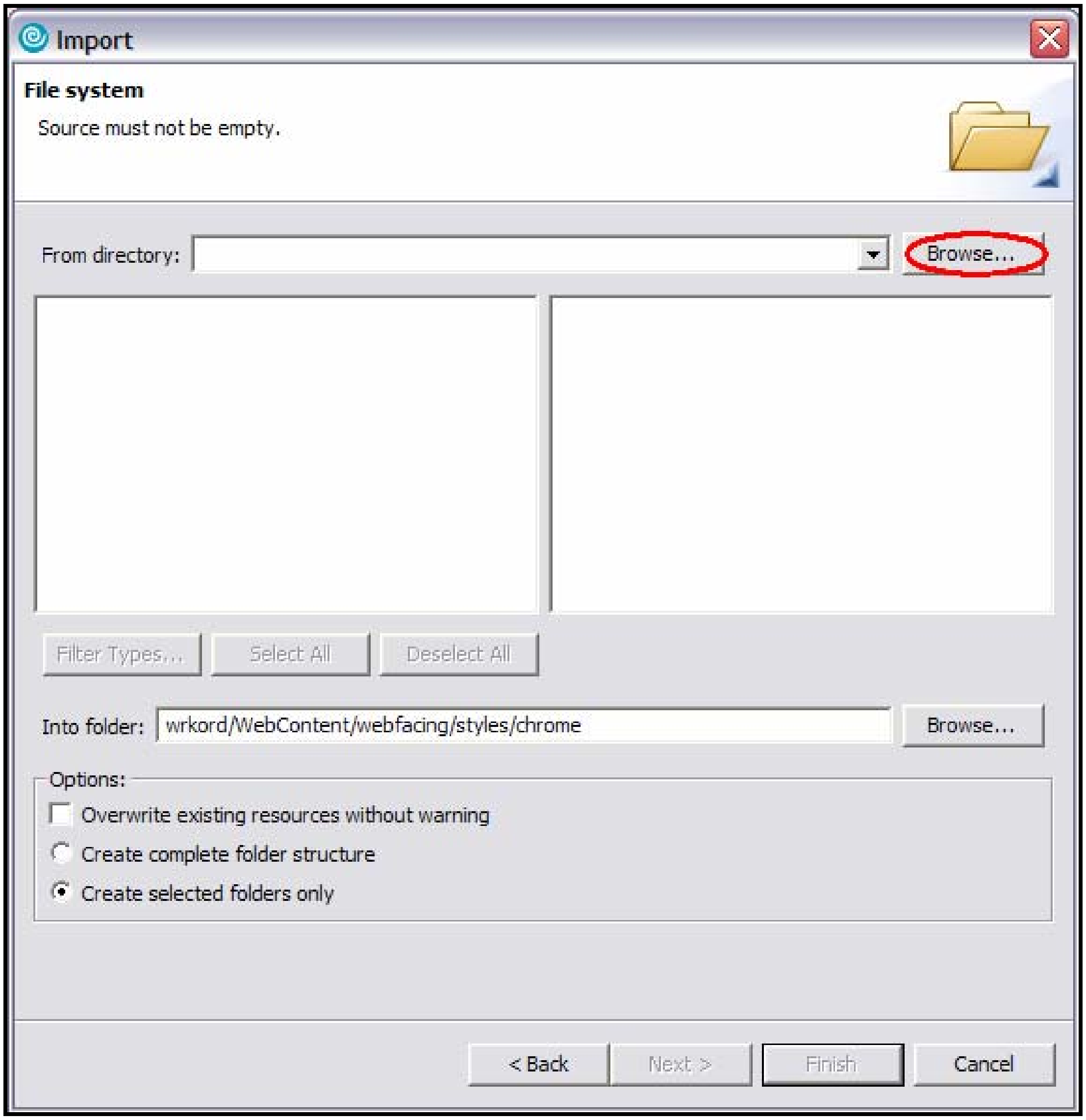
Select an import source:

- App Client JAR file
- Checkout Projects from CVS
- CODE Actions
- CODE Communications Console Settings
- CODE Project
- EAR file
- EJB JAR file
- Existing Ant Buildfile
- Existing Project into Workspace
- Existing WebSphere Studio 4.x ClearCase Project into Workspace
- External Features
- External Plug-ins and Fragments
- File system
- FTP
- Heap dump
- HTTP
- HTTP Recording
- J2EE Utility Jar

Figure 363. The Select page

5. Select **File system** and click **Next**.

6. The File system page is displayed (see Figure ***364***).



Import

File system

Source must not be empty.

From directory:

Into folder: wrkord/WebContent/webfacing/styles/chrome

Options:

□ Overwrite existing resources without warning

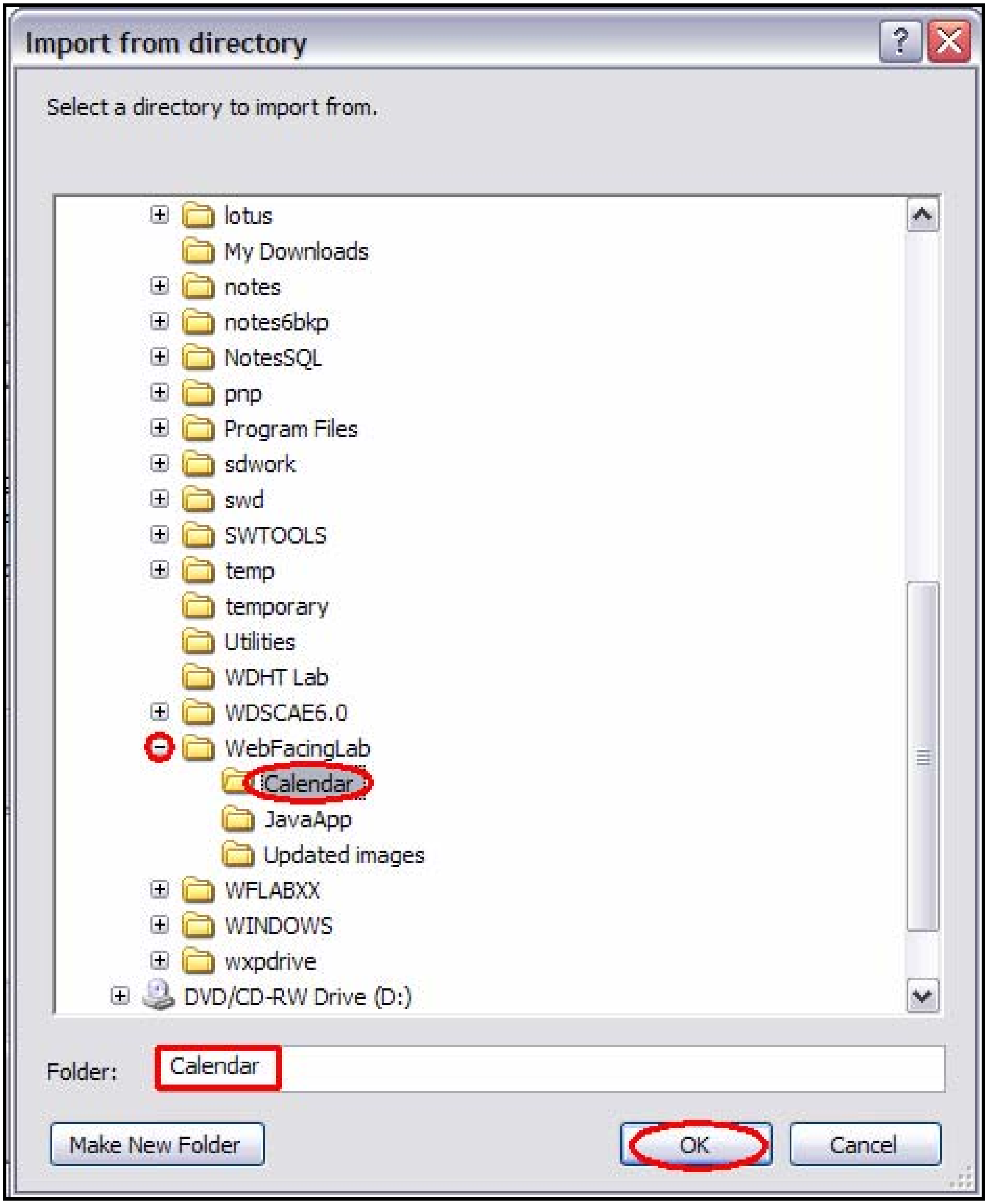
○ Create complete folder structure

● Create selected folders only

Figure 364. The File system page

7. Click the **Browse** button.

8. The Import from directory dialog is displayed (see Figure ***365***).

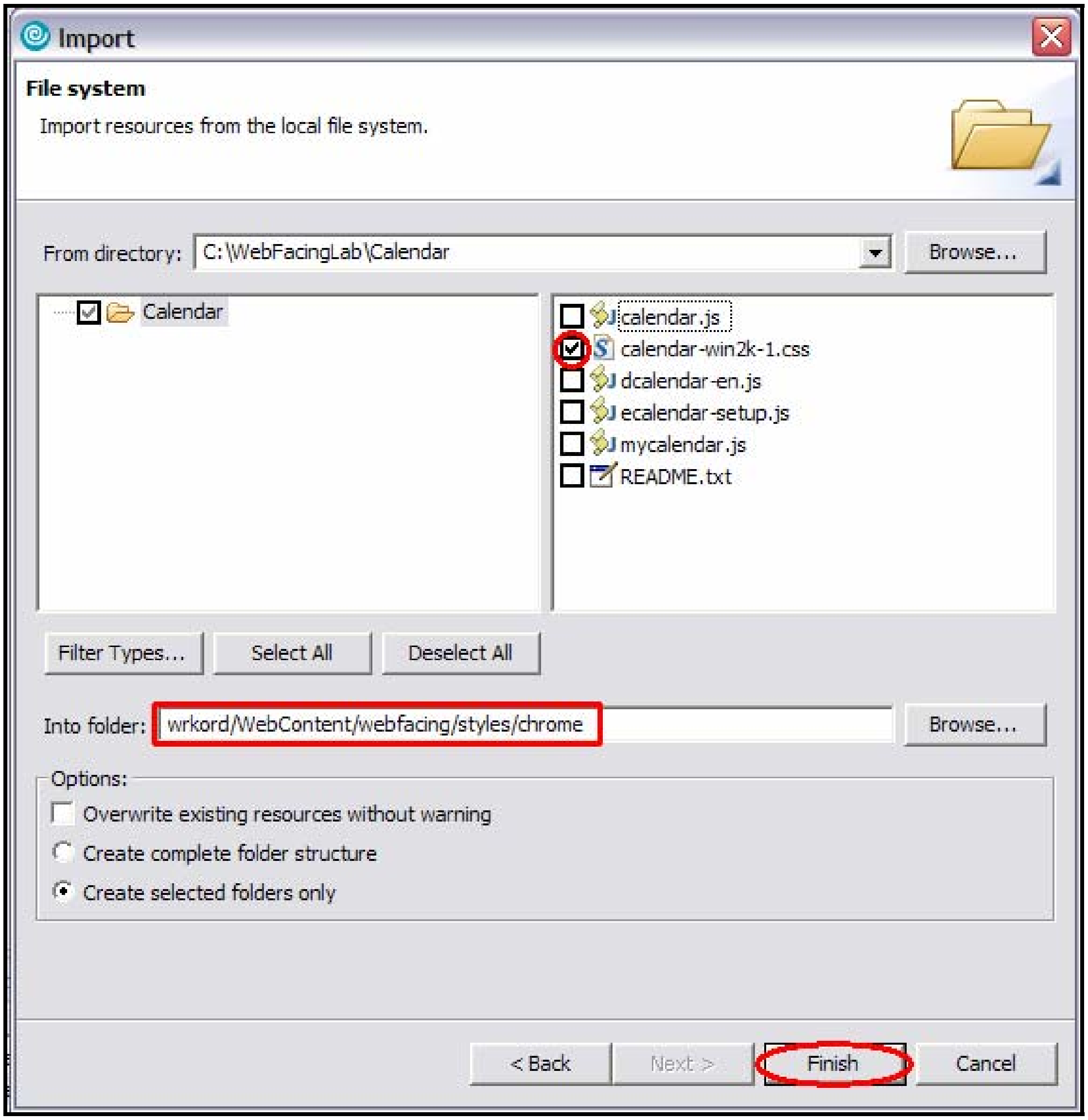


Folder: Calendar

Figure 365. The Import from directory dialog

9. Expand down to the **C:\WebFacingLab\Calendar** folder.
10. Click **OK**.

11. The File system page is displayed with the selected folder (see Figure ***366***).



Import

File system

Import resources from the local file system.

From directory: C:\WebFacingLab\Calendar

☑ Calendar

- □ calendar.js
- ☑ calendar-win2k-1.css
- □ dcalendar-en.js
- □ ecalendar-setup.js
- □ mycalendar.js
- □ README.txt

Into folder: wrkord/WebContent/webfacing/styles/chrome

Options:

- □ Overwrite existing resources without warning
- ○ Create complete folder structure
- ● Create selected folders only

Figure 366. The File system page is displayed with the selected folder.

12. In the right pane, check the box for the **calendar-win2k-1.css** file.
13. Ensure the **Into folder** field contains the path to the **chrome** folder in the **wrkord** project.
14. Click **Finish** to import the style sheet.
15. The **Navigator** view is shown with the imported style sheet (see Figure ***367***).

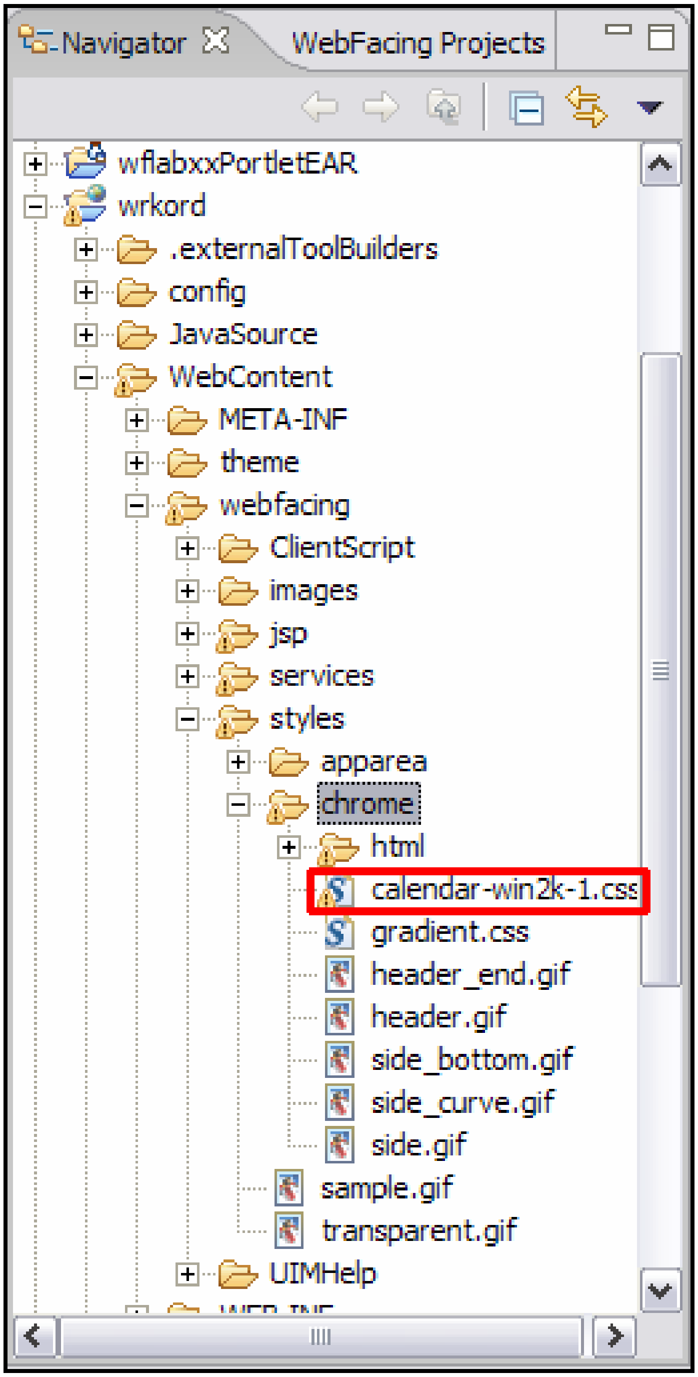


Figure 367. The Navigator view is shown with the imported style sheet.

Adding a reference to the style sheet to PageBuilder.jsp

To use a style sheet, a link to it must be made in a JSP. Because PageBuilder.jsp is sent to the browser after each request, you put the link to the style sheet in that page.

To put the style sheet link in PageBuilder.jsp, begin by performing the following steps while referring to
Figure ***368***:

1. Ensure you are in the **Navigator** view.
2. Expand **WebContent > webfacing > styles > chrome > html.**
3. Double-click **PageBuilder.jsp**.

