Table of Contents

Instructions

Requirements

- Java is required for the audio
- JavaScript is required to navigate

Click here to test your browser for compatibility.

Browsers

If you have one of these browsers you should be set.

- Netscape Communicator/Navigator 4.0.5 or better
- Microsoft Internet Explorer 4.0 or better

You can download this course and take it off-line

This course can be downloaded so you can conveniently take it off-line. You will need to have installed a copy of Adobe Acrobat Reader (you can download this reader by clicking here). If you want to download the Acrobat files related to this course click here. There are two separate files to download. One contains the foil screens (graphics), the other contains the speaker notes (text). The foil and speaker notes pages have been numbered to help you in putting them together.

Hotlinks

As you go through this course, you will observe, and probably take advantage of, frequent hotlinks that send you to other Web sites for more details about a particular topic. These hotlinks have been compiled in one long list at the end of the course, along with enough of a descriptor to remind you what they are about, and can be accessed by clicking on the LINKS option in the navigator bar.

We think these links are important, but you may prefer to visit these sites at some later point so that, for right now, you can stay focused on the “meat” within this course. You can clip the contents of this “Hotlinks” page into a document file on your local PC drive. Then, at your convenience, you can use these saved hotlinks to explore these topics in greater detail.

The Hotlinks page also has other Web site and reference information that pertains to this course.

Audio

All course pages in this IBE include streaming audio. You will see a button that looks like this in the lower left corner of the course window:

The audio clips will automatically start playing, so you only need to press this button if you want to pause or replay the clip.

You can turn the audio on and off. Click the 'Audio' button shown:

Speaker Notes
Open or close the Speaker Note window with these buttons.

Navigation
These are the general navigation buttons:

- Click NEXT to go forward one page.
- Click PREV to go back one page.
- The TOPICS page lists all the pages in this IBE for quick navigation.
- The LINKS page contains the hotlinks to related information.
- The TERMS page contains definitions of terminology used in this course.
- The FAQ page lists frequently asked questions about the subject.
- The QUIZ page asks you questions about what you've learned.
- Click EXIT to leave the course.

Author Biographical Information

George Farr,
Technical Development Manager
IBM Toronto Development Team

George N. Farr is currently the technical development manager for the RPG products, and an exciting development environment coming to an iSeries near you. He has held many positions including developer, team leader, development manager, and has been an iSeries and AS/400 planner and architect since 1985. He is a frequent, award-winning speaker at COMMON conferences, iSeries user groups, and other conferences worldwide. He has co-authored many industry magazine articles and many books including: *ILE: A First Look* and *RPG IV By Example*, and *Java For RPG Programmers* and *Java For S/390 and AS/400 COBOL Programmers*

George currently resides in Toronto, Ontario, Canada.

farr@ca.ibm.com

Author Biographical Information

Phil Coulthard,
Lead Architect, iSeries Application Development
IBM Toronto Development Team

Phil Coulthard is currently the lead architect for iSeries application development products at the IBM Toronto laboratory. He has worked as a developer, team leader, manager and general advocate of iSeries and AS/400 application development since 1986. He is also an author of many industry magazine articles and a frequent award-winning speaker at COMMON and other conferences and user groups. Together with George Farr, Phil co-authored the books *Java for RPG Programmers* and *Java for AS/400 and S/390 COBOL Programmers*.