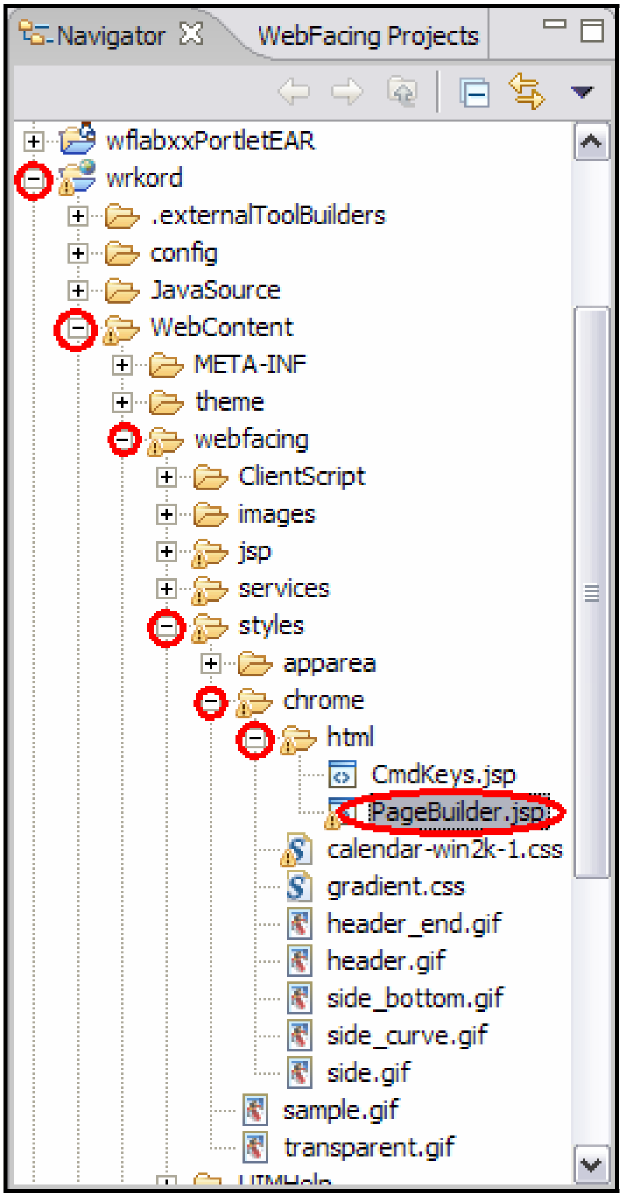
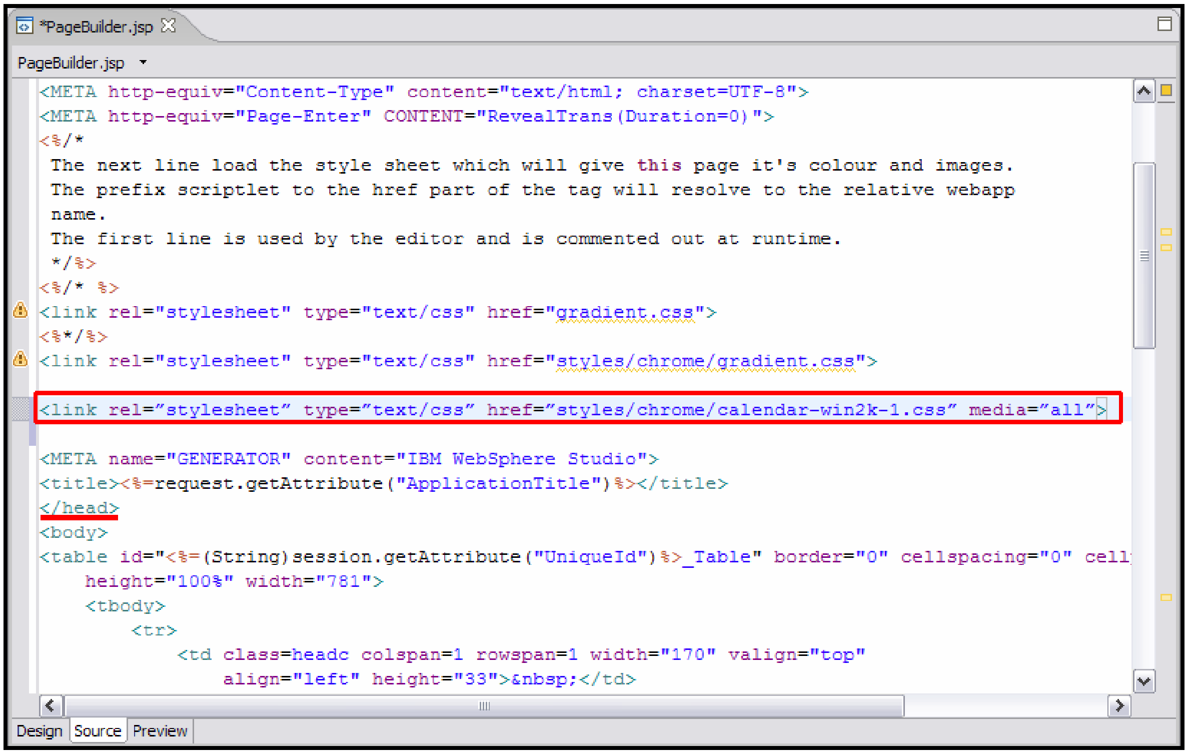


Figure 368. Adding a reference to the style sheet to PageBuilder.jsp

4. **Page designer** opens with PageBuilder.jsp (see Figure ***369***).



```
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
<META http-equiv="Page-Enter" CONTENT="RevealTrans(Duration=0)">
<%/*
 The next line load the style sheet which will give this page it's colour and images.
 The prefix scriptlet to the href part of the tag will resolve to the relative webapp
 name.
 The first line is used by the editor and is commented out at runtime.
 */%>
<%/* %>
<link rel="stylesheet" type="text/css" href="gradient.css">
<%*/%>
<link rel="stylesheet" type="text/css" href="styles/chrome/gradient.css">

<link rel=”stylesheet” type=”text/css” href=”styles/chrome/calendar-win2k-1.css” media=”all”>

<META name="GENERATOR" content="IBM WebSphere Studio">
<title><%=request.getAttribute("ApplicationTitle")%></title>
</head>
<body>
<table id="<%=(String)session.getAttribute("UniqueId")%>_Table" border="0" cellspacing="0" cell
    height="100%" width="781">
    <tbody>
        <tr>
            <td class=headc colspan=1 rowspan=1 width="170" valign="top"
                align="left" height="33">&nbsp;</td>
```

Figure 369. Page designer opens

5. Ensure the **Source** tab is selected.
6. Insert the following statement before the **</head>** tag:
   `<link rel=”stylesheet” type=”text/css”`
   `href=”styles/chrome/calendar-win2k-1.css” media=”all”>`
7. Press **Ctrl+s** to save your changes.

Exercise 18.3: Adding the Web Settings to invoke the calendar pop-up

As mentioned, you add the pop-up calendar to the previously created project **wrkord**. To invoke the pop-up calendar, add a hyperlink to a Web page using the Web Settings feature of the IBM WebFacing Tool.

To add the Web Settings, begin by performing the following steps while referrinf to Figure ***370***:

1. Ensure you are in the WebFacing perspective.
2. Click the **WebFacing Projects** tab to switch to the WebFacing Projects view.

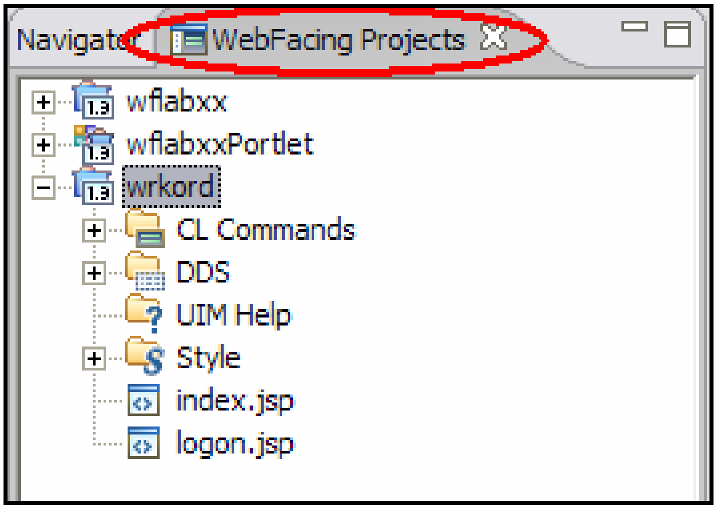


Figure 370. Adding Web Settings to invoke calendar pop-up

3. Expand the **wrkord > DDS** folders (see Figure 371).

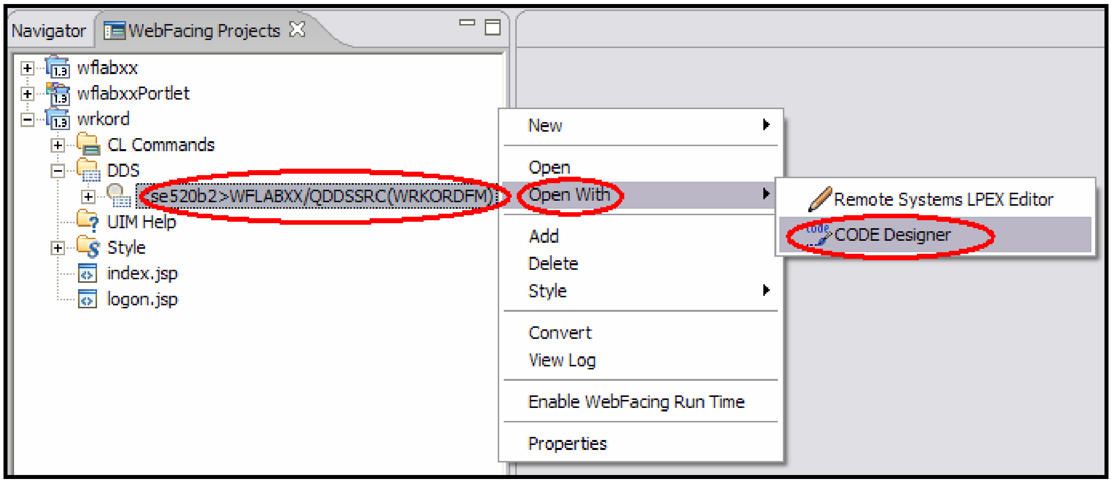


*Figure 371. Expand the **wrkord > DDS** folders.*

4. Right-click the **WRKORDFM** member.
5. Select **Open With > CODE Designer** from the pop-up menu. If prompted, specify your User ID and password.
6. CODE designer opens with the selected member and the **Group properties** dialog is displayed (see Figure 372):

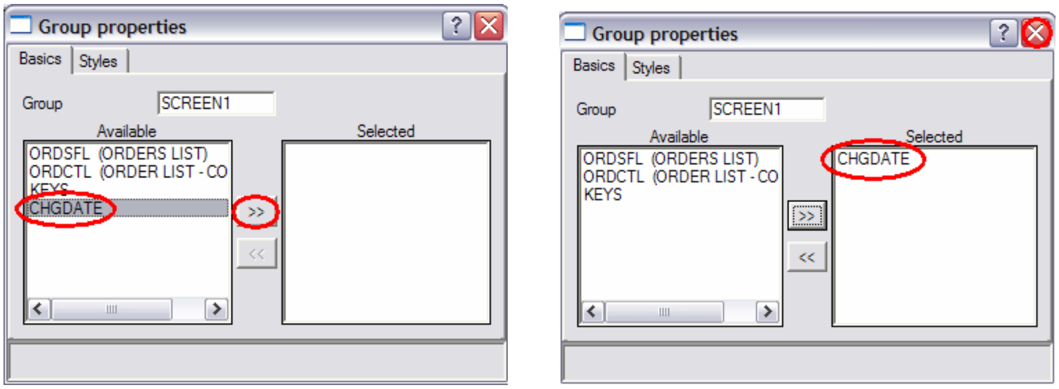
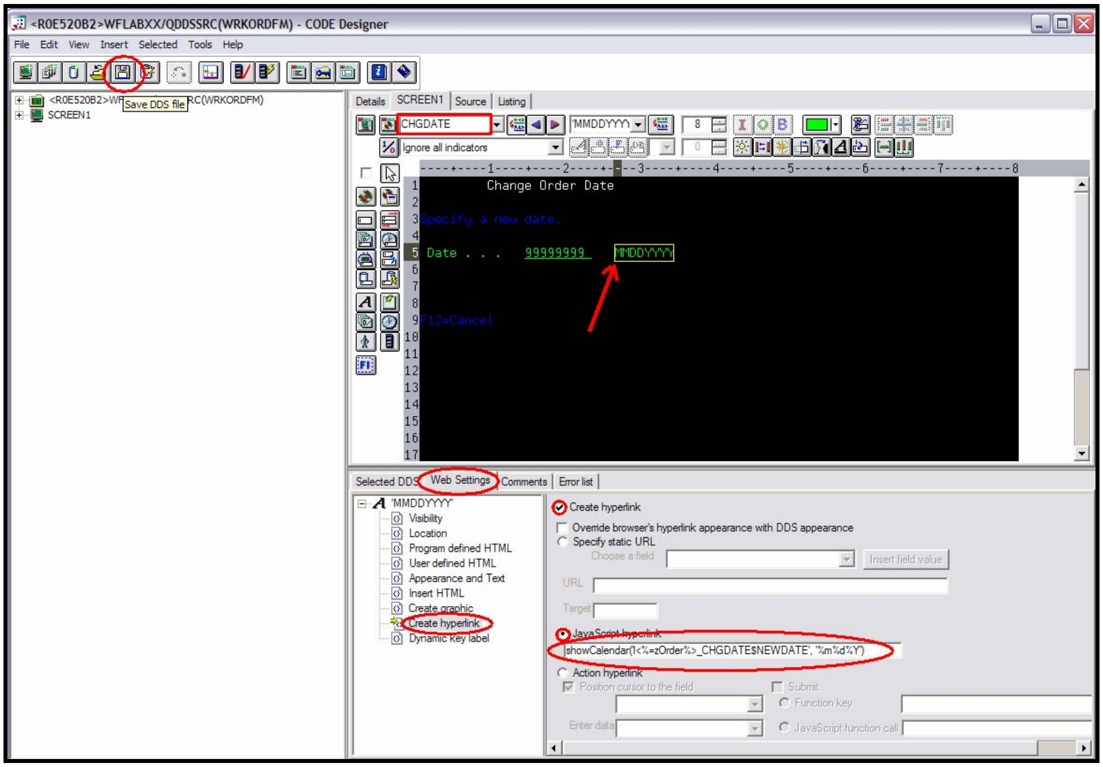


Figure 372. Group properties dialog

7. Select **CHGDATE** in the **Available** list, and click the >> button to move this format to the **Selected** list.
8. Click the **X** (at the top of the browser) to close the Group properties dialog.

9. The selected format, CHGDATE is displayed in CODE Designer (see Figure 373)


<R0E520B2>WFLABXX/QDDSSRC(WRKORDFM) - CODE Designer
File Edit View Insert Selected Tools Help
Save DDS file
<R0E520B2>WF RC(WRKORDFM)
SCREEN1
Details SCREEN1 Source Listing
Selected DDS Web Settings Comments Error list
'MMDDYYYY'
Visibility
Location
Program defined HTML
User defined HTML
Appearance and Text
Insert HTML
Create graphic
Create hyperlink
Dynamic key label

CHGDATE
MMDDYYYY
8
Ignore all indicators

```
         Change Order Date
Specify a new date.
 Date . . .   99999999    MMDDYYYY
F12=Cancel
```

☑ Create hyperlink
□ Override browser's hyperlink appearance with DDS appearance
○ Specify static URL
Choose a field
Insert field value
URL
Target
● JavaScript hyperlink
showCalendar(l<%=zOrder%>_CHGDATE$NEWDATE', '%m%d%Y')
○ Action hyperlink
☑ Position cursor to the field
□ Submit
○ Function key
Enter data
○ JavaScript function call

Figure 373. CHGDATE is displayed

10. Select the **MMDDYYY** constant.
11. Click the **Web Settings** tab.
12. In the left pane, select **Create hyperlink**.
13. In the right pane, select the check box **Create hyperlink**.
14. Also, in the right pane, select the radio button labeled **JavaScript hyperlink**.
15. In the entry field type:
    `showCalendar(‘l<%=zOrder%>_CHGDATE$NEWDATE’, ‘%m%d%Y’)`
    (**Note:** The ‘`l`’ is a lower case L. Type this data exactly as shown.)
16. Click the save icon 🖫 in the tool bar to save your changes.
17. Close CODE Designer.

Exercise 18.4: Convert the project with the new Web Settings

To pick up the all of the changes made to the project, it must be converted again.

To convert the project, perform the following steps while referring to

Figure ***374***:

1. Ensure you are in the WebFacing Projects view.
2. Expand the **wrkord** > **DDS** folders.
3. Right-click the **WRKORDFM DDS** member.
4. Select **Convert** from the pop-up menu.
5. The project is converted.

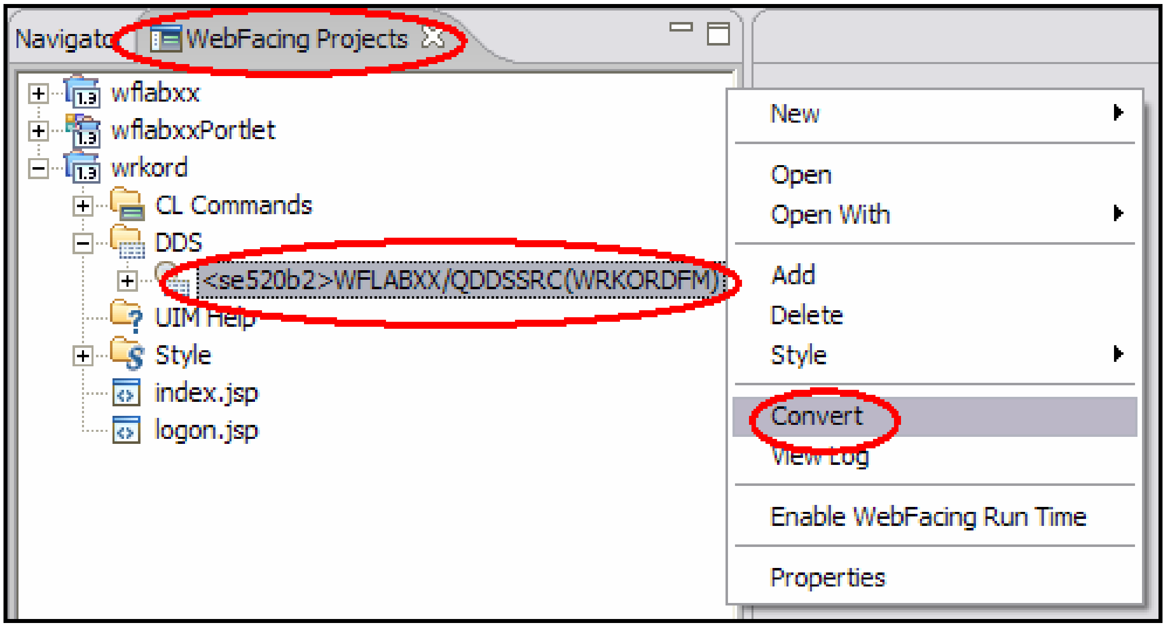


Figure 374. Converting the project with the new Web Settings

Exercise 18.5: Test the application with the JavaScript calendar

Now, you can test the project to see how the pop-up calendar works.

To test the project (see
Figure ***375***):

1. Ensure you are in the **WebFacing Projects** view.
2. Right-click the **wrkord** project and select **Run on Server** from the pop-up menu.