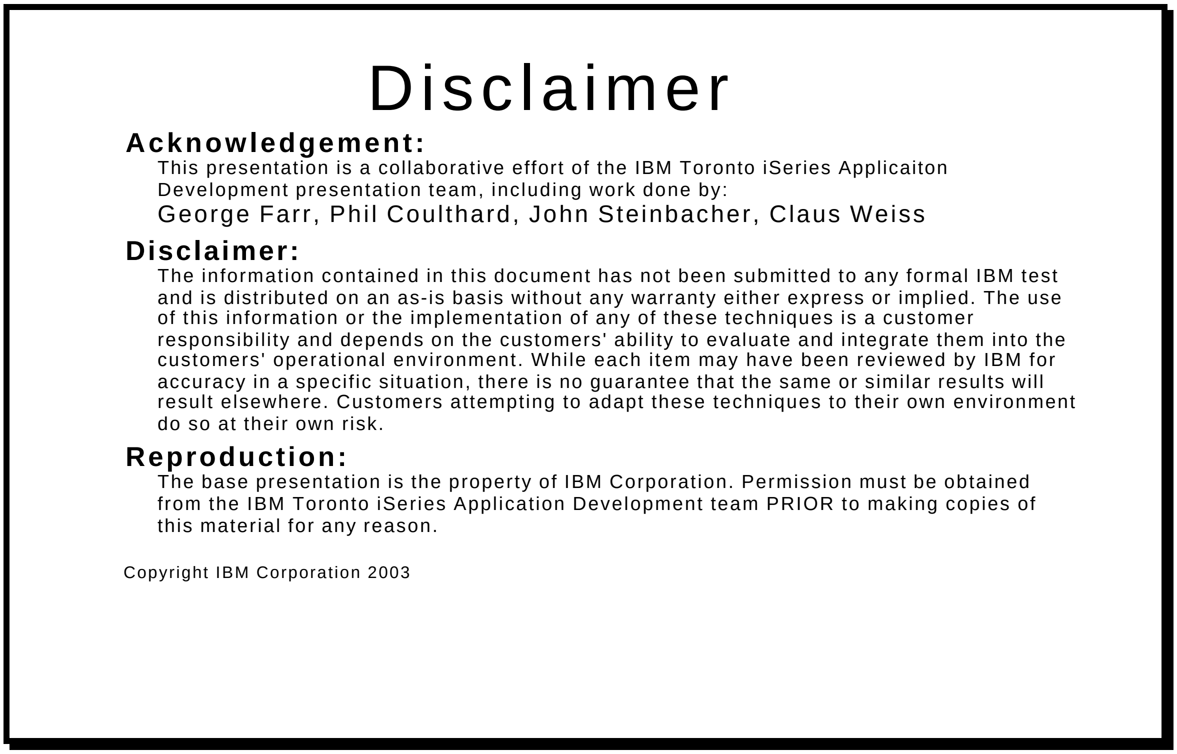
Phil currently resides in Aurora, Ontario, Canada.

coulthar@ca.ibm.com

Content updated by Michael Sandberg

Michael Sandberg,
iSeries e-business Consultant
IBM eServer Global Solution Enablement

Michael Sandberg is a member of the IBM eServer Global Solution Enablement team located in Rochester, Minnesota, and works to help IBM solution developers and customers integrate new iSeries technologies into their programming environments. These technologies include WebSphere Application Server, WebSphere Development Studio Client, IBM WebFacing Tool for iSeries, iSeries Access for Web, and WebSphere Host Publisher.



Disclaimer

Acknowledgement:

This presentation is a collaborative effort of the IBM Toronto iSeries Applicaiton Development presentation team, including work done by:

George Farr, Phil Coulthard, John Steinbacher, Claus Weiss

Disclaimer:

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

Reproduction:

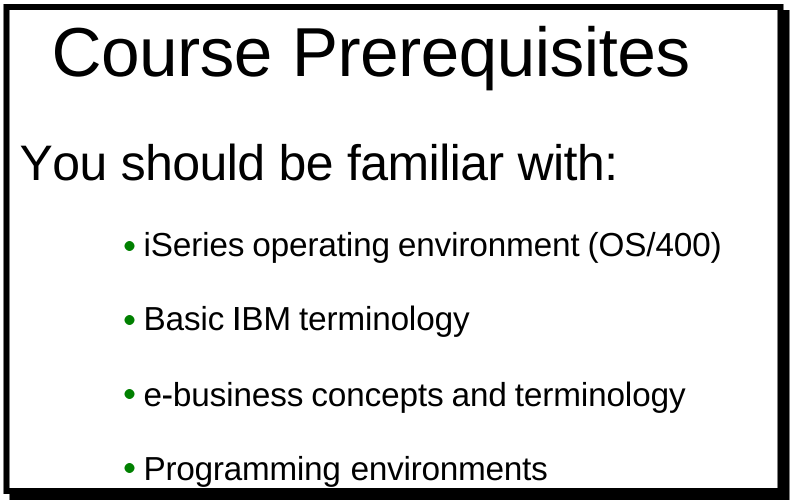
The base presentation is the property of IBM Corporation. Permission must be obtained from the IBM Toronto iSeries Application Development team PRIOR to making copies of this material for any reason.

Disclaimer

Acknowledgment: This presentation is a collaborative effort of the IBM Toronto iSeries Application Development presentation team, including work done by: George Farr, Phil Coulthard, John Steinbacher, and Claus Weiss. This course has been updated by Michael Sandberg of the IBM eServer Global Solutions organization to contain V5R2 release content.