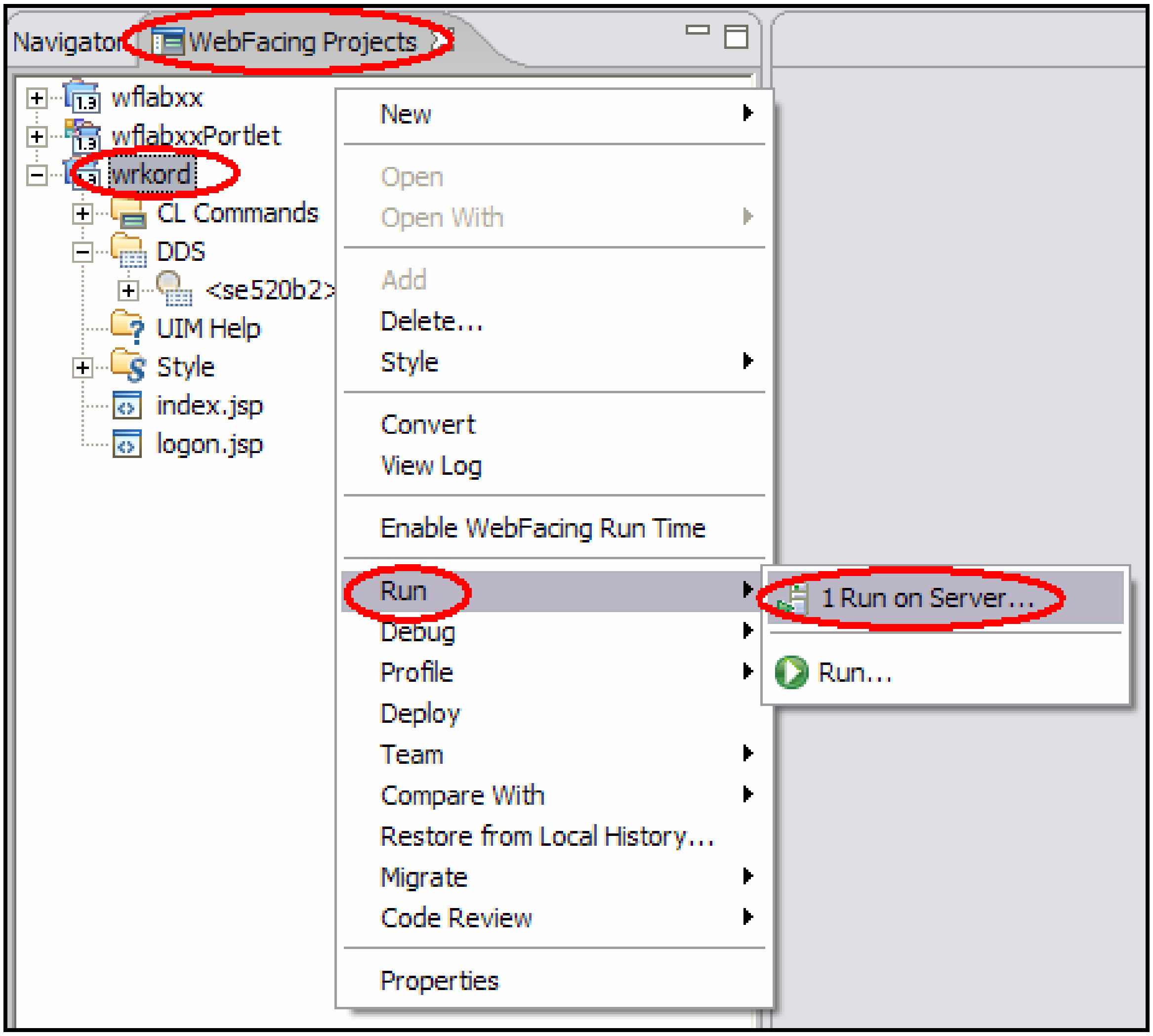
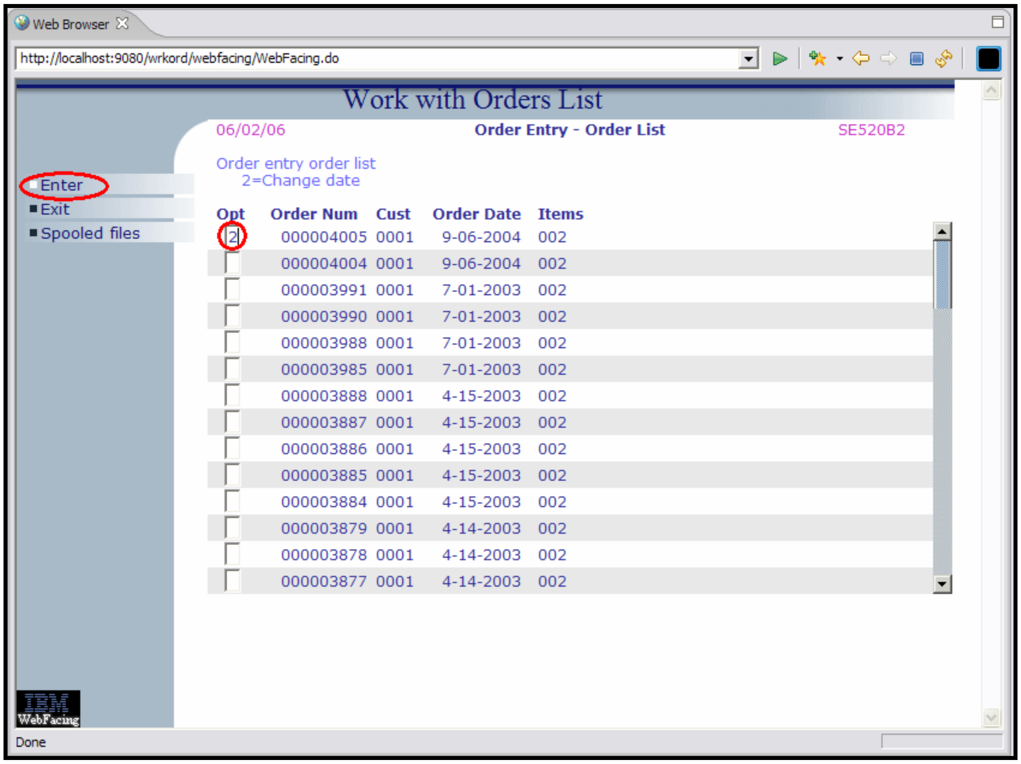


Figure 375. Test the application with the JavaScript calendar

3. When the index.jsp is displayed in the internal browser, click the **Launch** button.
4. The first screen of the application is displayed (see Figure 376).
5. Type 2 in any of the **Opt** fields and click **Enter**.



http://localhost:9080/wrkord/webfacing/WebFacing.do

Work with Orders List

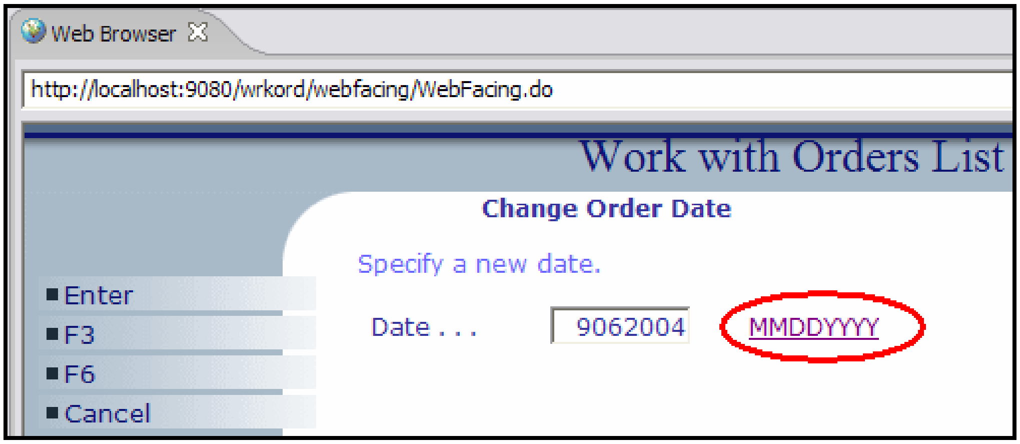
06/02/06 Order Entry - Order List SE520B2

Order entry order list
2=Change date

Opt	Order Num	Cust	Order Date	Items
2	000004005	0001	9-06-2004	002
	000004004	0001	9-06-2004	002
	000003991	0001	7-01-2003	002
	000003990	0001	7-01-2003	002
	000003988	0001	7-01-2003	002
	000003985	0001	7-01-2003	002
	000003888	0001	4-15-2003	002
	000003887	0001	4-15-2003	002
	000003886	0001	4-15-2003	002
	000003885	0001	4-15-2003	002
	000003884	0001	4-15-2003	002
	000003879	0001	4-14-2003	002
	000003878	0001	4-14-2003	002
	000003877	0001	4-14-2003	002

Figure 376. The first screen of the application is displayed.

6. The Change Order Date display is shown (see
Figure ***377***).
7. Notice the **MMDDYYYY** constant is now a hyperlink.
8. Select the hyperlink.



http://localhost:9080/wrkord/webfacing/WebFacing.do

Work with Orders List

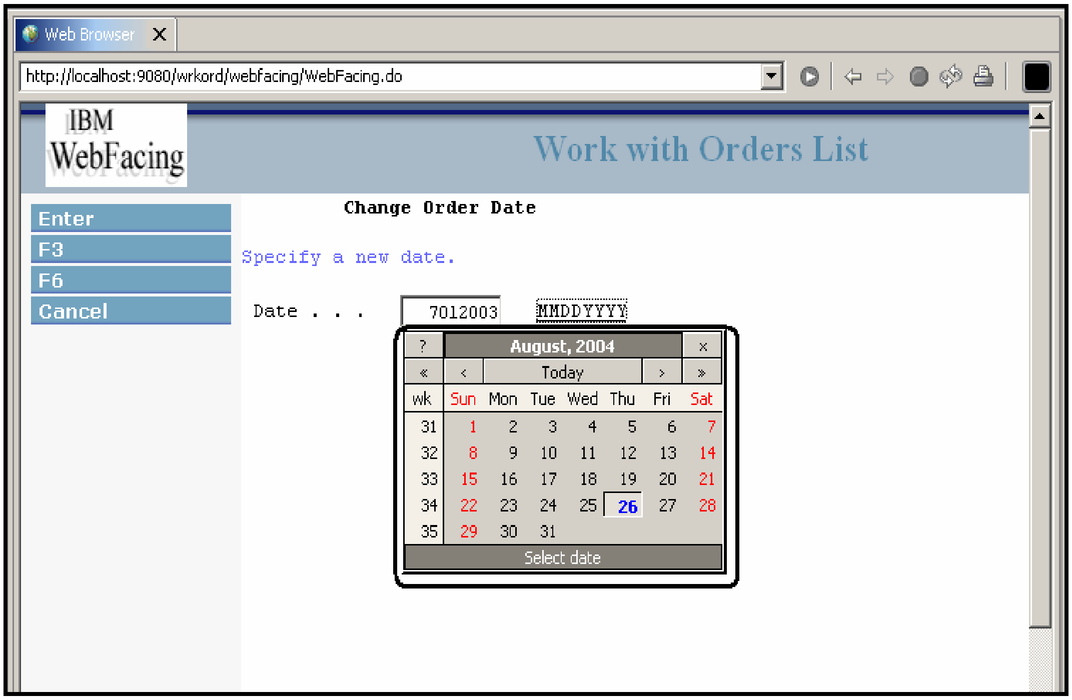
Change Order Date

Specify a new date.

Date . . . 9062004 MMDDYYYY

Figure 377. The Change Order Date display

9. The pop-up calendar is displayed (see Figure ***378***).



Change Order Date

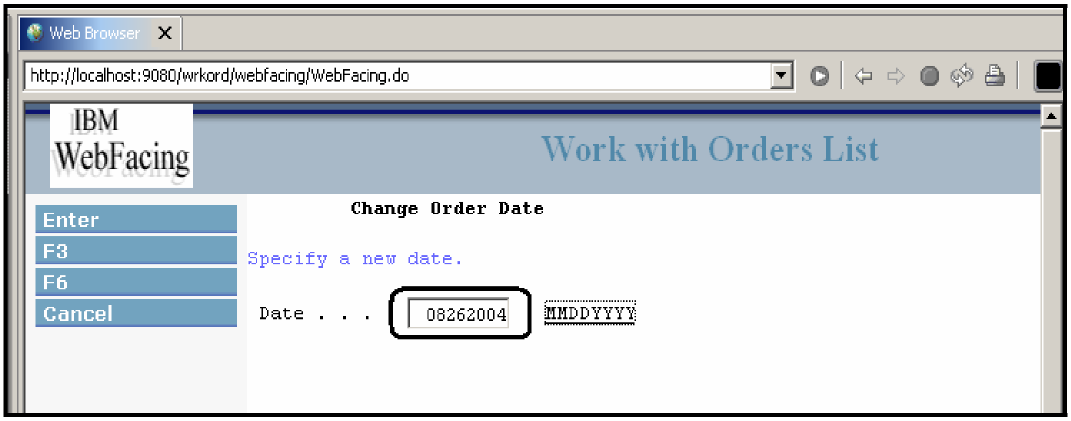
Specify a new date.

Date . . . 7012003 MMDDYYYY

wk	Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6	7
32	8	9	10	11	12	13	14
33	15	16	17	18	19	20	21
34	22	23	24	25	26	27	28
35	29	30	31				

Figure 378. Pop-up calendar

10. Click a day in the calendar.
11. The new date is now in the Date field (see Figure 379).



Change Order Date

Specify a new date.

Date . . . 08262004 MMDDYYYY

*Figure 379. The new date is in the **Date** field.*

12. Press **Enter** to return to the first screen with the updated date.

Recap

You have completed “Adding a pop-up calendar.” You now have the information to understand how to:

- Add additional JavaScript to your WebFacing project
- Add additional style sheets to your WebFacing project
- Reference style sheets in your WebFacing project
- Add Web Settings to invoke the JavaScript
- Test the project to verify the added files work as expected

As mentioned, there are many other JavaScript calendars on the Internet that can be used for this purpose. With an intermediate level of understanding of JavaScript you can create your own customized JavaScript pop-up calendar.

There are many ways of invoking the pop-up calendar. In this exercise, you created a hyperlink for a constant field that invoked the JavaScript function. You can also have created a graphic next the Date field that when clicked invoke the JavaScript function. Another possibility, if there is limited room on the screen, is to allow the user to double-click in the Date field to invoke the pop-up calendar.

Advanced Web Settings example

To enhance the results of the IBM WebFacing Tool, you need to understand the JSPs that are created for each record format that you have converted. These JSPs simply represent and use the same data and field names that are in your original DDS source. Understanding these JSPs gives you the power to enhance the user interface of your WebFacing application. There are two ways to make changes to your WebFacing application, you can use Web Settings or you can directly edit the generated JSPs. Customizations made using Web Settings are saved as comments in your DDS and are automatically regenerated during future conversions. You can also edit JSP files directly using the powerful HTML, Java, and JavaScript editing features of the Page Designer tool. However, changes made in this way are lost on reconversion unless you save your edits and manually reinsert them after re-conversion. It is also possible to save these direct edits in a Web Setting and not lose the changes on reconversion. This last approach gives you benefits of both Page Designer and Web Settings.

In this chapter, you learn how to change the user interface of your WebFacing application by directly editing JSPs using Page Designer. You add JavaScript to a JSP so that subfile rows change color when the mouse moves over them. You then learn how to save this modification using Web Settings in CODE Designer so that your changes are saved when you reconvert the DDS member.

To accomplish these learning objects, several steps are involved, including:

- Exercise 19.1: Understanding and editing JSPs using Page Designer
- Exercise 19.2: Change the color of a subfile row example using Page Designer
- Exercise 19.3: Using Web Settings to save your edits for re-conversion

The exercises within each chapter must be completed in order. Start with “Exercise 19.1: Understanding and editing JSPs using Page Designer.”

Length of time

This chapter takes approximately 40 minutes to complete.

Exercise 19.1: Understanding and editing JSPs using Page Designer

In this exercise, you use WebSphere Development Studio Client to edit JSPs using Page Designer and add Web Settings using CODE Designer. If WebSphere Development Studio Client is not running, start it now.

Ensure you are in the WebFacing perspective with your project, and you are in the WebFacing Projects view.

Select record JSP to edit using Page Designer

To open the JSP source file using Page Designer:

1. Select the WebFacing project you have been working on in the previous chapter, wflabxx.
2. Expand this project by clicking the plus sign (**+**) beside its icon in the WebFacing Projects view.
3. Expand the ORDENTD DDS file and then the ORDCTL record format by clicking the plus sign (**+**) beside their icons. Now right-click the ORDCTL.jsp file and click **Open With** and **Page Designer** on the pop-up menu. The Page Designer tool opens the ORDCTL.jsp file (see Figure 380).

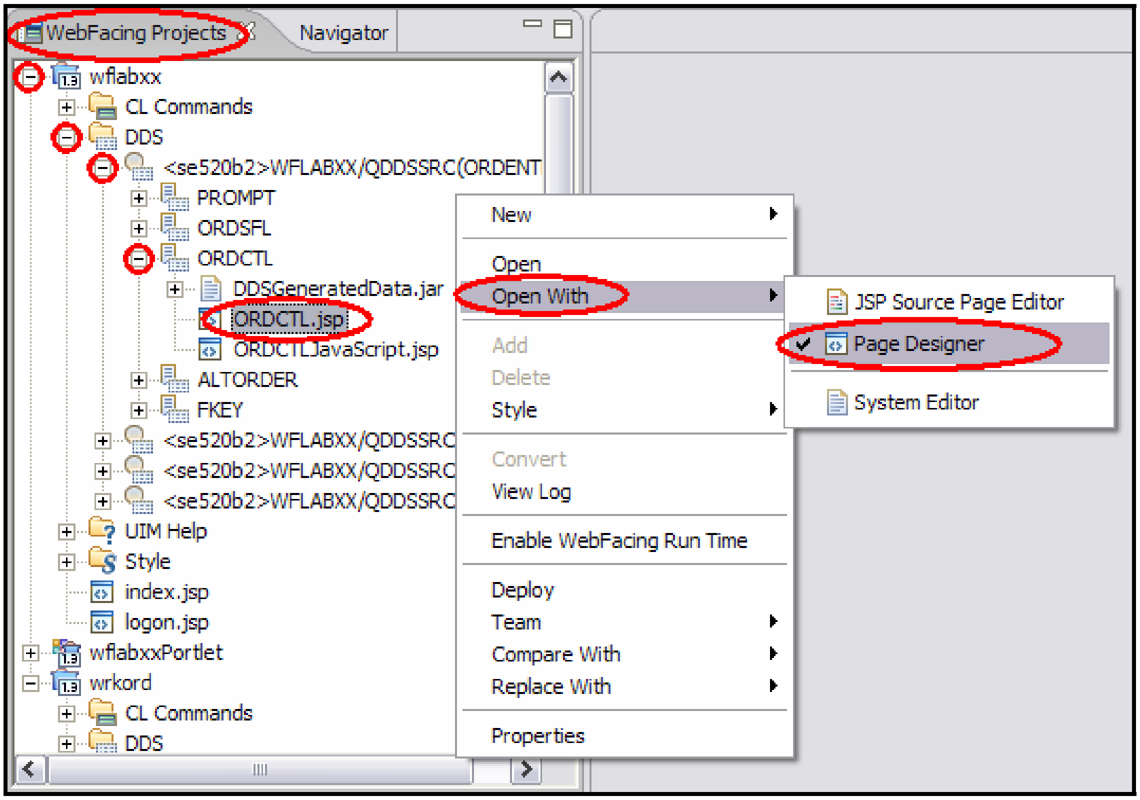
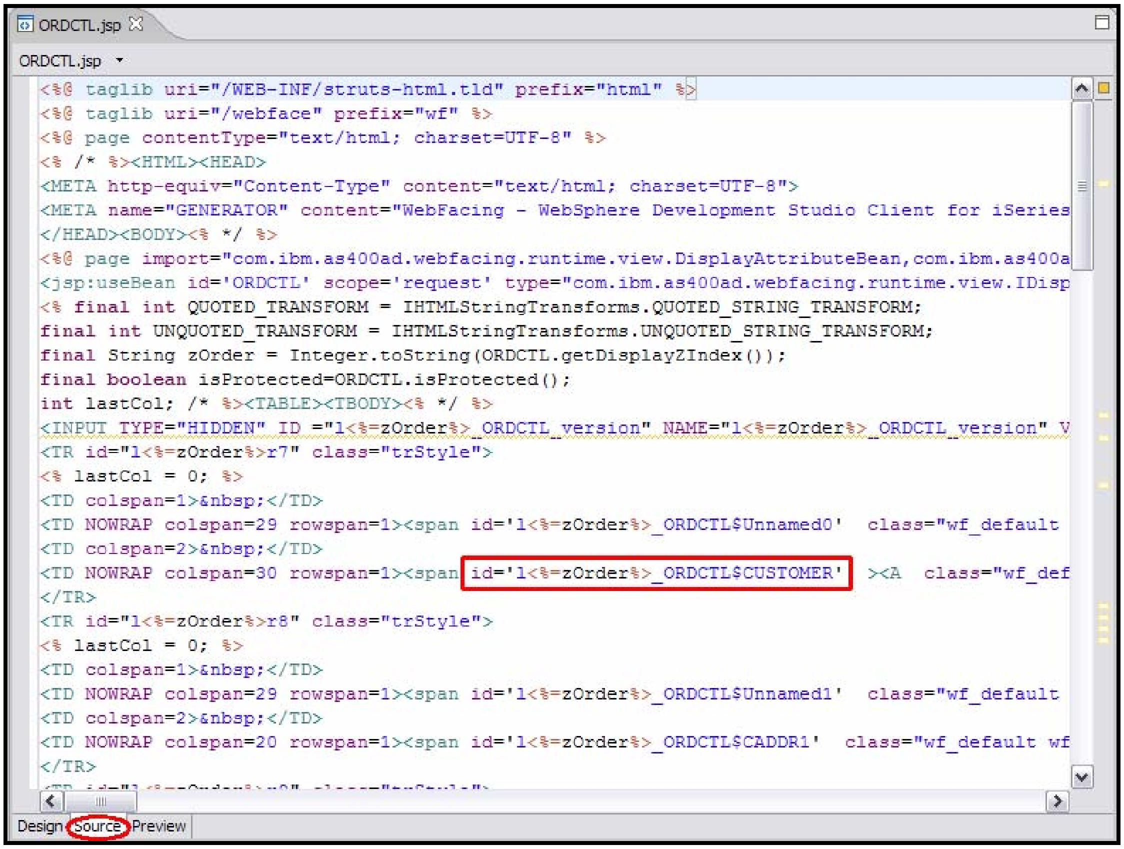


Figure 380. Select record JSP to edit using Page Designer.

Understanding the fields in the JSP source

To look at the JSP source you must: do the following:

1. Make sure that the Page Designer tool is already open with the source tab selected. Find an example of a named field with an ID equal to: `id='l<%=zOrder%>_ORDCTL$CUSTOMER'` in the JSP source. Note that the Design and preview tabs for this JSP are not very usable. This is because the code in the JSP is quite complex and is not meant to be displayed on its own. The JSP is included at run time by PageBuilder.jsp to make up the Web page, which can consist of several records formats just like your 5250 screens (see Figure 381).



```
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/webface" prefix="wf" %>
<%@ page contentType="text/html; charset=UTF-8" %>
<% /* %><HTML><HEAD>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
<META name="GENERATOR" content="WebFacing - WebSphere Development Studio Client for iSeries
</HEAD><BODY><% */ %>
<%@ page import="com.ibm.as400ad.webfacing.runtime.view.DisplayAttributeBean,com.ibm.as400a
<jsp:useBean id='ORDCTL' scope='request' type="com.ibm.as400ad.webfacing.runtime.view.IDisp
<% final int QUOTED_TRANSFORM = IHTMLStringTransforms.QUOTED_STRING_TRANSFORM;
final int UNQUOTED_TRANSFORM = IHTMLStringTransforms.UNQUOTED_STRING_TRANSFORM;
final String zOrder = Integer.toString(ORDCTL.getDisplayZIndex());
final boolean isProtected=ORDCTL.isProtected();
int lastCol; /* %><TABLE><TBODY><% */ %>
<INPUT TYPE="HIDDEN" ID ="l<%=zOrder%>_ORDCTL_version" NAME="l<%=zOrder%>_ORDCTL_version" V
<TR id="l<%=zOrder%>r7" class="trStyle">
<% lastCol = 0; %>
<TD colspan=1>&nbsp;</TD>
<TD NOWRAP colspan=29 rowspan=1><span id='l<%=zOrder%>_ORDCTL$Unnamed0'  class="wf_default
<TD colspan=2>&nbsp;</TD>
<TD NOWRAP colspan=30 rowspan=1><span id='l<%=zOrder%>_ORDCTL$CUSTOMER'  ><A  class="wf_def
</TR>
<TR id="l<%=zOrder%>r8" class="trStyle">
<% lastCol = 0; %>
<TD colspan=1>&nbsp;</TD>
<TD NOWRAP colspan=29 rowspan=1><span id='l<%=zOrder%>_ORDCTL$Unnamed1'  class="wf_default
<TD colspan=2>&nbsp;</TD>
<TD NOWRAP colspan=20 rowspan=1><span id='l<%=zOrder%>_ORDCTL$CADDR1'  class="wf_default wf
</TR>
```

Figure 381. Fields in a JSP source

2. Look through the JSP source to see HTML tags such as `<span>`, Java code also called scriptlets surrounded by `<% %>`, and JavaScript function calls like this:

   ```
   onClick="<wf:js function="setCursor"/>(7,
   2,null,'<%=(String)session.getAttribute("UniqueId")%>');"
   ```

   The above is an example of how one of the WebFacing JavaScript functions is called. The setCursor JavaScript function is placed on the onClick HTML attribute to set the cursor on a field when the field is selected. A custom tag is used to call the JavaScript function and pass a parameter to it that includes some Java code. This is a WebFacing custom tag that prefixes `c_i_e_i_w_versionNumber` (for example, `c_i_e_i_w_6012`) to the setCursor JavaScript function name. This allows the JavaScript function to have a unique name, which is required to run more than one WebFacing portlet in the WebSphere Portal environment. Normally, simple JavaScript is surrounded by `<SCRIPT language='JavaScript1.3'> </SCRIPT>` tags. You do not need to understand every detail in the JSP, just learn to recognize the different types of tags and find out where the fields are. In the next three steps, you try to find the source in the JSP for different types of fields and compare it with the DDS source.

3. This is an example of an ORDCTL record format text constant:

   ```
   JSP source: <span id='l<%=zOrder%>_ORDCTL$Unnamed0'
   DDS source: A              7  2'Customer name . . . . . . . :'
   ```

   `<%=zOrder%>` is a variable used at run time to identify the layer number that your record format is in. Because you do not know how the System i program writes this record format, the layer number can be different when this record format is used on different screens. The layer number is followed by `_recordformatname$` and then the text constant is called Unnamed0, because text constants are not named fields in the DDS.

4. This is an example of an ORDSFL subfile output field in the ORDCTL JSP. This JSP includes fields and data from the subfile control record ORDCTL, but also includes fields and data from the subfile record ORDSFL and names them with the ORDCTL name rather than the ORDSFL name. This is because only one JSP is generated on conversion for a subfile control record and its corresponding subfile record.

   ```
   JSP source: <span…id='l<%=zOrder%>_ORDCTL$PARTNBR__O$<%=rrn%>'
   DDS source: A    PARTNBR_O R    O 14  6REFFLD(IID *LIBL/ITEM)
   ```

   The `<span` HTML tag starts the output field. It is followed by some attributes that you do not need to worry about (they are replaced with …), but if you find the ID,notice that it has a similar name to the original DDS field name, and it has an extra underscore character. The field name PARTNBR_O is converted to PARTNBR__O. This is an example of how special characters are treated when converting from the DDS field name to the HTML ID. Upper case A-Z and 0-9: has no transform

   - '_' becomes "__"
   - '@' becomes "_x"
   - '#' becomes "_y"
   - '$' becomes "_z"

   Other special characters valid in DDS field, record, or file names become something like "_" concatenated with the four-digit alphabetic characters, which represent the hexadecimal value of that special character.

   `<%=rrn%>` is a variable that identifies the subfile relative record number for each subfile row.

5. This is an example of an input-capable field from the ORDSFL subfile record format that you are able to find in the ORDCTL.jsp source:

   ```
   JSP source: <INPUT…id="l<%=zOrder%>_ORDCTL$OPT$<%=rrn%>">
   DDS source: A            OPT            1Y 0B 14  3EDTCDE(Z)
   ```

   Here, the `<INPUT` HTML tag starts the input capable field.

Exercise 19.2: Change the color of a subfile row example using Page Designer

In this example, you see how direct edits to a JSP can be used to try new code in your record formats. You add JavaScript to a JSP so that subfile rows change color when the mouse moves over them.

Adding JavaScript to the JSP source

To edit the JSP source, perform the steps below. (**Note:** the example JavaScript used in the example provided might not work with all PCs and Browsers and might fail.)

1. Go to the Page Designer tool that has the ORDCTL.jsp file you were working with in the previous exercise.
2. Find the OPT subfile input-capable field in the ORDCTL.jsp source. It looks something similar to this:

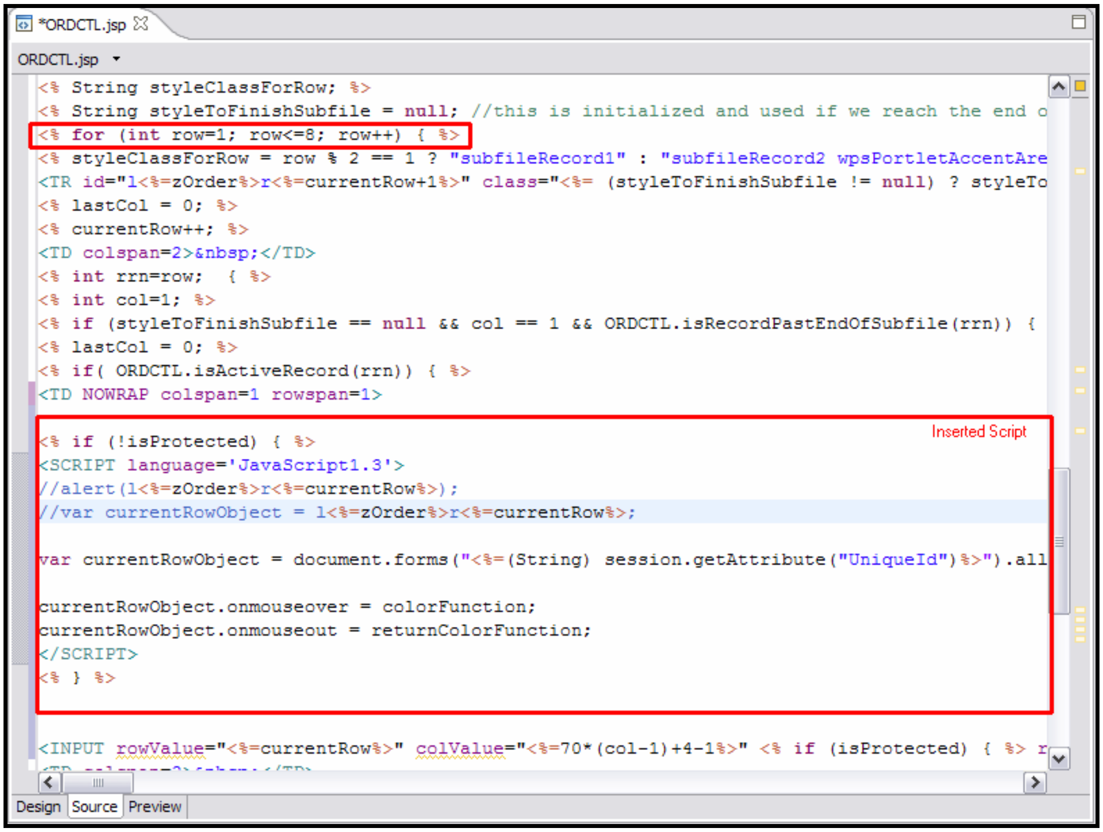
   ```
   <INPUT rowValue="<%=currentRow%>" colValue="<%=70*(col-1)+4-1%>" <%
   if (isProtected) { %> readonly tabindex=-1 <% } %>
   id="l<%=zOrder%>_ORDCTL$OPT$<%=rrn%>"
   ```

3. Place your cursor just before the HTML `<INPUT` tag and copy and paste in the following JavaScript code:

   ```
   <% if (!isProtected) { %>
   <SCRIPT language='JavaScript1.3'>
   //alert(l<%=zOrder%>r<%=currentRow%>);
   //var currentRowObject = l<%=zOrder%>r<%=currentRow%>;

   var currentRowObject = document.forms("<%=(String)
   session.getAttribute("UniqueId")%>").all("l<%=zOrder%>r<%=currentRow%>"
   );

   currentRowObject.onmouseover = colorFunction;
   currentRowObject.onmouseout = returnColorFunction;
   </SCRIPT>
   <% } %>
   ```

   The first and last line allow the JavaScript code to be included when the subfile is in an active layer. For example, when you prompt and a Window record appears, the subfile is not active. The second and second to last line start and end your JavaScript code. The third line is an alert statement that is commented out with //. If you remove the comment tags, you can use this type of alert statement to debug your JavaScript. The next line shows you how you set the variable `currentRowObject` to select the row in the subfile that you want to change the color of. That line is commented out to show you what you would normally do to get the row object. However, to enable this application for both Web and Portal environments, you use the currentRowObject as defined in the fifth line. In the portal environment, if you have two WebFacing portlets running, there might be more than one row with the same ID. To reference the correct object, the unique form ID is used to retrieve the ID within that form. This way, it work for both Portal and Web environments. The next two lines are calls to the two *functions* that you define in the next section “Adding a user-defined JavaScript file to your project.”

Notice that the Java variables <%=zOrder%> and <%=currentRow%> are used to build the above JavaScript. The Java *for* loop in the subfile section of the JSP creates a different copy of the JavaScript code for each subfile row (see Figure 382).



```
<% String styleClassForRow; %>
<% String styleToFinishSubfile = null; //this is initialized and used if we reach the end o
<% for (int row=1; row<=8; row++) { %>
<% styleClassForRow = row % 2 == 1 ? "subfileRecord1" : "subfileRecord2 wpsPortletAccentAre
<TR id="l<%=zOrder%>r<%=currentRow+1%>" class="<%= (styleToFinishSubfile != null) ? styleTo
<% lastCol = 0; %>
<% currentRow++; %>
<TD colspan=2>&nbsp;</TD>
<% int rrn=row;  { %>
<% int col=1; %>
<% if (styleToFinishSubfile == null && col == 1 && ORDCTL.isRecordPastEndOfSubfile(rrn)) {
<% lastCol = 0; %>
<% if( ORDCTL.isActiveRecord(rrn)) { %>
<TD NOWRAP colspan=1 rowspan=1>

<% if (!isProtected) { %>
<SCRIPT language='JavaScript1.3'>
//alert(l<%=zOrder%>r<%=currentRow%>);
//var currentRowObject = l<%=zOrder%>r<%=currentRow%>;

var currentRowObject = document.forms("<%=(String) session.getAttribute("UniqueId")%>").all

currentRowObject.onmouseover = colorFunction;
currentRowObject.onmouseout = returnColorFunction;
</SCRIPT>
<% } %>


<INPUT rowValue="<%=currentRow%>" colValue="<%=70*(col-1)+4-1%>" <% if (isProtected) { %> r
```

Figure 382. JavaScript code

To view the compiled output and see your resulting JavaScript, you need to run the application on the Web. You can do this when running the application in one of the following sections of this example and viewing the HTML results when the ORDCTL record format is displayed on the Browser.

Adding a User-defined JavaScript file to your project

To edit the JSP source:

1. Click the **Navigator** tab to switch to the Navigator view.

2. Locate the **usr** folder in the WebFacing project. It is located in the following directory hierarchy: wflabxx\WebContent\webfacing\ClientScript\ (see Figure 383).

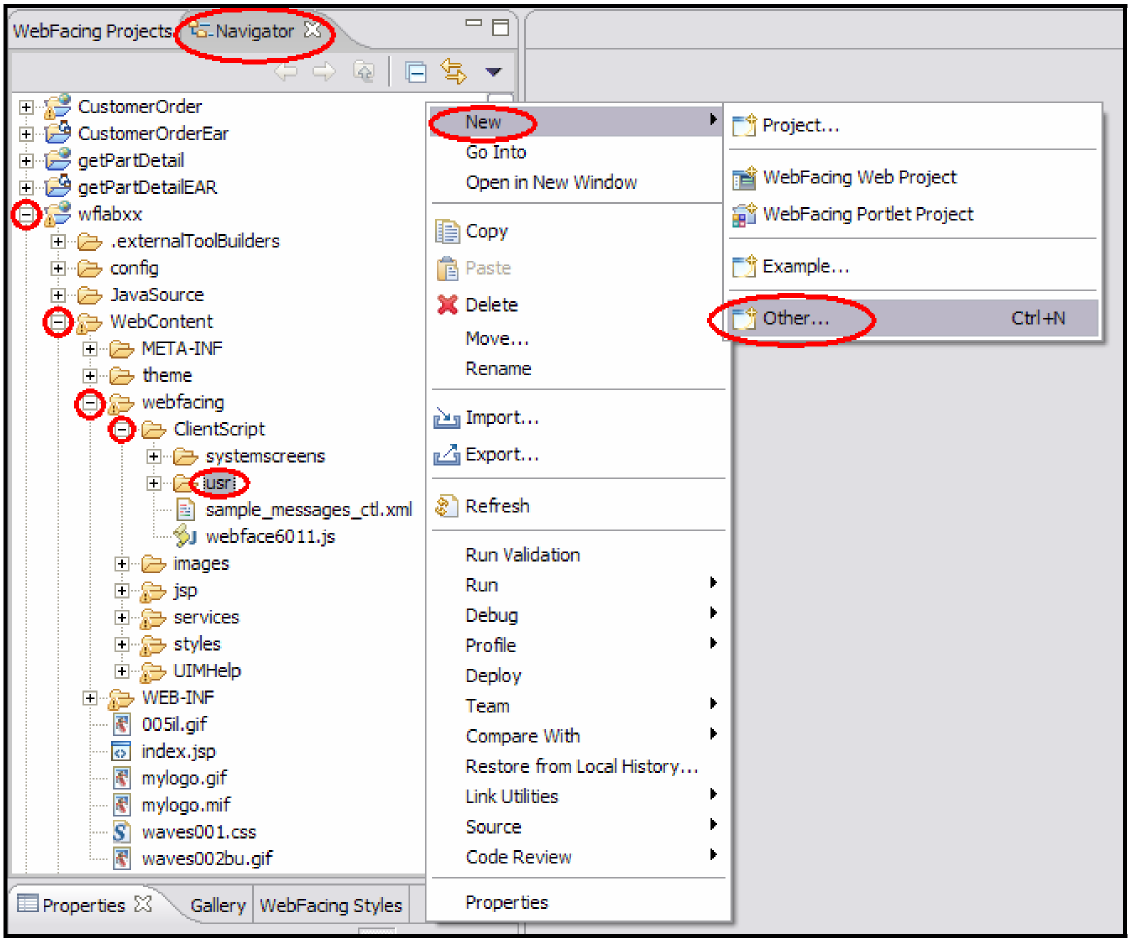
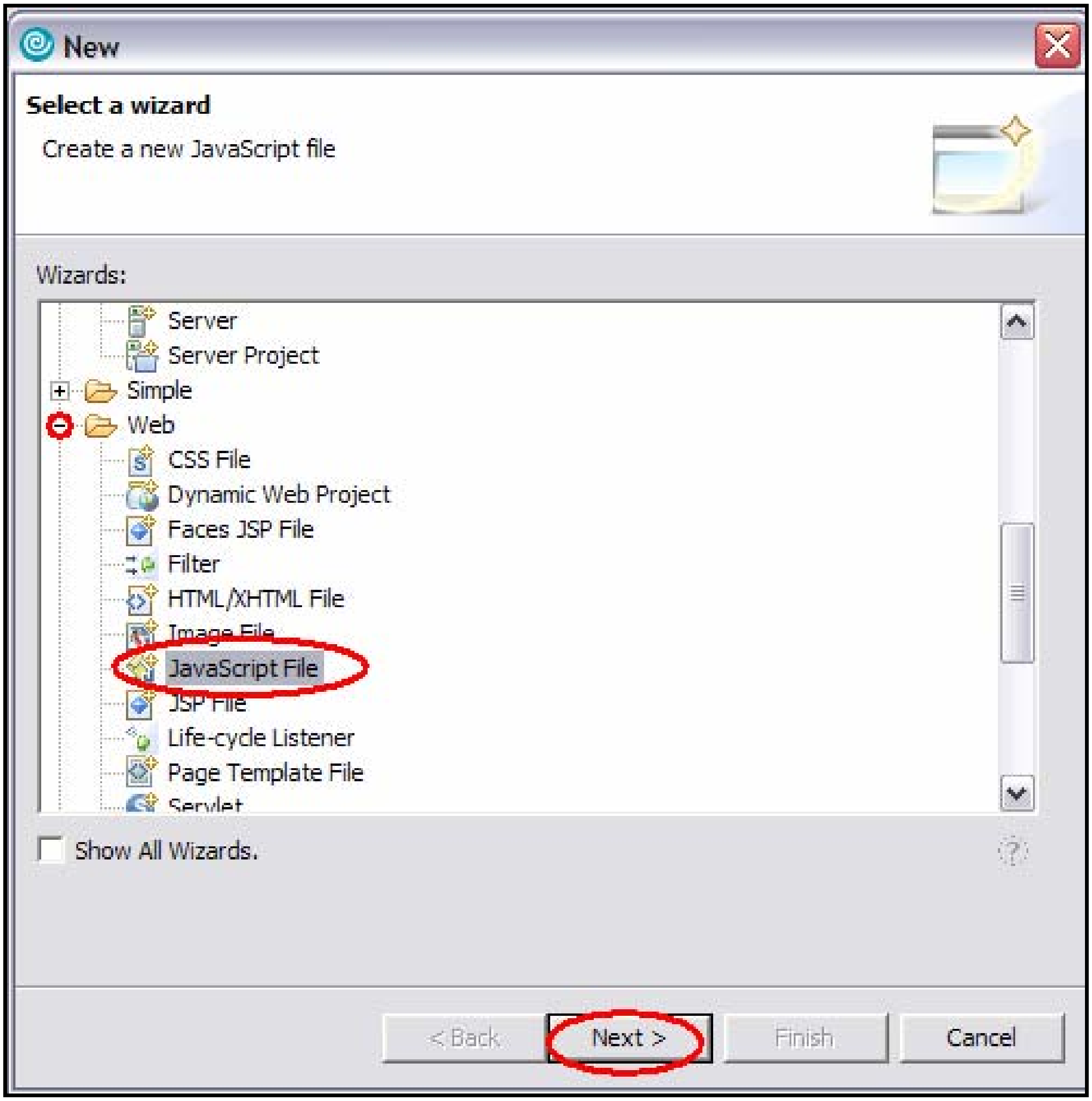