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

Reproduction: The base presentation is the property of IBM Corporation. Permission must be obtained from the IBM Toronto iSeries Application Development team PRIOR to making copies of this material for any reason.



Course Prerequisites

You should be familiar with:

- iSeries operating environment (OS/400)
- Basic IBM terminology
- e-business concepts and terminology
- Programming environments

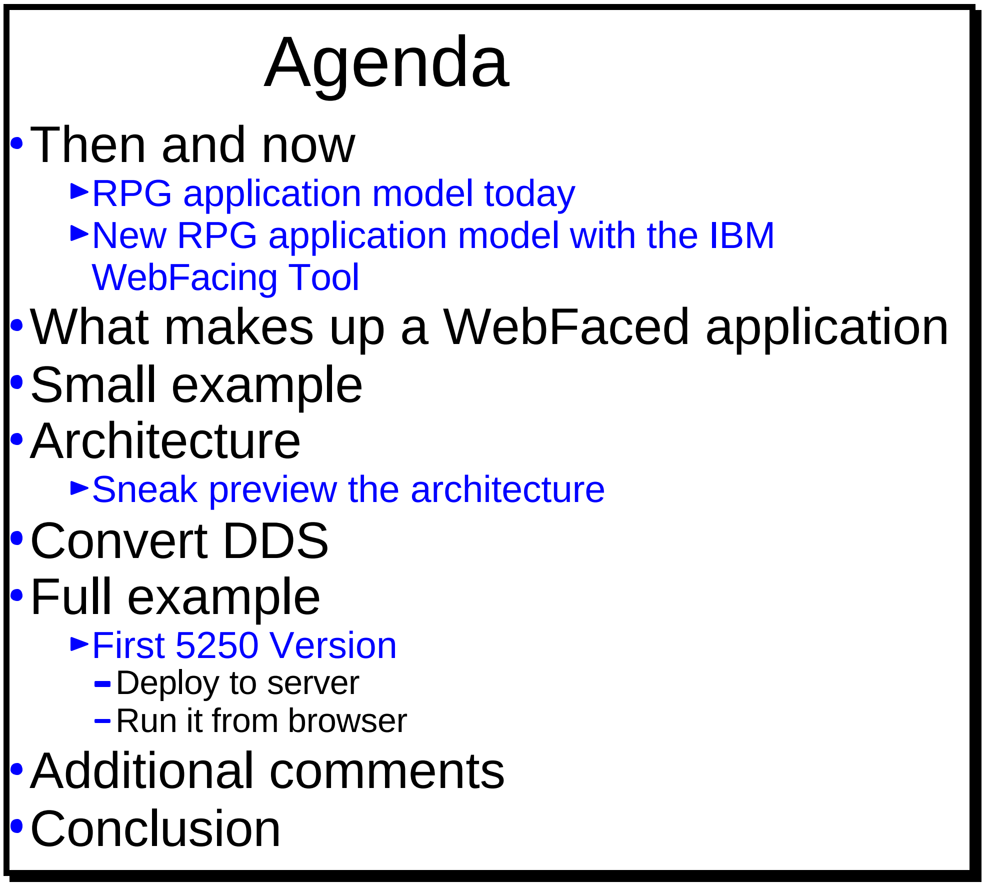
Course Prerequisites

Hi there, and welcome to "**WebFacing your iSeries Applications – V5R2 Update**," an on-line education course. First off, let's review the course prerequisites. We will assume that you are familiar with the IBM® eServer® iSeries™ operating environment, as well as basic IBM terminology. Therefore, many terms will not be further explained in this learning course. However, we do provide a glossary at the end of the course that is a review of many acronyms and terms used throughout this material. Should you want to access the glossary while taking the course, click on the link provided in the navigation bar.

Additionally, you should have a general understanding of how the Internet works and what tools and techniques are widely available today to address consumer needs. The majority of this material assumes a basic understanding of Web concepts. Additionally, to get the most out of this course and comprehend the benefits presented for Web-facing, you should also understand how programming environments work.

If at any point, you wish to jump back to a previous section of this course, simply click on the "Agenda" hotlink in the navigation bar, and then click on the segment you want to review.

You will also be occasionally quizzed on the material you have learned, so that you can get a sense for your level of understanding before moving on to new topics. Then, at the end of the course, you will have the opportunity to take a final test regarding all the material.



Agenda

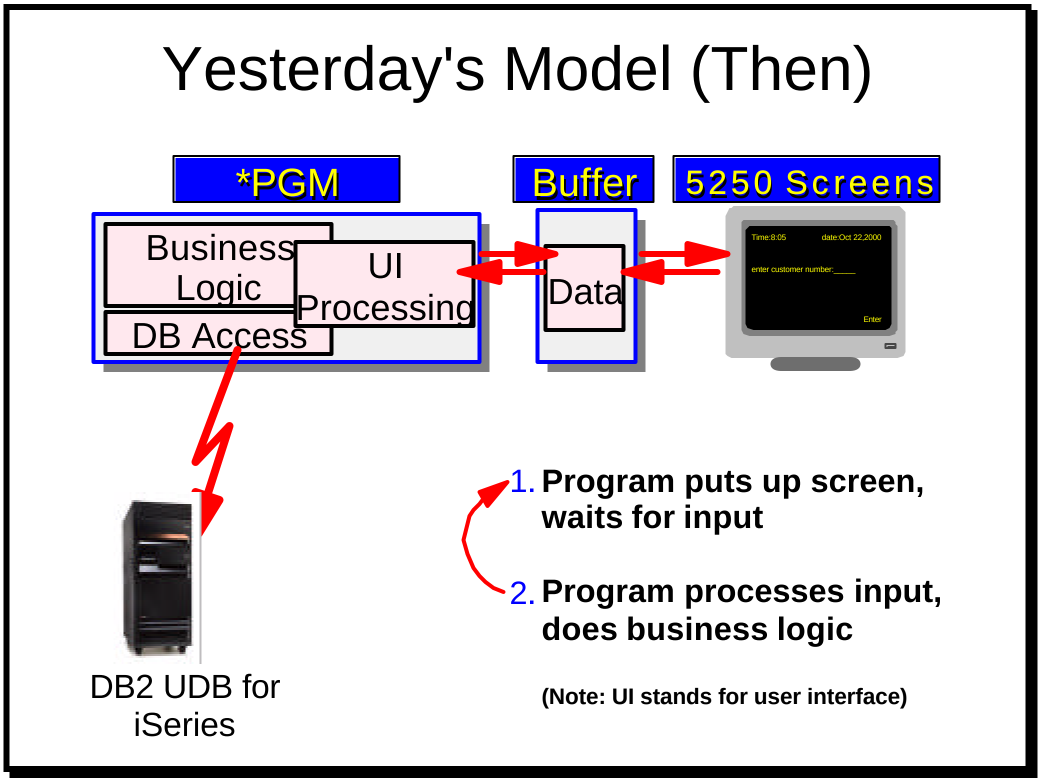
- Then and now
  - RPG application model today
  - New RPG application model with the IBM WebFacing Tool
- What makes up a WebFaced application
- Small example
- Architecture
  - Sneak preview the architecture
- Convert DDS
- Full example
  - First 5250 Version
    - Deploy to server
    - Run it from browser
- Additional comments
- Conclusion

Overview/Agenda

Here is the agenda for “**WebFacing your iSeries Applications**.” First, to gain an appreciation of the effort that is involved in Web-facing an application, we will look at the older RPG model that many developers and programmers became familiar with. Then, we will contrast that model with the newer model that involves Web-facing an application by using the IBM WebFacing Tool for iSeries. Additionally, we will touch on other available ways to get an application on the Web.

Next, we review exactly what makes up a Web-faced application and show a small example. Then, the pros and cons of Web-facing an application are covered to help you in the decision-making process regarding the exact method(s) that are used in a Web-facing effort.

Before reviewing a full example, we will explain the architecture of IBM WebFacing Tool for iSeries and look at converting DDS (Data Description Specifications) so that the example will be straightforward. In conclusion, we will discuss the full example and wrap up the course with our closing comments and ideas. At the end of this on-line course is a list of Web sites that provides additional information related to this course, if you are interested.



Yesterday's Model (Then)

*PGM

Business Logic

UI Processing

DB Access

Buffer

Data

5250 Screens

DB2 UDB for iSeries

1. **Program puts up screen, waits for input**
2. **Program processes input, does business logic**