Figure 383. Editing the JSP source

3. Right-click the **usr** folder icon in the Navigator view and click **New > Other** on the pop-up menu.

4. The **New** dialog appears (see Figure 384).



New

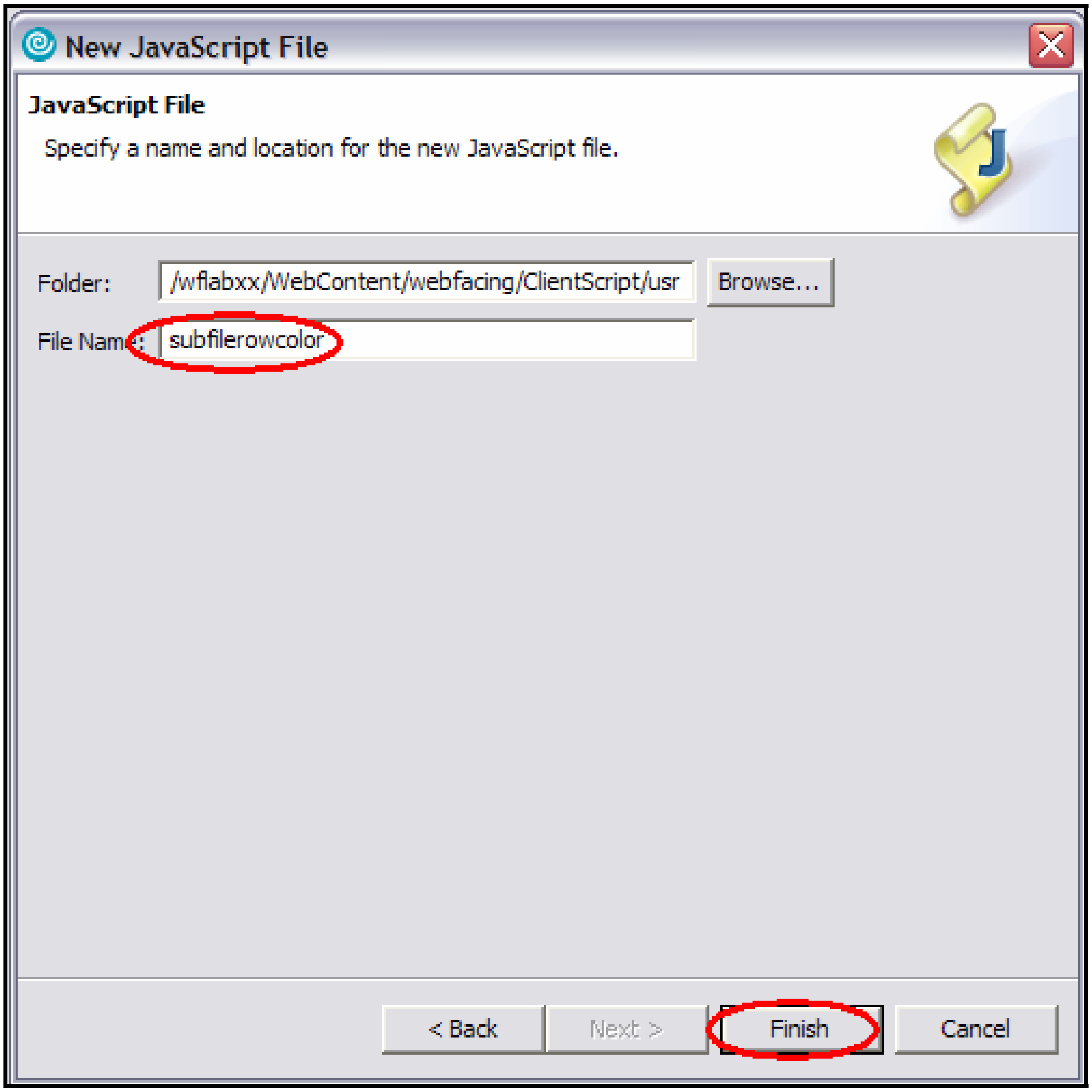
Select a wizard

Create a new JavaScript file

Figure 384. The New dialog

5. To create a new JavaScript file, click **Web** and then **JavaScript File** in the **New** dialog and click **Next**.

6. Enter a name for your new JavaScript file. It does not matter what name you use, but for example, you can use subfilerowcolor as the name. Click **Finish** (see Figure 385).



New JavaScript File

JavaScript File

Specify a name and location for the new JavaScript file.

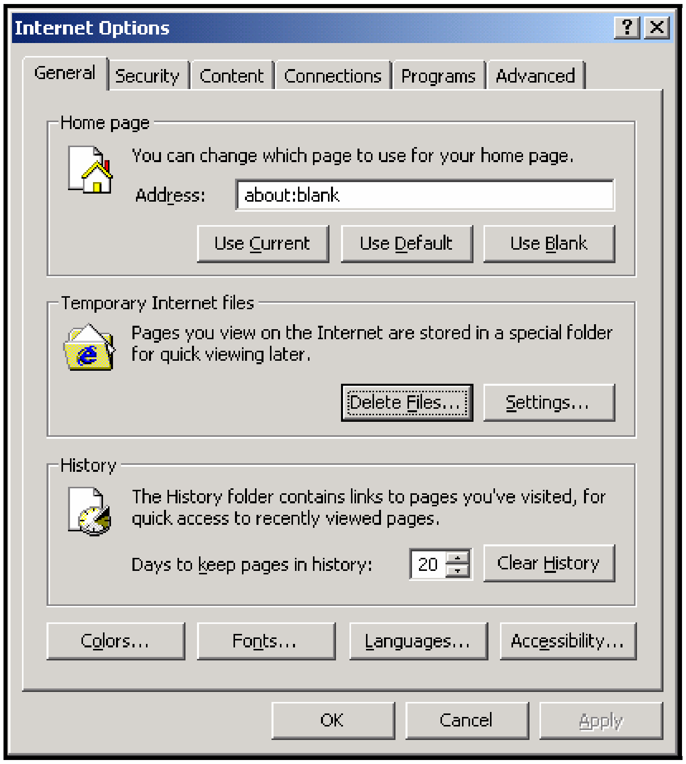
Folder: /wflabxx/WebContent/webfacing/ClientScript/usr

File Name: subfilerowcolor

*Figure 385. Enter a name for your new JavaScript file and click **Finish**.*

A JavaScript file is created with the extension .js in the usr folder. This file is automatically included when you run your WebFacing project.

Note: You need to restart the application server or the application when you add a new JavaScript file. In addition, if you modify the user-defined JavaScript file, you might need to clear the files saved in your browser to get the changes to take effect. To do this in your Web browser, select **File** > **Tools** > **Internet Options…** and in the **General** tab under **Temporary Internet files** select **Delete Files** (see Figure 386).



Internet Options

Home page

You can change which page to use for your home page.

Address: about:blank

Temporary Internet files

Pages you view on the Internet are stored in a special folder for quick viewing later.

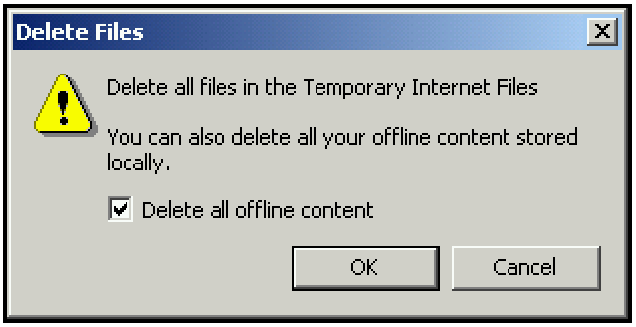
History

The History folder contains links to pages you've visited, for quick access to recently viewed pages.

Days to keep pages in history: 20

Figure 386. Clear the files saved in your browser to get the changes to take effect.

7. Now select **OK** in the next dialog that appears (see Figure 387).



Delete Files

Delete all files in the Temporary Internet Files

You can also delete all your offline content stored locally.

☑ Delete all offline content

*Figure 387. Select **OK** to delete all offline content.*

8. In the editor that appears for the subfilerowcolor.js file, enter the following JavaScript code. This defines the JavaScript functions that you are calling in the ORDCTL.jsp.

```
// subfilerowcolor.js

// newFunction
var line     = null;

function colorFunction(){
//      alert(window.event.srcElement.parentElement.tagName);
      if (window.event.srcElement.parentElement.tagName == "TR"){
         line = window.event.srcElement.parentElement;
      }
      line.oldStyle = line.style.backgroundColor;
      line.style.backgroundColor = '#9999ff';
}
function returnColorFunction(){
      if (window.event.srcElement.parentElement.tagName == "TR"){
         line = window.event.srcElement.parentElement;
      }
      if (line.oldStyle != null){
         line.style.backgroundColor = line.oldStyle;
      }
}
```

The first function `colorFunction` is called when a mouse over event is triggered on a particular subfile row. This changes the color of the subfile row to '#9999ff' (which is blue) when the mouse is placed over the row.

The second function `returnColorFunction` is called when a mouse out event is triggered on a particular subfile row. This function uses the JavaScript variables `line` and `oldStyle` to change the row color back to its original color when the mouse is moved away from the row.

You might wonder at this point, why you do not just put these color changes on the table row tag directly. This is because you do not have access to the output of the table row on conversion of your DDS record formats. So, to save the information for future conversion, all modifications must be done in Web Settings and Web Settings modifications can only be done on fields or field tags like the <span> and <input> tags or before and after fields. This is why you manipulate the TR attributes indirectly using JavaScript.

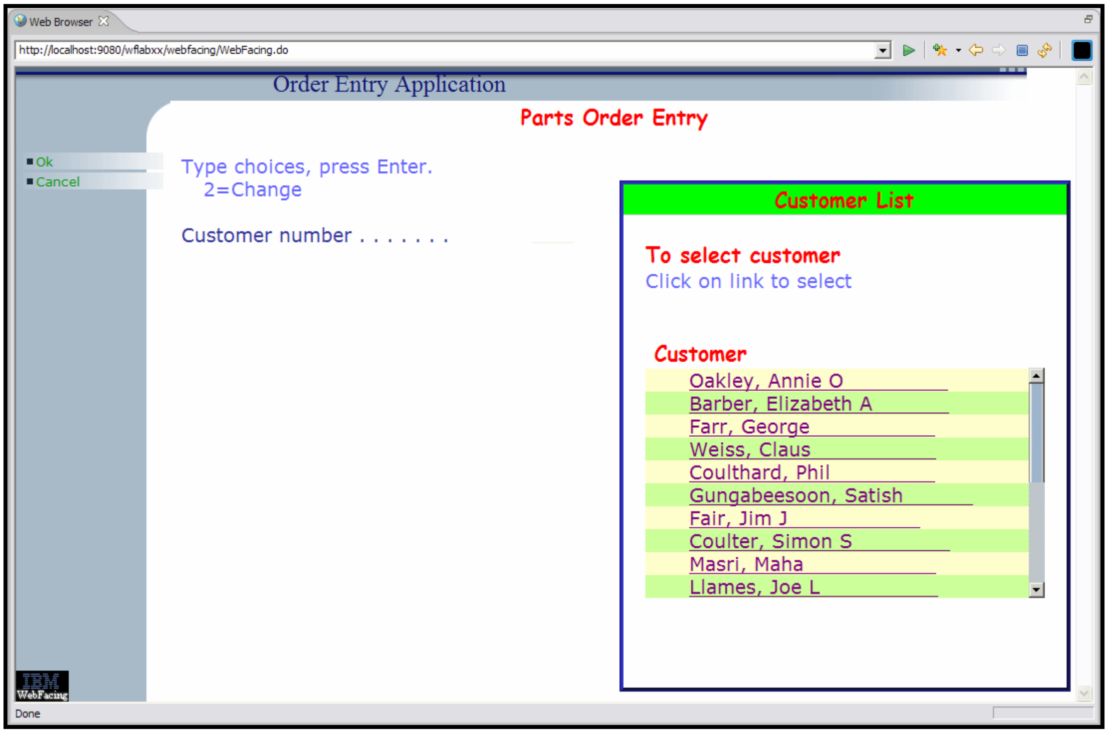
Running the WebFacing application for this subfile example

Follow these steps to view the subfile that you have been working with in this example:

1. Right-click the wflabxx project icon in the WebFacing Projects view.
2. Click **Run on Server** on the pop-up menu.
3. Go to the browser pane and click the **Launch** link.

   You see the first screen of your application. Nothing has changed there.

4. Click the **Prompt** push button, or press command key **F4** (see Figure 388).



Order Entry Application

Parts Order Entry

Type choices, press Enter.
  2=Change

Customer number . . . .

Customer List

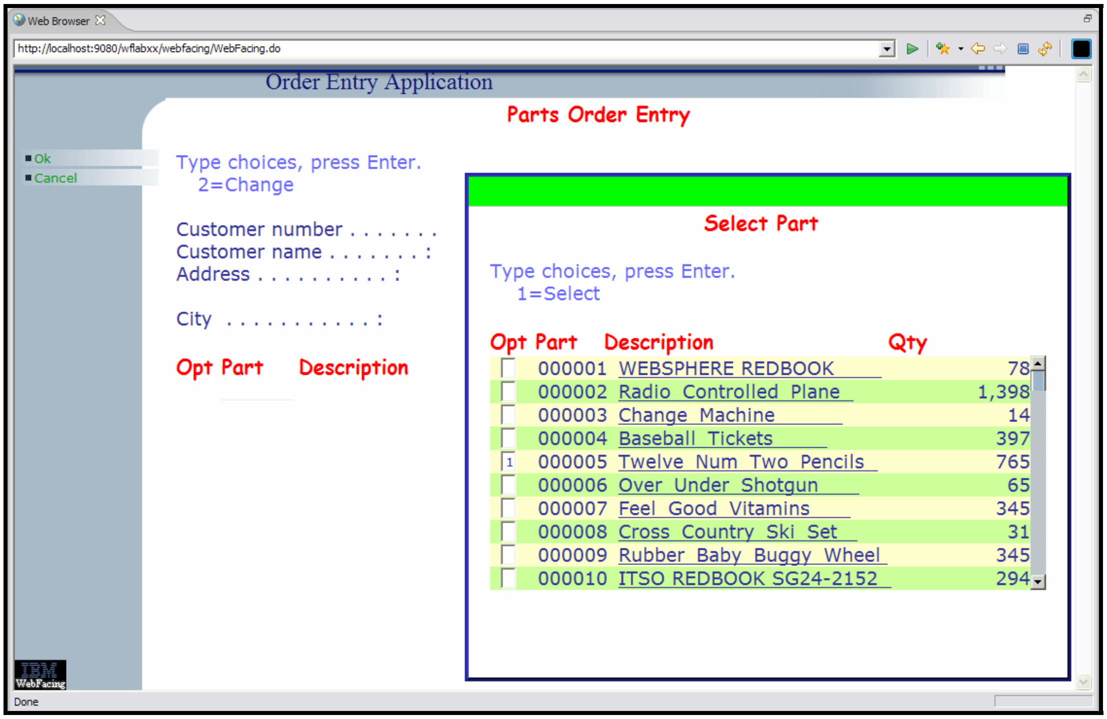
To select customer
Click on link to select

Customer
Oakley, Annie O
Barber, Elizabeth A
Farr, George
Weiss, Claus
Coulthard, Phil
Gungabeesoon, Satish
Fair, Jim J
Coulter, Simon S
Masri, Maha
Llames, Joe L

Figure 388. First screen of Order Entry Application (showing Customer List).

5. Select a customer from the list by clicking the link.
6. Press command key **F4** to see a list of available parts.

7. The Select Part panel opens (see Figure 389).



Order Entry Application

Parts Order Entry

Type choices, press Enter.
2=Change

Customer number . . . .
Customer name . . . . :
Address . . . . :

City . . . . :

Opt Part Description

Select Part

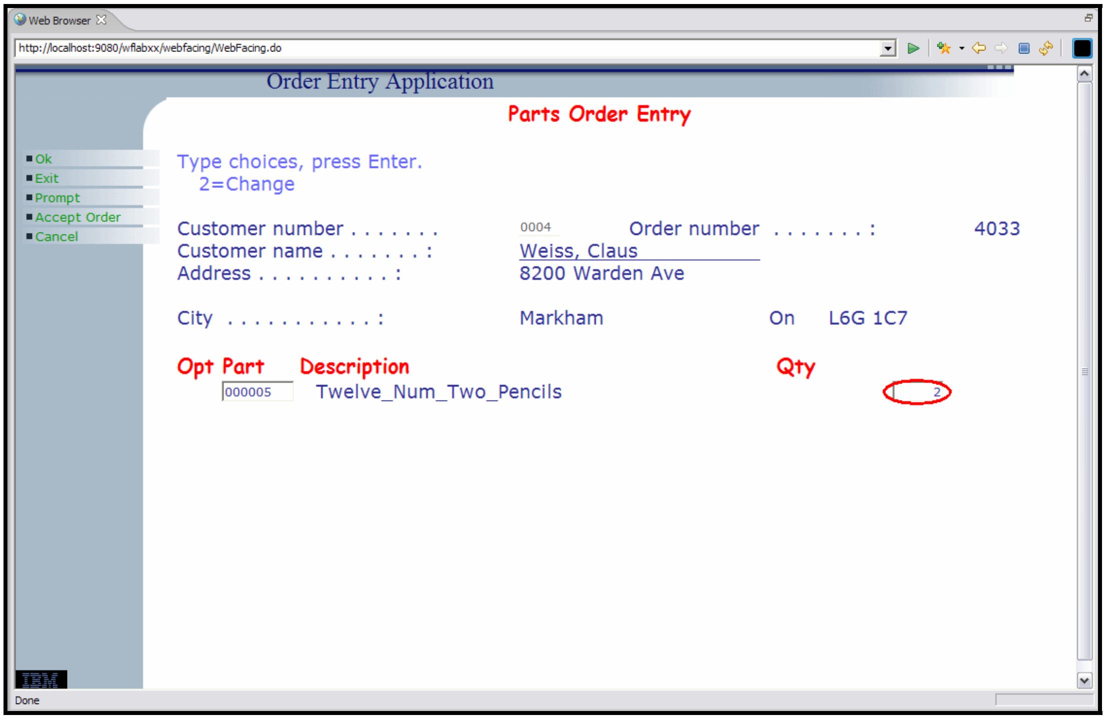
Type choices, press Enter.
1=Select

Opt	Part	Description	Qty
□	000001	WEBSPHERE REDBOOK	78
□	000002	Radio Controlled Plane	1,398
□	000003	Change Machine	14
□	000004	Baseball Tickets	397
1	000005	Twelve Num Two Pencils	765
□	000006	Over Under Shotgun	65
□	000007	Feel Good Vitamins	345
□	000008	Cross Country Ski Set	31
□	000009	Rubber Baby Buggy Wheel	345
□	000010	ITSO REDBOOK SG24-2152	294

Figure 389. The Select Part panel

8. In the **Options** field, type 1 next to a part that has lots of quantity left and press **Enter**.

9. Back on the order screen, the part appears in the order on the top line of the parts list (see Figure 390).



Order Entry Application

Parts Order Entry

Type choices, press Enter.
2=Change

Customer number . . . . 0004 Order number . . . . : 4033
Customer name . . . . : Weiss, Claus
Address . . . . : 8200 Warden Ave

City . . . . : Markham On L6G 1C7

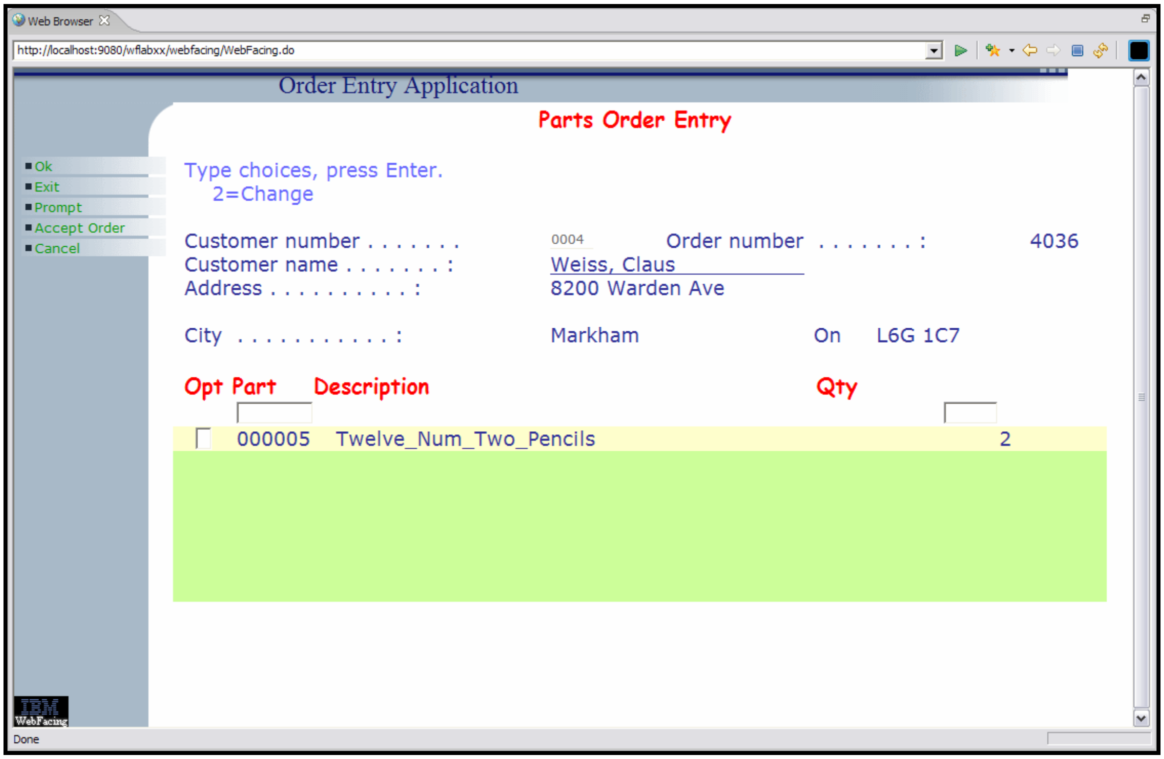
Opt	Part	Description	Qty
	000005	Twelve_Num_Two_Pencils	2

Figure 390. The part appears on the order screen.

Next, you order a certain quantity of this part.

10. Change the **Qty** field to a value, for example, **2**.
11. Press the **Enter** key to add the part and quantity ordered to the order.

The detail order line for this part, with the quantity specified, is now part of this order (see Figure 391).



Parts Order Entry

Type choices, press Enter.
2=Change

Customer number . . . . 0004 Order number . . . . : 4036
Customer name . . . . : Weiss, Claus
Address . . . . : 8200 Warden Ave

City . . . . : Markham On L6G 1C7

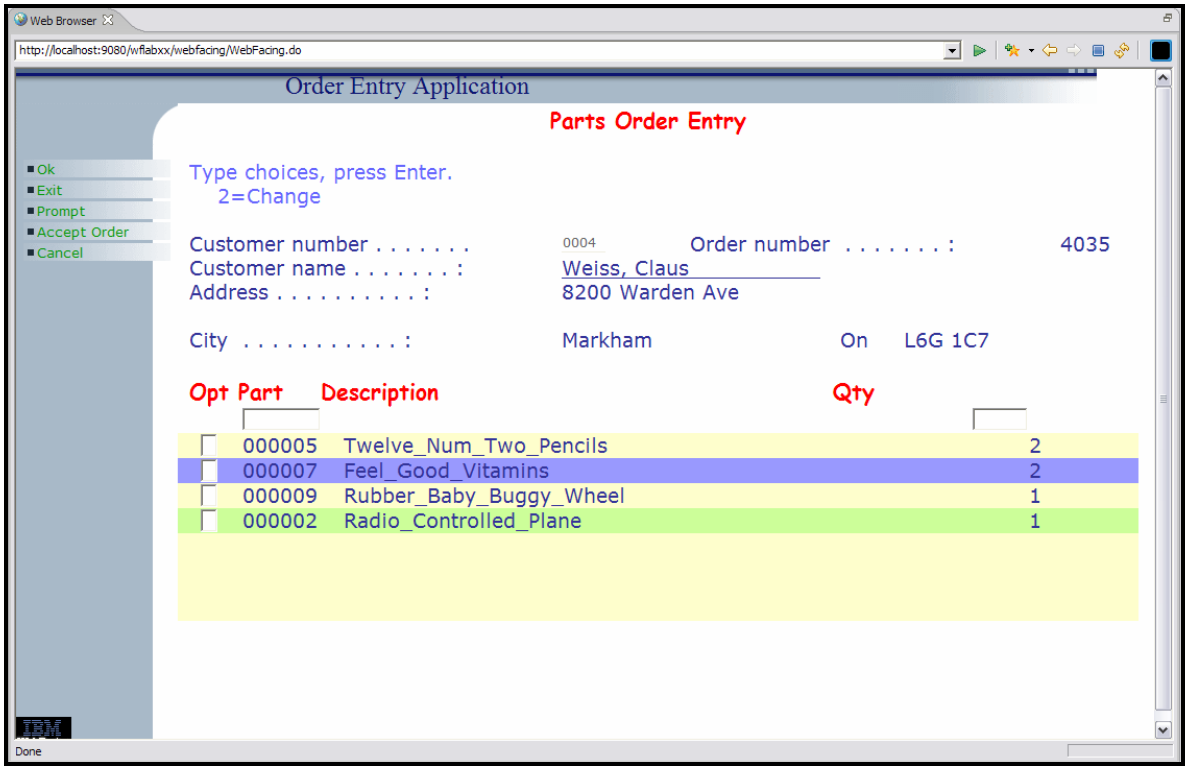
Opt	Part	Description	Qty
□	000005	Twelve_Num_Two_Pencils	2

Figure 391. The detail order line for the part

Add more parts until your order is complete.

12. Press command key **F4** to see a list of available parts.
13. Select a part from the parts list and press **Enter**.
14. Specify the part quantity and press **Enter**.
15. Now move the mouse over one of the parts in your order.

You see that each row is highlighted when the mouse passes over it and returns to its original color when the mouse leaves the row (see Figure 392).



Order Entry Application

Parts Order Entry

Type choices, press Enter.
2=Change

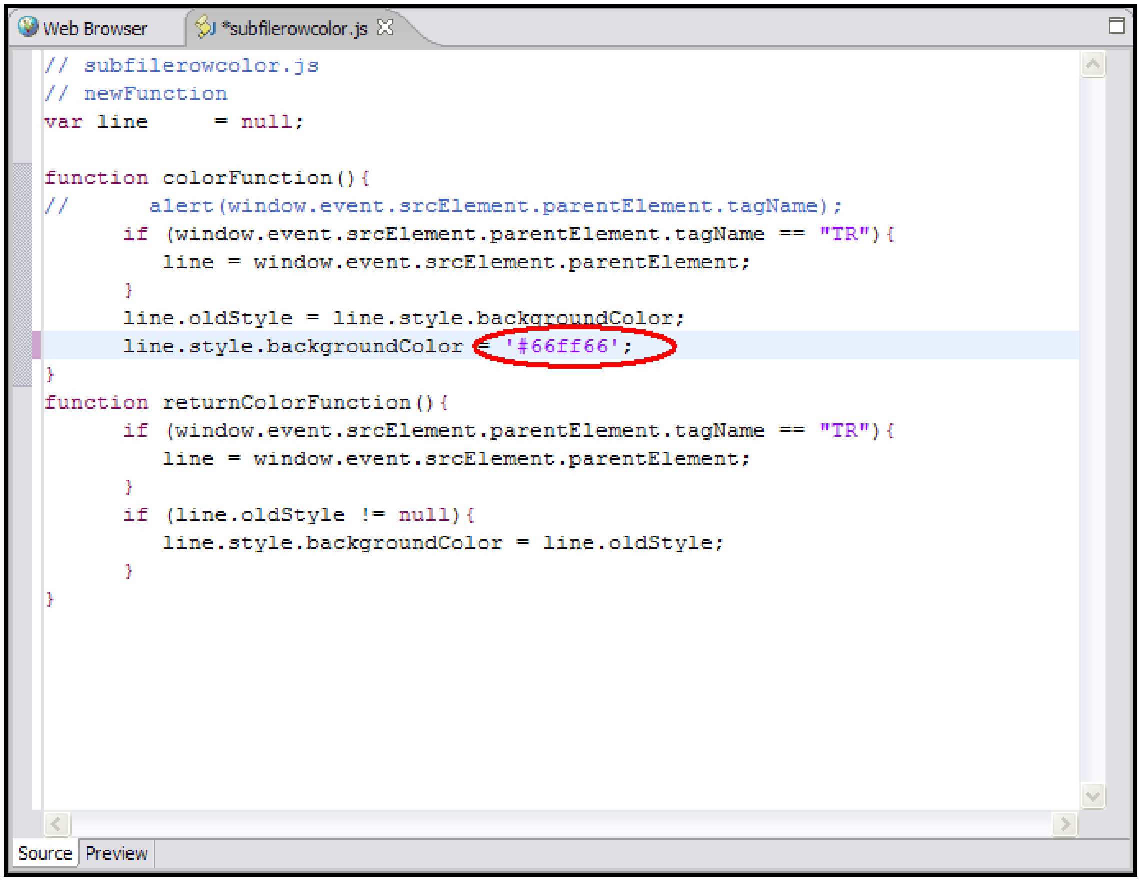
Customer number . . . . 0004 Order number . . . . : 4035
Customer name . . . . : Weiss, Claus
Address . . . . : 8200 Warden Ave

City . . . . : Markham On L6G 1C7

Opt	Part	Description	Qty
□	000005	Twelve_Num_Two_Pencils	2
□	000007	Feel_Good_Vitamins	2
□	000009	Rubber_Baby_Buggy_Wheel	1
□	000002	Radio_Controlled_Plane	1

Figure 392. Detail parts list for order

16. Go back to the JavaScript editor of the subfilerowcolor.js file (see Figure 393).



```
// subfilerowcolor.js
// newFunction
var line     = null;

function colorFunction(){
//         alert(window.event.srcElement.parentElement.tagName);
        if (window.event.srcElement.parentElement.tagName == "TR"){
            line = window.event.srcElement.parentElement;
        }
        line.oldStyle = line.style.backgroundColor;
        line.style.backgroundColor = '#66ff66';
}
function returnColorFunction(){
        if (window.event.srcElement.parentElement.tagName == "TR"){
            line = window.event.srcElement.parentElement;
        }
        if (line.oldStyle != null){
            line.style.backgroundColor = line.oldStyle;
        }
}
```

Figure 393. JavaScript editor

17. Change the hex value `#9999ff` to a different hex value like `#66ff66`. You can use the Colors tab under the source page to find the color that you want (see Figure 394).

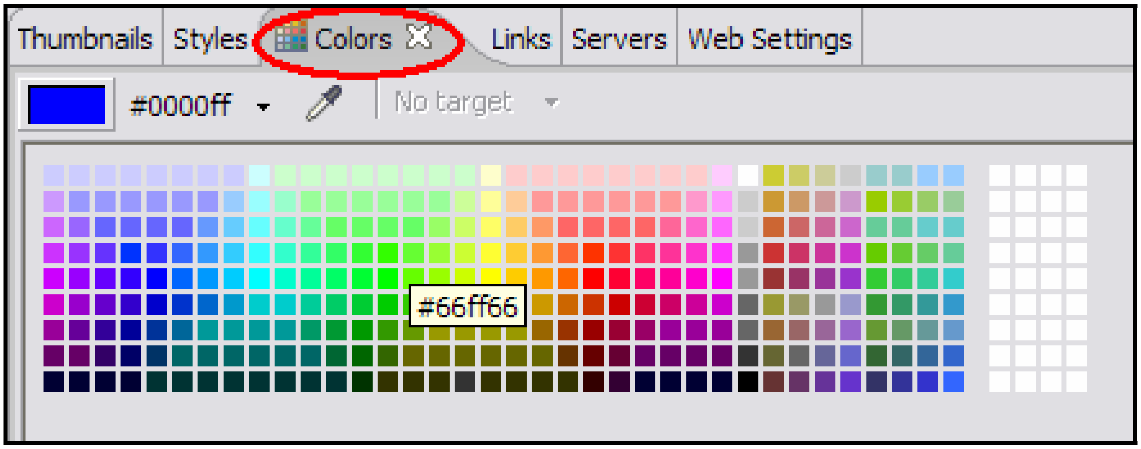
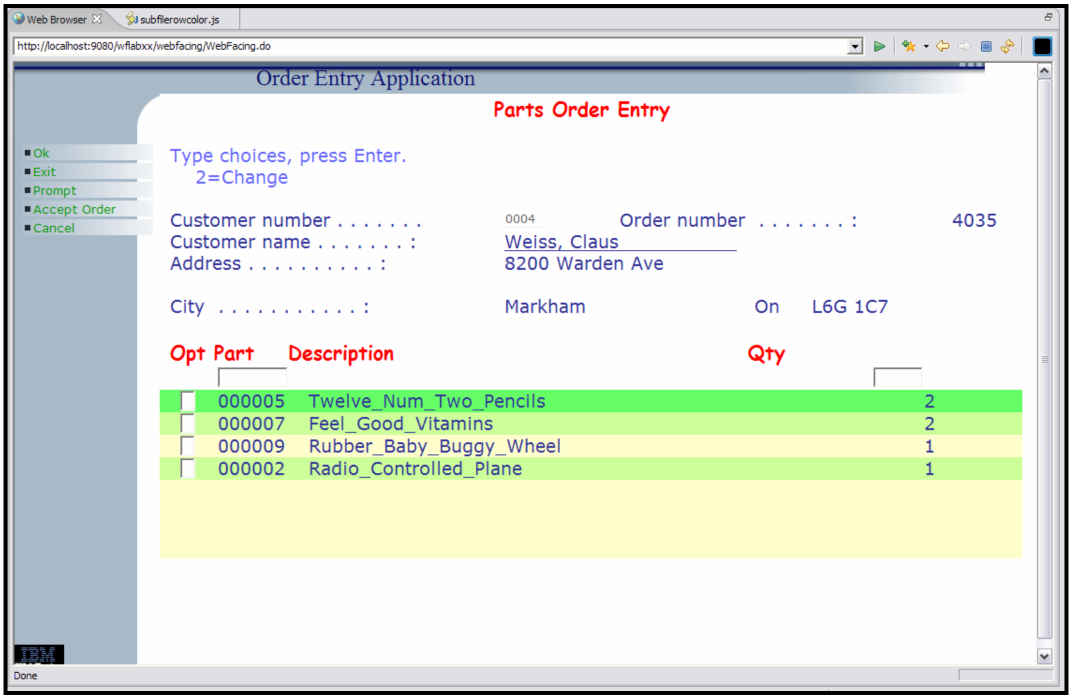


Figure 394. Choose a color

18. Go back to the browser pane and press **Enter**. You see the new color being used when the mouse goes over the subfile rows (see Figure 395).



Order Entry Application

Parts Order Entry

Type choices, press Enter.
2=Change

Customer number . . . . 0004 Order number . . . . : 4035
Customer name . . . . : Weiss, Claus
Address . . . . : 8200 Warden Ave

City . . . . : Markham On L6G 1C7

Opt	Part	Description	Qty
□	000005	Twelve_Num_Two_Pencils	2
□	000007	Feel_Good_Vitamins	2
□	000009	Rubber_Baby_Buggy_Wheel	1
□	000002	Radio_Controlled_Plane	1

Figure 395. New color is depicted on Parts Order Entry screen

Exercise 19.3: Using Web Settings to save your edits for reconversion

In this exercise, you see how you can save the same code using Web Settings that you entered by directly editing a JSP. This ensures that you do not lose your changes on reconversion.

Adding JavaScript using Web Settings in CODE Designer

To start CODE Designer (see Figure 396):

1. From the WebFacing Projects perspective, expand the **DDS** folder of the wflabxx project.
2. Right-click **ORDENTD** source member inside the DDS folder.
3. Click **Open With > CODE Designer** on the pop-up menu.