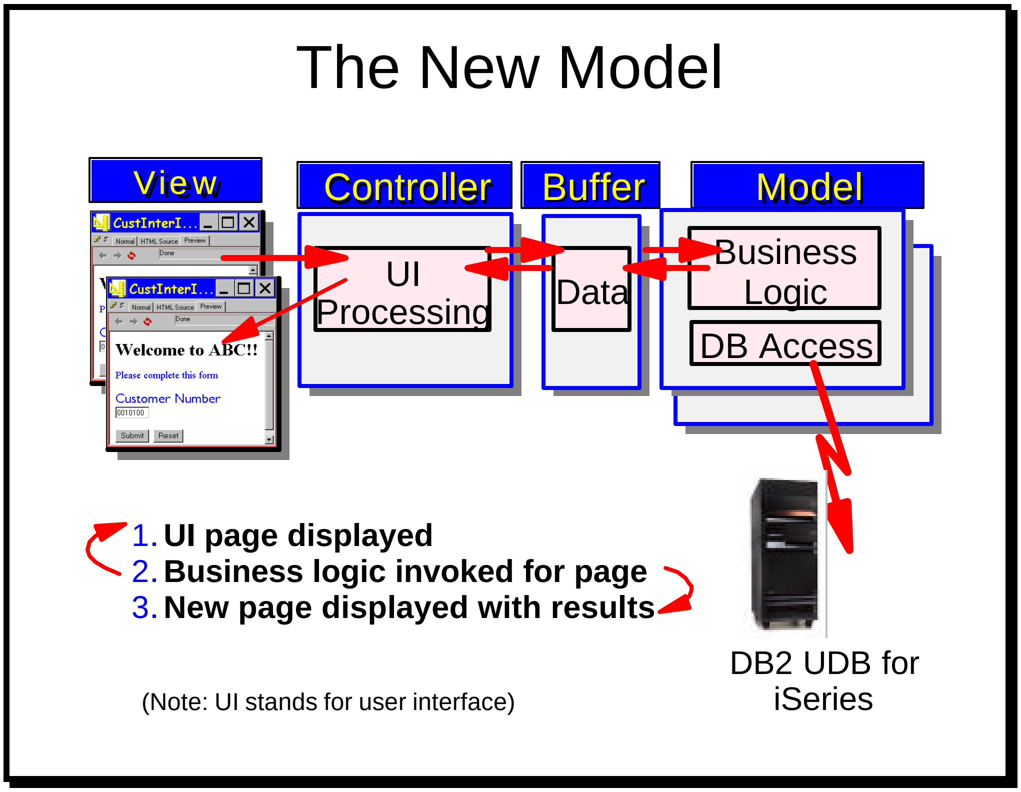
(Note: UI stands for user interface)

Yesterday's RPG Model (Then)

This graphic illustrates the commonly-used programming model for iSeries servers. When an application was written, everything was put together. A user interface was created with DDS. The RPG or COBOL program included code to: write and read the screen, process the screen based on what keys were pressed, and make decisions about what to show the user next. The programmer also wrote code for the business, database access, and report logic. All this coding “blended” nicely into a program.

The problem with yesterday’s approach is the difficulty faced when substituting or augmenting any one piece of a program, such as the user interface (UI). UIs are difficult to change because the code is married to the 5250 interface. The IBM WebFacing Tool for iSeries allows the blended program to continue to run as is—thinking it is talking to a 5250 when, in fact, you are replacing the green-screen interface!

This tool does not address all the requirements for developing new, Web-enabled applications—it has a specific role. If an application is strategic with long term needs for a newer interface, you can go into the code, rip out the 5250 interface from the business logic, and build a new application that separates the UI from the business logic. However, it is hard to rip out the 5250 interface—this is not what the IBM WebFacing Tool for iSeries is about. Instead, this tool provides a short-term bridge to get an application running in a Web browser when that application already uses a 5250 interface. The IBM WebFacing Tool for iSeries “buys time” while you rewrite the program. Also, the tool could be a good, long-term option for an application where you want to maintain both the 5250 interface and a Web-user interface.



The New Model

1. UI page displayed
2. Business logic invoked for page
3. New page displayed with results

(Note: UI stands for user interface)

The New Application Model

In a perfect world, you would use the new architectural model illustrated here and write units of code that work in concert with each other. First, a controller piece of code is written whose job is to interact with the user. The controller code handles the data input from the user and then passes that data input to the next program unit—the business logic or “model.” This logic has no code capable of processing the user interface or deciding the UI flow. That is the role of the controller. Rather it only does discrete business logic functions such as: create a customer, delete a customer, do payroll, etc. This is where all database logic resides. When the model unit of code makes a response back to the controller code, the controller code decides how to update the user interface or “view” to reflect that response. In a Web application, for example, the controller code decides which Web page to show next and sends the response to that page. You might notice, though, that the whole request-response scenario starts with the user in some initial UI, such as the initial menu or initial Web page. The selections made by the user from that point on decide which discrete business logic APIs are invoked in response. Unlike a traditional architecture, in the new world, the UI drives the logic, not the other way around.