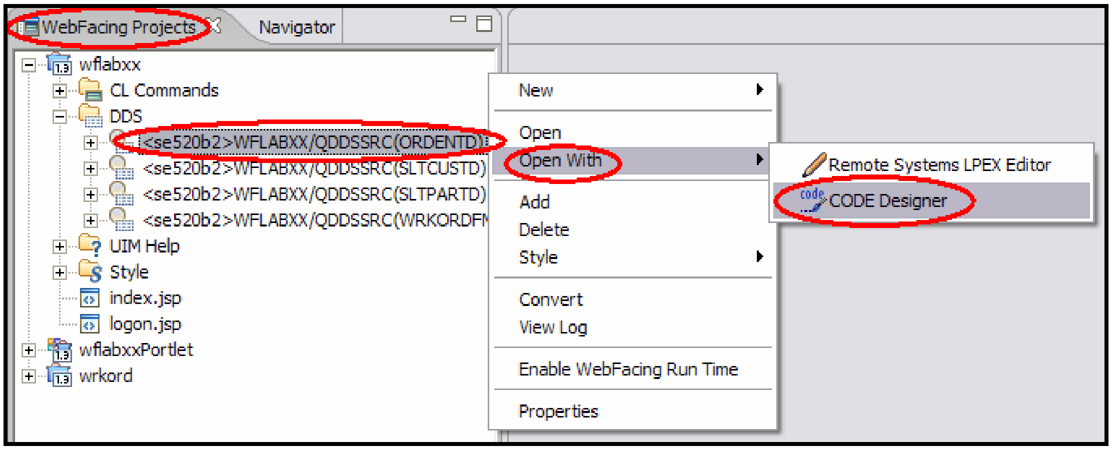
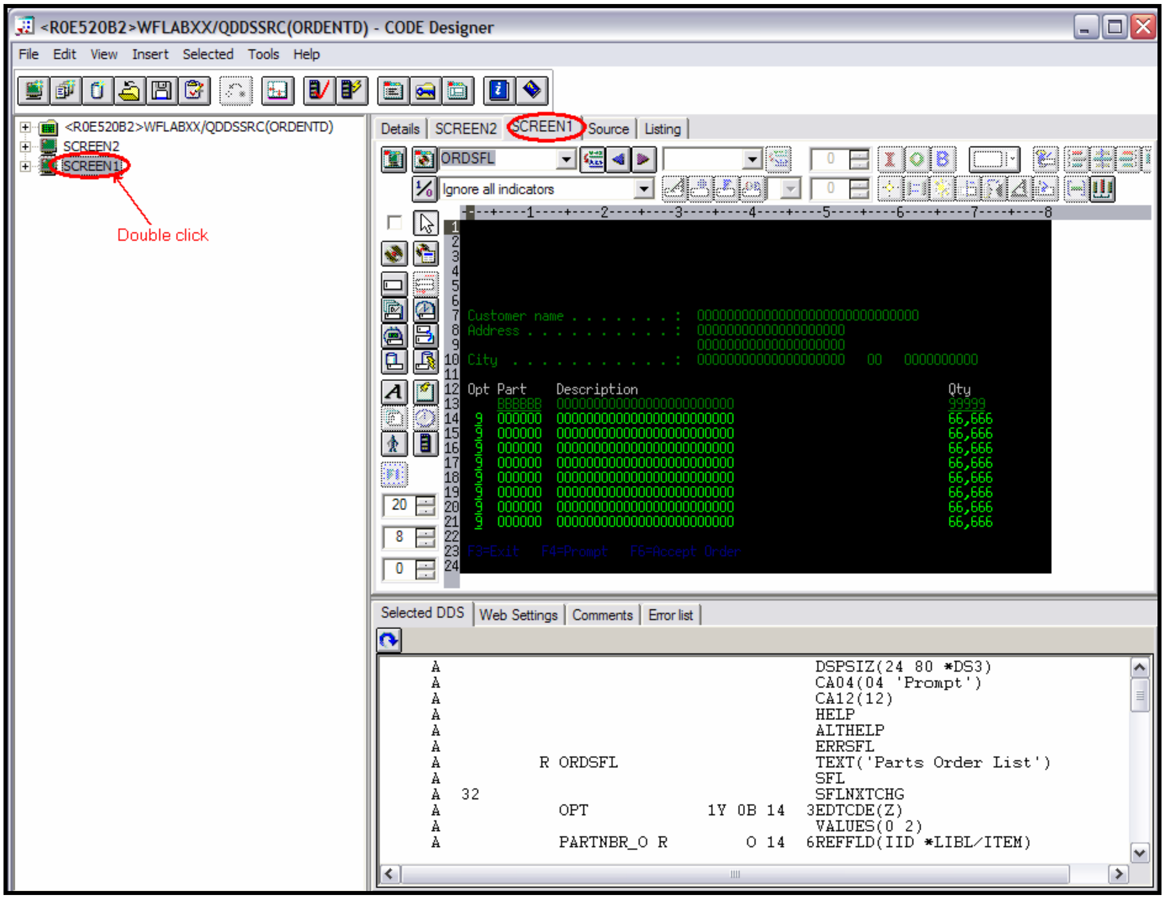

WebFacing Projects
Navigator
wflabxx
CL Commands
DDS
<se520b2>WFLABXX/QDDSSRC(ORDENTD)
<se520b2>WFLABXX/QDDSSRC(SLTCUSTD)
<se520b2>WFLABXX/QDDSSRC(SLTPARTD)
<se520b2>WFLABXX/QDDSSRC(WRKORDFI
UIM Help
Style
index.jsp
logon.jsp
wflabxxPortlet
wrkord
New
Open
Open With
Add
Delete
Style
Convert
View Log
Enable WebFacing Run Time
Properties
Remote Systems LPEX Editor
CODE Designer

Figure 396. Starting CODE Designer

CODE Designer loads the DDS member. This takes a moment.

The DDS member loaded and CODE Designer opens (see Figure 397).

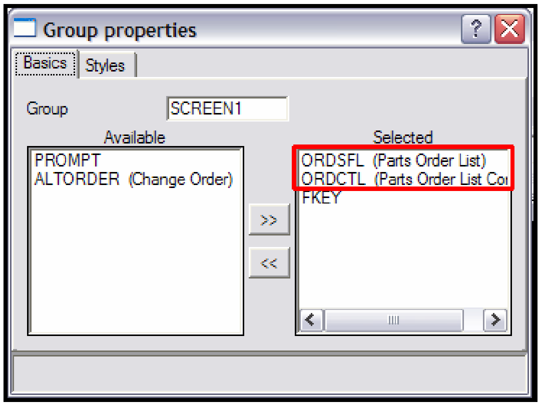


```
Customer name . . . . . . . :  000000000000000000000000000000
Address . . . . . . . . . . :  00000000000000000000
                               00000000000000000000
City  . . . . . . . . . . . :  00000000000000000000   00   0000000000

Opt Part    Description                                          Qty
    BBBBBB  000000000000000000000000                             99999
 9  000000  000000000000000000000000                             66,666
 9  000000  000000000000000000000000                             66,666
 9  000000  000000000000000000000000                             66,666
 9  000000  000000000000000000000000                             66,666
 9  000000  000000000000000000000000                             66,666
 9  000000  000000000000000000000000                             66,666
 9  000000  000000000000000000000000                             66,666
 9  000000  000000000000000000000000                             66,666

F3=Exit   F4=Prompt   F6=Accept Order
```

```
     A                                      DSPSIZ(24 80 *DS3)
     A                                      CA04(04 'Prompt')
     A                                      CA12(12)
     A                                      HELP
     A                                      ALTHELP
     A                                      ERRSFL
     A          R ORDSFL                    TEXT('Parts Order List')
     A                                      SFL
     A  32                                  SFLNXTCHG
     A            OPT            1Y 0B 14  3EDTCDE(Z)
     A                                      VALUES(0 2)
     A            PARTNBR_O R        O 14  6REFFLD(IID *LIBL/ITEM)
```

Figure 397. The DDS member loaded and CODE Designer opens.

4. Select the **SCREEN1** tab on the notebook if it is not already selected. Also, double-click the SCREEN1 icon on the left to make sure that the ORDCTL and ORDSFL record formats are selected for SCREEN1 (see **Error! Reference source not found.**).



Group SCREEN1

Available
PROMPT
ALTORDER (Change Order)

Selected
ORDSFL (Parts Order List)
ORDCTL (Parts Order List Co
FKEY

*Figure 398. Ensure that the **ORDCTL** and **ORDSFL** record formats are selected.*

5. Expand **SCREEN1** and the **ORDSFL** record format by clicking the plus sign (**+**) beside their icons. Then select the **OPT** field (see Figure 399).

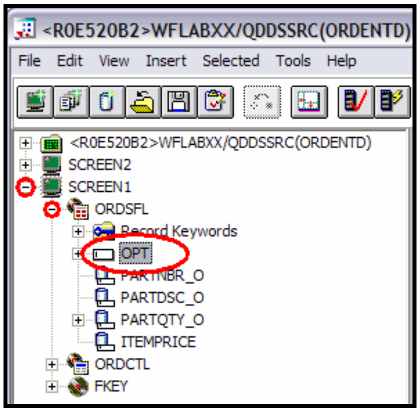

<R0E520B2>WFLABXX/QDDSSRC(ORDENTD)
File Edit View Insert Selected Tools Help
<R0E520B2>WFLABXX/QDDSSRC(ORDENTD)
SCREEN2
SCREEN1
ORDSFL
Record Keywords
OPT
PARTNBR_O
PARTDSC_O
PARTQTY_O
ITEMPRICE
ORDCTL
FKEY

Figure 399. Select the OPT field.

6. Click the **Web Settings** tab (see Figure 400).

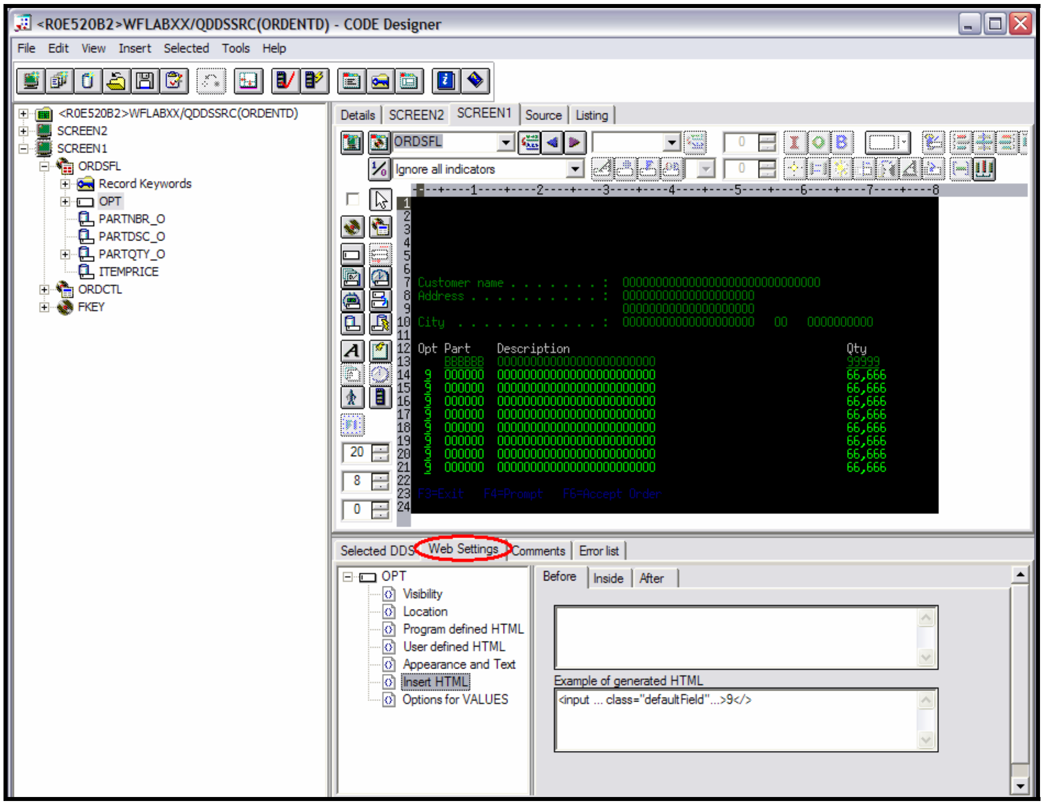
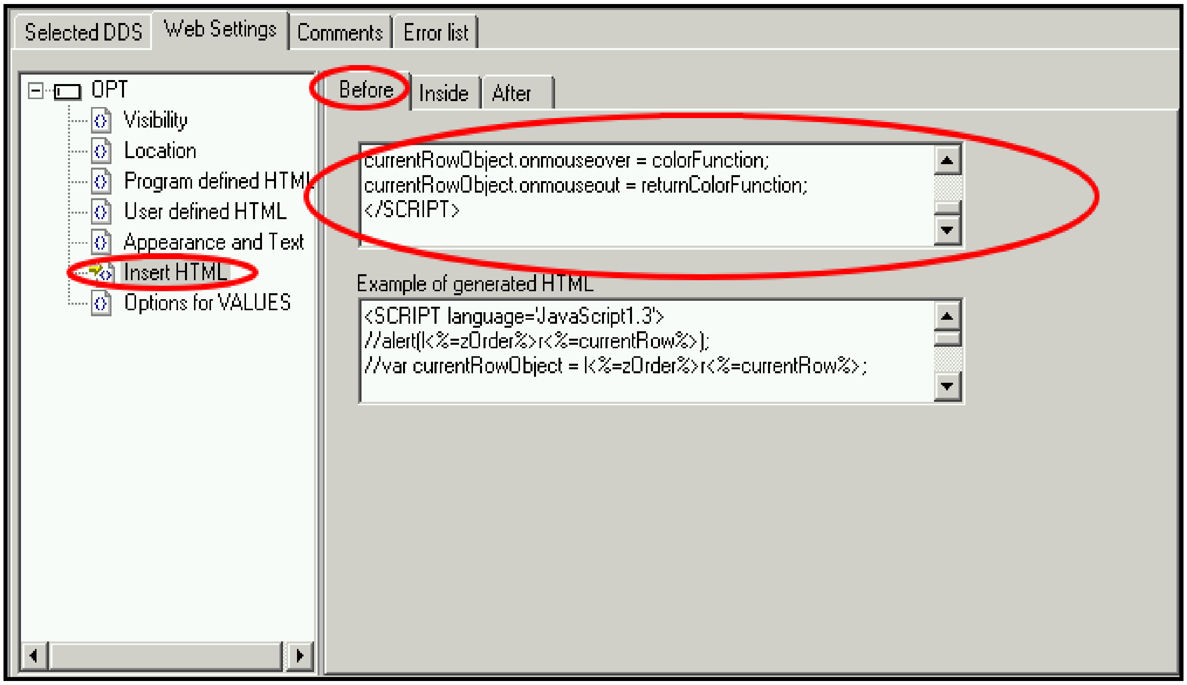


*Figure 400. Click the **Web Settings** tab*

7. In the Web Settings section of the screen, notice a list of available Web Settings for the **OPT** field. Click **Insert HTML** from the **Web Settings** list (see Figure 401).



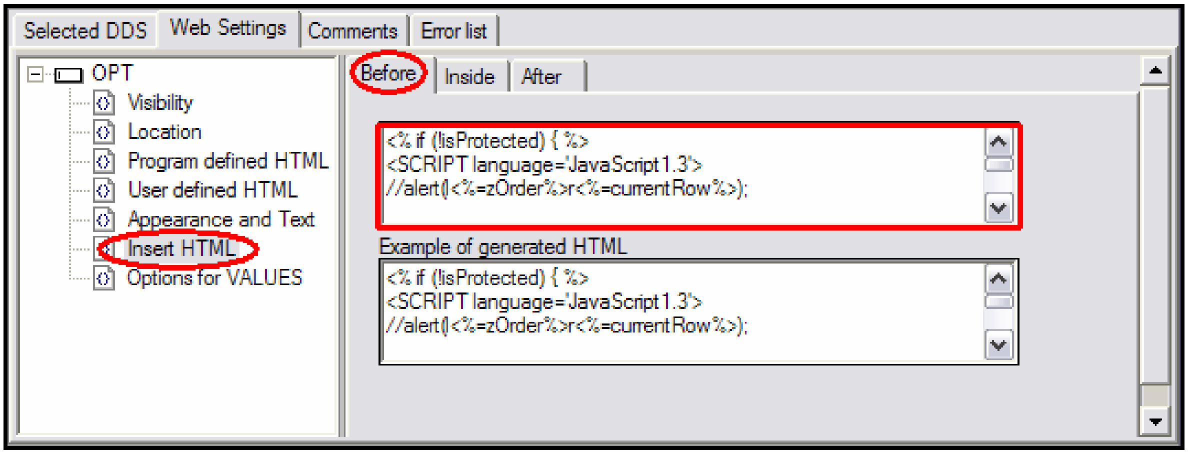
```
currentRowObject.onmouseover = colorFunction;
currentRowObject.onmouseout = returnColorFunction;
</SCRIPT>
```

Example of generated HTML

```
<SCRIPT language='JavaScript1.3'>
//alert(l<%=zOrder%>r<%=currentRow%>);
//var currentRowObject = l<%=zOrder%>r<%=currentRow%>;
```

Figure 401. List of different Web Settings available for the ***OPT*** *field*

8. Select the **Before** tab to the right of the list (see Figure 402).



```
<% if (!isProtected) { %>
<SCRIPT language='JavaScript1.3'>
//alert(l<%=zOrder%>r<%=currentRow%>);
```

Example of generated HTML

```
<% if (!isProtected) { %>
<SCRIPT language='JavaScript1.3'>
//alert(l<%=zOrder%>r<%=currentRow%>);
```

Figure 402. Click *Insert HTML* ***from the Web Settings list.***

9. Enter the JavaScript that you previously entered directly into the ORDCTL.jsp into the text area.

```
<% if (!isProtected) { %>
<SCRIPT language='JavaScript1.3'>
//alert(l<%=zOrder%>r<%=currentRow%>);
//var currentRowObject = l<%=zOrder%>r<%=currentRow%>;

var currentRowObject = document.forms("<%=(String)
session.getAttribute("UniqueId")%>").all("l<%=zOrder%>r<%=currentRow%>"
);

currentRowObject.onmouseover = colorFunction;
currentRowObject.onmouseout = returnColorFunction;
</SCRIPT>
<% } %>
```

Just to recap, you specified the following when the application is converted with the IBM WebFacing Tool; the JavaScript is inserted before the HTML `<INPUT` tag for the **OPT** field. Now this setting has been saved for re-conversion. Remember that even though this field is in the ORDSFL record, only one JSP is generated on conversion for a subfile control record and its corresponding subfile record.

10. Use **File>Save** or **Ctl-S** to save the changes you have made to the ORDENTD file. You can now reconvert the ORDENTD source member and test your application again using the steps in the previous section titled “Running the WebFacing application for this subfile example.”

Recap

You have completed “Changing the user interface using Page Designer and saving your changes for re-conversion using CODE Designer.” You now have the information to understand how to:

- Edit a converted record format JSP using Page Designer
- Understand the field names as they relate to your DDS source in the converted JSPs
- Change the Web user interface by directly editing a JSP and adding JavaScript to it
- Test the changed application
- Save the changes for re-conversion using CODE Designer
- Reconvert the application and test the changes again

Note: The green-screen user interface has not changed if the application is invoked as a green-screen application. It looks similar to the original user interface. The Web Settings apply only to the WebFacing user interface.

Configuring the WebFacing server

In this chapter, you learn how to use the WebFacing server configuration Data Area to configure the WebFacing server for up to 16 interactive subsystems.

It is desirable to configure multiple interactive subsystems if your WebFaced applications are being accessed by a large number of users. Using multiple interactive subsystems improves the scalability of WebFacing by load balancing WebFacing application jobs so that they are spread optimally among the configured subsystems.

To accomplish these learning objectives, several steps are involved, including:

- Exercise 20.1: Creating additional subsystems
- Exercise 20.2: Adding workstation entries
- Exercise 20.3: Changing the WebFacing server configuration
- Exercise 20.4: Testing the configuration

You need a 5250 emulator on your workstation to carry out the above exercises.

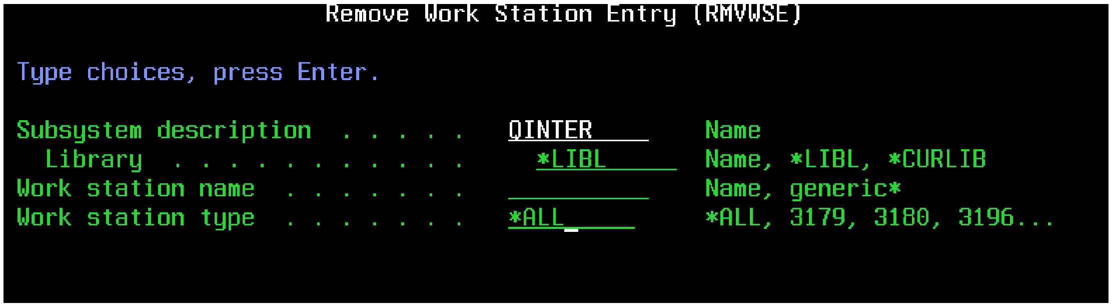
Length of time

This chapter takes approximately 45 minutes to complete.

Exercise 20.1: Creating additional subsystems

You now create additional interactive subsystems. You model these subsystems after the QINTER subsystem. Before proceeding with the creation of the subsystems, you need to modify the QINTER subsystem so that it does not intercept WebFacing job requests.