In the scenario shown here, the business logic program would not know anything about the user interface and would not be making decisions about what screen to show next. It would simply get a request, with data, and give back the response, which is data. The controller code would do all the user interface thinking. The buffer going back and forth between the business logic model and the controller would be language-dependent. For example, using current technology to create a Web application, the controller would be a Java® servlet, the buffer would be a JavaBean™, and the view would be a Web page—perhaps a JavaServer Page™ (JSP)™.

The “new” model, which is depicted in this graphic, is a classic, industry-design pattern and represents the goal most developers want to achieve. When the three application components — view, controller, and model (which includes business logic as well as database access)—are separate pieces of code, you can more easily move into the future by setting yourself up to embrace ever-changing technologies without having to rewrite the entire application.

This is because, while the final answer today for user interfaces is the Web, the final answer tomorrow will be different. In light of these facts, you need to be able to easily modify each code component independently. So, how does the IBM WebFacing Tool for iSeries help? It moves you toward this goal, although not all the way. It generates the JSPs and servlet, but the business logic is still contained in the existing RPG or COBOL program as-is, still with its 5250 I/O (controller) logic. Therefore, the model still has the logic to determine screen flow, and the servlet acts in a more mundane role of feeding data to and from the JSP. The tool does not re-architect your application to achieve this model/view/controller goal. It is simply designed to quickly give an alternative user interface to your display file—specifically, a Web user interface. This tool is a compromise, while you write your new native Web applications. To some extent, it allows you to emulate the view/controller/model paradigm, without rewriting existing blended applications.

Ways To "Web"

Lotus Domino

- Can serve document databases to Web
  - Can access Java via agents
  - Can access RPG or COBOL

IBM WebSphere Commerce

- Formerly IBM Net.Commerce
- For selling over the Internet
  - Store fronts, Internet bidding, etc.
- Uses servlets, EJBs
  - Can access RPG or COBOL via ILE C++ or Java

Ways to “Web”

Before discussing IBM WebFacing Tool for iSeries in more depth, let's look at different ways to “Web." You might be familiar with some of these.

For starters, using Lotus® Domino™ is a fantastic way to get an application up on the Web if you are not tied to the 5250 logic that dominates much of the iSeries application landscape. Domino can serve document databases to the Web. Also, programmers will often use Domino and JSPs together. WebSphere® applications are used for transactions and Domino is used for collaborative types of things.

Another way to “Web" is via the use of WebSphere Commerce. WebSphere Commerce is useful if you do not want the Web application to get into all the details related to money-based Web transactions, such as taxes, credit card information, and secure payment servers. The features of WebSphere Commerce are ideal if selling over the Internet is a requirement. WebSphere Commerce uses servlets and EJBs™ (Enterprise JavaBeans™) and can also access RPG, COBOL, C++, or Java programs.

Other Ways To Web

- Java applets
  - Embedded in HTML pages
    - Authored with VisualAge for Java
- IBM WebSphere Development Studio Client for iSeries
  - Design JSPs and HTML
  - Call RPG programs and procedures

Other Ways to “Web”

Java applets are yet another vehicle for getting applications to the Web. For example, with an in-house application where everyone is connected via a T1 line, applets are great. However, for code deployed over the Internet, applets can be a performance concern because it is not possible to predict how long it will take for the applets to come down to the browser.

In general, most Java developers avoid applets today because of the download time. In particular, using applets to handle the entire application has pretty much stopped, because of the frustrating “applet loading” message. But little applets sprinkled in an application can be nice. If you decide to incorporate the use of applets because your company does have a well-controlled network, then the applets would be used inside HTML pages and would be created with WebSphere Development Studio Client for iSeries. If you are already comfortable with RPG, this WebSphere product includes VisualAge for RPG, allowing you to write GUI in RPG, and then automatically generate Java code! This creates “thick client” GUI applications, which in essence, are applets.

IBM also offers some unique iSeries support in its WebSphere Development Studio Client for iSeries—for managing Web sites and designing Web pages. The iSeries support comes in the form of iSeries-specific plug-ins that have been added to the Eclipse-based tooling. These plug-ins can be used when creating new applications. *[NOTE: The idea of the Eclipse Project is to create "Apache for developer” tools— an open source framework that provides many of the underlying services software developers need. This would be a "toolkit for designing toolkits." Not just a set of APIs, the framework will consist of real code designed to do real work.]*

Other Ways To "Web" (continued)

CGI-Bin Program

- Like a servlet, but written in compiled language like RPG
  - Original or classic way to create HTML pages "on the fly"
  - Not as easy, not as portable, not as scalable, as servlets + JSPs

IBM WebFacing Tool for iSeries

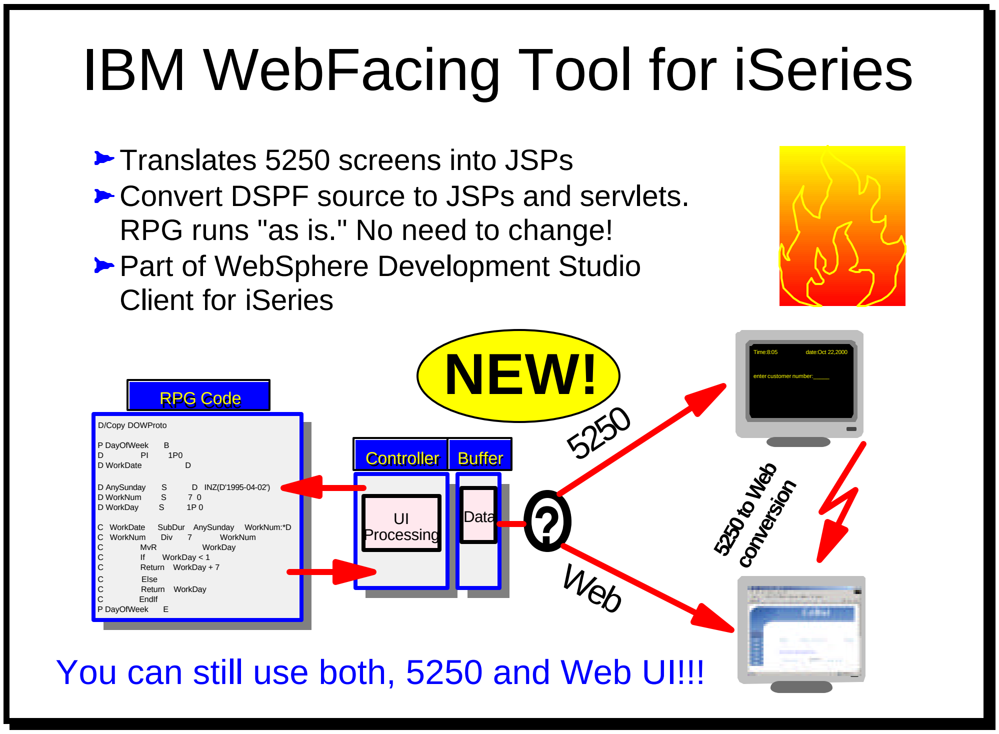
- Part of WebSphere Development Studio Client for iSeries & WebSphere Development Studio Client for iSeries Advanced

Other Ways to “Web” (continued)

There are still “more ways to ‘Web’!” The CGI-bin interface was an early solution in the history of the Internet if you needed to create interactive Web-based applications. CGI stands for “common gateway interface,” while “bin” stands for “binary.” CGI is a standard method used by a Web server to pass browser requests to an application program and for the Web server to receive data back from the application program to forward to the user. Since the CGI interface is a standard, a CGI application can be written in many different languages; including C, C++, RPG, COBOL, and PERL.

CGI-bin programs were the original way HTML pages were created “on the fly." However, there are some disadvantages to CGI, which has led to the rapid acceptance of servlets and JSPs. First, CGI has no industry-standard way of externally describing the Web pages. You either hard-code the HTML tags into your business logic, or you invent a way to define substitution variables in your HTML source file. In either case, you will not get the benefit of many industry tools for authoring the Web pages, as JSPs offer. Second, by default, CGI programs are stateless and hence proprietary coding techniques are needed to remember field data between Web pages. Servlets offer industry standard APIs for this. Third, CGI programs written in compiled languages do not enjoy the benefits of Java, such as server portability, security, and memory management. That being said, you can look at servlets as Java CGI programs, and JSPs as an industry-standard way to externally describe the Web pages.

Today, JSPs and servlets are where IBM is currently focusing its resources and this is also where the IT industry is heading. Because servlets and JSPs are in demand by programmers who develop Web interfaces for iSeries servers, IBM now offers the IBM WebFacing Tool for iSeries (a part of the WebSphere Development Studio Client for iSeries) as an alternative to get applications up and running on the Web, leveraging servlets and JSPs. This tool is exciting because it is relatively painless to implement and it also incorporates IT industry standards in its approach. Let’s see how it works.