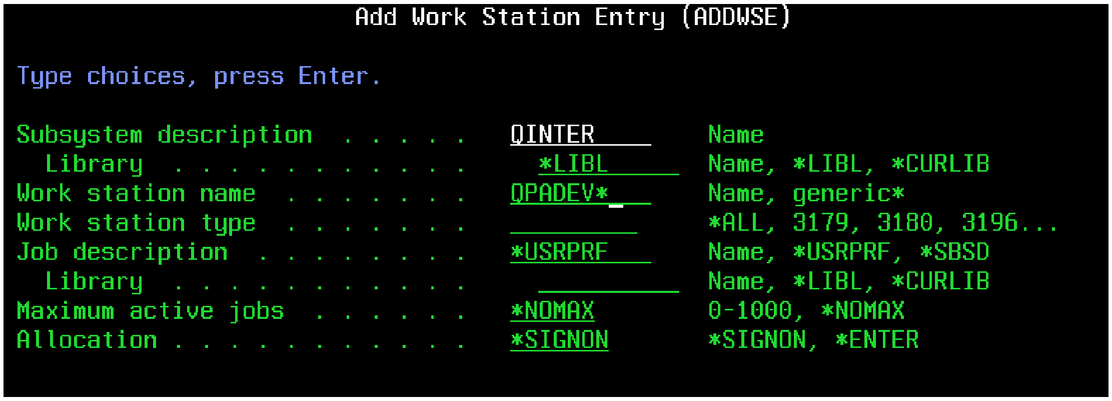
1. To prevent the QINTER subsystem from handling WebFacing application jobs, modify the subsystem to remove the *ALL work station entry and replace it with the display device name convention used for your system. For example, some systems use QPADEV*.
2. In your 5250 emulator session, issue the i5/OS **RMVWSE** command and press **F4** to prompt for the required parameters. Enter the parameters as shown in Figure 403):



```
                      Remove Work Station Entry (RMVWSE)

Type choices, press Enter.

Subsystem description . . . . .   QINTER        Name
  Library  . . . . . . . . . . .    *LIBL       Name, *LIBL, *CURLIB
Work station name  . . . . . . .                Name, generic*
Work station type  . . . . . . .   *ALL         *ALL, 3179, 3180, 3196...
```

Figure 403. Remove Work Station Entry (RMVWSE) screen

3. Add the generic device names used by your company by issuing the i5/OS **ADDWSE** command and press **F4** to prompt for the required parameters. Enter the parameters as shown below, replacing QPADEV* with your generic device names (see Figure 404).



```
                        Add Work Station Entry (ADDWSE)

Type choices, press Enter.

Subsystem description . . . . .   QINTER        Name
  Library  . . . . . . . . . . .    *LIBL       Name, *LIBL, *CURLIB
Work station name  . . . . . . .   QPADEV*      Name, generic*
Work station type  . . . . . . .                *ALL, 3179, 3180, 3196...
Job description  . . . . . . . .   *USRPRF      Name, *USRPRF, *SBSD
  Library  . . . . . . . . . . .                Name, *LIBL, *CURLIB
Maximum active jobs  . . . . . .   *NOMAX       0-1000, *NOMAX
Allocation . . . . . . . . . . .   *SIGNON      *SIGNON, *ENTER
```

Figure 404. Add Work Station Entry (ADDWSE) screen

4. You must restart the QINTER subsystem for these changes to take effect. Howver, before restarting the subsystem, you need to ensure that the interactive job from which the subsystem commands are issued is running under a

subsystem other than QINTER. You can transfer the current job from QINTER to another subsystem, for example QBASE, by issuing the following command:

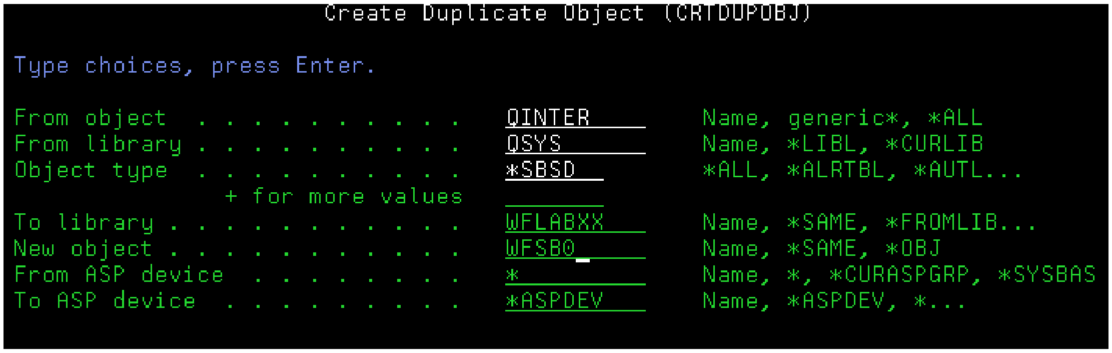
```
TFRJOB JOBQ(QBASE)
```

5. After your current job is running under QBASE, you can issue the following commands to restart the QINTER subsystem:

```
ENDSBS SBSD(QSYS/QINTER)
STRSBS SBSD(QSYS/QINTER)
```

6. You now create two interactive subsystems for handling WebFacing application jobs. The subsystems that you create can be modeled after the QINTER subsystem. To do this, you can copy the QINTER subsystem description and remove the 5250 generic workstation names QPADEV* and *CONS.

   a. To copy the QINTER subsystem description, issue the **CRTDUPOBJ** command and press **F4** to create a subsystem description with the name WFSBS0 based on subsystem QINTER in QSYS (see Figure 405).



```
                      Create Duplicate Object (CRTDUPOBJ)

 Type choices, press Enter.

 From object  . . . . . . . . . .   QINTER        Name, generic*, *ALL
 From library . . . . . . . . . .   QSYS          Name, *LIBL, *CURLIB
 Object type  . . . . . . . . . .   *SBSD         *ALL, *ALRTBL, *AUTL...
                + for more values
 To library . . . . . . . . . . .   WFLABXX       Name, *SAME, *FROMLIB...
 New object . . . . . . . . . . .   WFSB0         Name, *SAME, *OBJ
 From ASP device  . . . . . . . .   *             Name, *, *CURASPGRP, *SYSBAS
 To ASP device  . . . . . . . . .   *ASPDEV       Name, *ASPDEV, *...
```

Figure 405. Create Duplicate Object (CRTDUPOBJ) screen

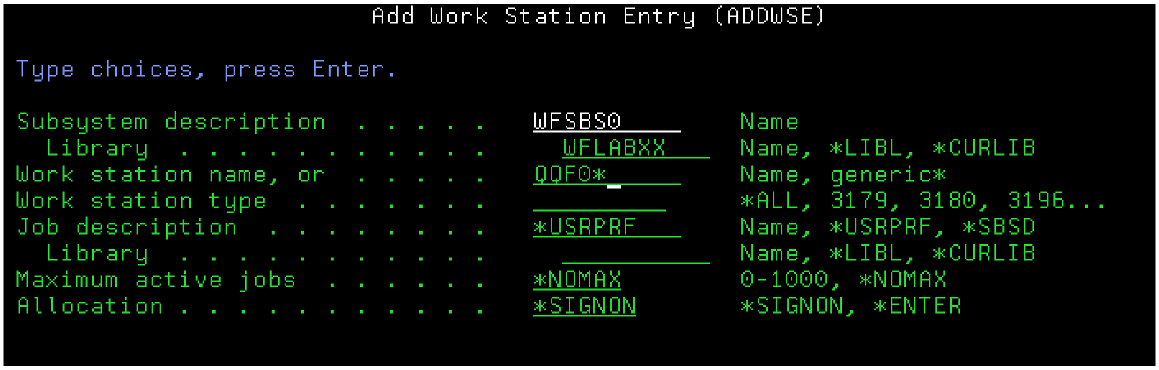
   b. Remove the workstation entries *CONS, and QPADEV* from the subsystem using the **RMVWSE** command as described earlier.
   c. Repeat steps **a** and **b** for subsystem WFSBS1.

Exercise 20.2: Adding Workstation entries for WebFacing

In this part of the exercise, you add workstation entries to the subsystems created in exercise 20.1 to handle the WebFacing application jobs.

WebFacing application job names follow the generic pattern **QQF0*** when no WebFacing specific subsystem has been configured. In this example, you configure the WebFacing server for two dedicated subsystems. The WebFacing server then generates application job names that follow the generic patterns QQF0* and QQF1*. The assigning of application job naming is in accordance with the pattern **QQFN***, where **N** is the number in hexadecimal format, of configured subsystems minus one.

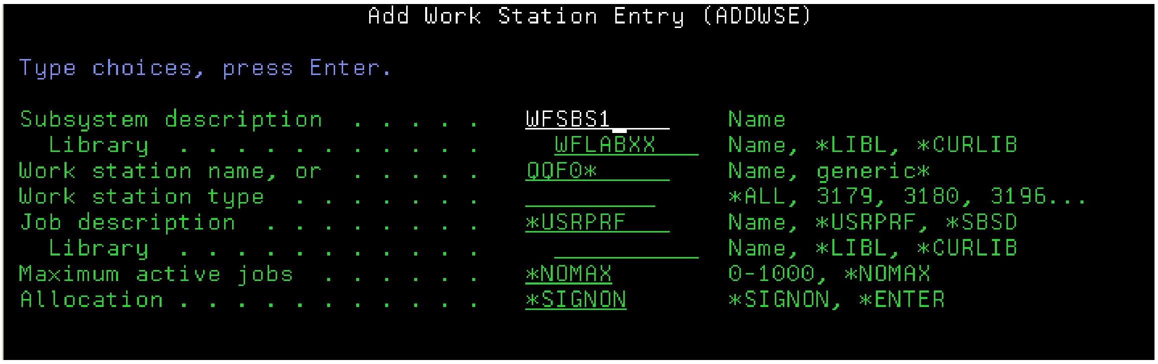
1. To add workstation entries to subsystem WFSBS0, issue the command ADDWSE as follows in Figure 406):



```
                        Add Work Station Entry (ADDWSE)

 Type choices, press Enter.

 Subsystem description . . . . . .   WFSBS0         Name
   Library . . . . . . . . . . .       WFLABXX      Name, *LIBL, *CURLIB
 Work station name, or . . . . .     QQF0*          Name, generic*
 Work station type . . . . . . .                    *ALL, 3179, 3180, 3196...
 Job description . . . . . . . .     *USRPRF        Name, *USRPRF, *SBSD
   Library . . . . . . . . . . .                    Name, *LIBL, *CURLIB
 Maximum active jobs . . . . . .     *NOMAX         0-1000, *NOMAX
 Allocation . . . . . . . . . . .    *SIGNON        *SIGNON, *ENTER
```

Figure 406. Adding workstation entries to subsystem WFSBS0

2. Repeat the above step for subsystem WFSBS1 as follows in Figure 407):



```
                        Add Work Station Entry (ADDWSE)

 Type choices, press Enter.

 Subsystem description . . . . . .   WFSBS1         Name
   Library . . . . . . . . . . .       WFLABXX      Name, *LIBL, *CURLIB
 Work station name, or . . . . .     QQF0*          Name, generic*
 Work station type . . . . . . .                    *ALL, 3179, 3180, 3196...
 Job description . . . . . . . .     *USRPRF        Name, *USRPRF, *SBSD
   Library . . . . . . . . . . .                    Name, *LIBL, *CURLIB
 Maximum active jobs . . . . . .     *NOMAX         0-1000, *NOMAX
 Allocation . . . . . . . . . . .    *SIGNON        *SIGNON, *ENTER
```

Figure 407. Adding workstation entries to subsystem WFSBS1

3. You now need to configure the WebFacing server to generate WebFacing job names according to generic names QQF0* and QQF1*.

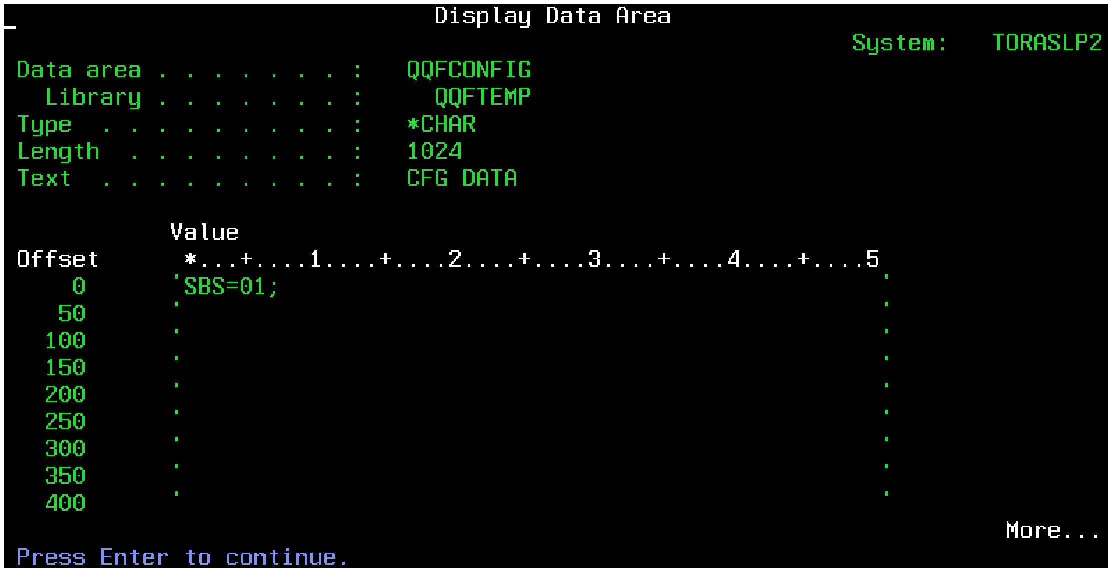
Exercise 20.3: Changing the WebFacing server configuration

In this exercise, you configure the WebFacing server to create jobs that can be assigned to the two subsystems just created.

The WebFacing server uses a data area to store its configuration data (see Figure 408).

1. Issue the following command to display the configuration data area:

```
DSPDTAARA DTAARA(QQFTEMP/QQFCONFIG)
```

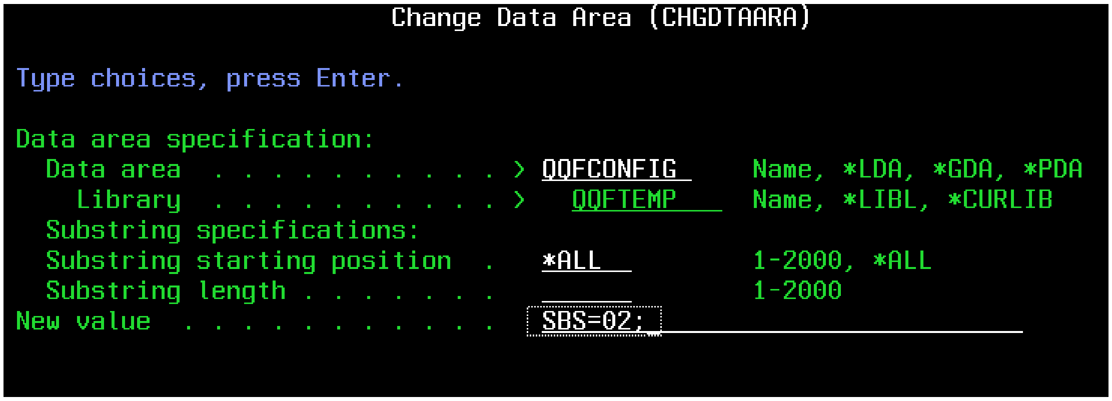


```
                              Display Data Area
                                                          System:   TORASLP2
Data area . . . . . . . :   QQFCONFIG
  Library . . . . . . . :     QQFTEMP
Type  . . . . . . . . . :   *CHAR
Length  . . . . . . . . :   1024
Text  . . . . . . . . . :   CFG DATA

           Value
Offset      *...+....1....+....2....+....3....+....4....+....5
    0      'SBS=01;                                            '
   50      '                                                   '
  100      '                                                   '
  150      '                                                   '
  200      '                                                   '
  250      '                                                   '
  300      '                                                   '
  350      '                                                   '
  400      '                                                   '
                                                                   More...
Press Enter to continue.
```

Figure 408. Display Data Area screen

2. By default, the WebFacing server is configured for 1 subsystem. To change the number of subsystems to 2 (see Figure 409), issue the following command:

```
CHGDTAARA DTAARA(QQFTEMP/QQFCONFIG) VALUE('SBS=02;')
```



```
                        Change Data Area (CHGDTAARA)

Type choices, press Enter.

Data area specification:
  Data area . . . . . . . . . . > QQFCONFIG     Name, *LDA, *GDA, *PDA
    Library . . . . . . . . . . >   QQFTEMP     Name, *LIBL, *CURLIB
  Substring specifications:
  Substring starting position  .  *ALL          1-2000, *ALL
  Substring length . . . . . . .                1-2000
New value  . . . . . . . . . . .  SBS=02;
```

Figure 409. Change the number of subsystems to 2.

Here is what you have accomplished thus far:

- The preceding configuration of the subsystems and WebFacing server causes the system to deliver WebFacing application jobs to the two subsystems.
- The WebFacing server generates application job names alternating between QQF0* and QQF1*. These jobs are assigned to subsystems WFSBS0 and WFSBS1 respectively.

In a similar way, you can create more subsystems and assign the appropriate workstation entries to them, and configure the WebFacing server to control the generic naming pattern that the WebFacing server uses. This capability allows you to control the spread of WebFacing application jobs across multiple subsystems. The WebFacing server can support a maximum of 16 subsystems, allowing application jobs of the following names to be created: QQFN* where N = 0 to F.

You now proceed to test the new configuration.

Exercise 20.4: Testing the configuration

To activate the new configuration, you need to do the following:

1. End the server by issuing the following command:

   ```
   ENDTCPSVR *WEBFACING
   ```

2. Start the new subsystems by issuing the following commands:

   ```
   STRSBS SBSD(WFLABXX/WFSBS0)
   STRSBS SBSD(WFLABXX/WFSBS1)
   ```

3. Restart the server by issuing the following commands:

   ```
   STRTCPSVR *WEBFACING
   ```

4. You can now invoke multiple sessions of a WebFaced application from multiple browser instances to observe the spread of the WebFacing application jobs across the subsystems. To observe the subsystems, issue the following command:

   ```
   WRKACTJOB SBS(WFSBS0 WFSBS1)
   ```

   The WebFacing QQF0* and QQF1* jobs are assigned to subsystems WFSBS0 and WFSBS1, respectively.

Recap

You have completed “Configuring the WebFacing server.” You now have the information to understand how to:

- Create additional subsystems dedicated to WebFacing application jobs
- Change the WebFacing server configuration Data Area to control the generic name patterns used by the WebFacing server when generating WebFacing application job names
- Control the spread of WebFacing application jobs across multiple dedicated subsystems

Controlling the spread of the jobs across subsystems allows you to tune the performance and scalability of the WebFacing server depending on the number of concurrent clients accessing WebFaced applications on the target system.

Summary

As you can see with the IBM WebFacing Tool, you can quickly convert your DDS display file source members so that the user interface of your System i programs can run in a browser. When you convert your DDS display files, JSP files and Java beans are generated for you that substitute for the DDS code and make Web access possible.

In the WebFacing Project wizard, you can select one or more DDS source members to convert, and select a Web look and feel from one of several predefined styles, or you can design your own Web style for use with your applications. The tool creates three JSP and XML files for each record format. The XML files hold the data for the record format, or control its appearance or other characteristics, and the JSP handles displaying the Web version of the screen, prompting for data, and handling input errors. The wizard generates an application home page to launch the Web-enabled version of your program.

When you invoke a converted application from the browser, the WebFacing server on the System i model starts the server program. The server intercepts all calls to READ, WRITE, and EXFMT operations to DSPFs, so that in many cases your program (*PGM) can run without modifications, and without even detecting that it is being accessed using WebFacing. You might need to make coding changes if your application uses DDS keywords that are not supported by WebFacing, or if you want to modify the DDS screens so that the conversion to Web format produces a more attractive or consistent result.

Note: For more information on WebSphere Development Studio Client and the IBM WebFacing Tool, see: http://**ibm.com**/software/awdtools/wdt400.

Appendix B. About the author

Michael J. Sandberg
e-business technical consultant, IBM ISV Solutions Enablement

Michael J. Sandberg is a WebSphere specialist, focusing on Web-to-host technologies for the System i platform. His role within IBM PartnerWorld is to assist solution providers in developing new online applications for the System i platform using IBM strategic tools and technologies such as the WebSphere family of products.

Trademarks

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
the IBM logo
eServer
iSeries
System i
i5/OS
WebSphere
PartnerWorld

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Information is provided "AS IS" without warranty of any kind.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.