IBM WebFacing Tool for iSeries

- Translates 5250 screens into JSPs
- Convert DSPF source to JSPs and servlets. RPG runs "as is." No need to change!
- Part of WebSphere Development Studio Client for iSeries

```
D/Copy DOWProto

P DayOfWeek      B
D                PI        1P0
D WorkDate                      D

D AnySunday       S          D   INZ(D'1995-04-02')
D WorkNum         S          7  0
D WorkDay         S          1P 0

C   WorkDate     SubDur   AnySunday    WorkNum:*D
C   WorkNum      Div      7            WorkNum
C                MvR                   WorkDay
C                If       WorkDay < 1
C                Return   WorkDay + 7
C                Else
C                Return   WorkDay
C                EndIf
P DayOfWeek      E
```

Time:8:05 date:Oct 22,2000

enter customer number:______

You can still use both, 5250 and Web UI!!!

IBM WebFacing Tool for iSeries

The IBM WebFacing Tool for iSeries is exciting. It allows you to easily generate JSPs and servlets from 5250-based programs. This is achieved by treating the entire blended program as “business logic only”—even though the program contains both business logic and user interface logic (that is, the RPG program, by virtue of its structured nature, determines which screen to display next and what to include on that screen).

The IBM WebFacing Tool for iSeries generates JSPs, JavaBeans, and servlets—based on the 5250 display file DDS—to create a Web interface. This Web interface code works in concert with the RPG program, quickly allowing strategic business applications to be presented on the Web. Generally, using the UI requires no changes to the RPG program, and the application can still also be accessed via a 5250 interface. *[NOTE: RPG program changes are needed only if the DDS includes uncommon DDS keywords not yet supported by IBM WebFacing Tool for iSeries. Keyword support can be verified by a downloadable tool on the IBM WebFacing Tool Web site.]* With this tool, companies can move, with minimal programming changes, toward the new Web-facing model discussed earlier in this course.

The IBM WebFacing Tool for iSeries is part of the WebSphere Development Studio Client for iSeries product, which in turn is part of WebSphere Development Studio. The IBM WebFacing Tool for iSeries consists of two pieces—a conversion tool for the DSPF-to-JSP conversion; and a runtime intercept that enables an iSeries application to run as a Web-faced application for the Web. The IBM WebFacing Tool for iSeries includes a simple wizard to facilitate the process of Web-facing an application.

[NOTE: Check the Web for PTFs and service packs before using the IBM WebFacing Tool for iSeries because regular and repeated updates will be available through those channels. IBM will continue to expand the DDS keywords supported. Reference the "Hotlinks" section of this course for a link to these PTFs, service packs, and DDS keyword information.]

IBM WebFacing Tool for iSeries

- What is it?
  - Something really cool!!!
  - Easily converts existing interactive applications into e-business applications!
    - **Development time: convert DSPF DDS to JSPs and Beans**
    - **Runtime: intercept buffers to drive JSPs versus DSPFs**
  - Allows interactive optimization for the Web
  - Changes the ***face*** of iSeries servers

What is IBM WebFacing Tool for iSeries?

So what exactly is this tool and why was it developed? IBM has had a very positive response to this product.

Besides converting existing, interactive applications into e-business applications, the IBM WebFacing Tool for iSeries addresses specific requirements. Currently, there are many technologies for converting 5250 screens at runtime to something else that lasts for a split second in memory. These technologies continue to have a contributory role to play. However, the IBM WebFacing Tool for iSeries complements these and other Web-enabling technologies.

The IBM WebFacing Tool for iSeries is designed for development shops with strategic applications that need to be quickly evolved into Web applications, while also preserving the 5250 user interface. These developers want development-time control over the Web user interface, just as they have development-time control over the 5250 user interface. Developers want to leverage their skills, maybe even their skills in SDA (Screen Design Aid) to create new interfaces.

The development-time conversion window provided by this tool is its key differentiator when compared to other Web-facing options. During the development of the IBM WebFacing Tool for iSeries, there is a source-to-source conversion (DSPF DDS is converted to JSPs and JavaBeans). To run a “Web-faced” program, the entire application must have been taken through the Web-facing process. Then at runtime, the call to workstation data management is intercepted. In fact, IBM changed the iSeries operating system to enable this tool to function properly—to intercept the workstation data management call. The key is that Web-faced applications skip the normal step of producing a 5250 datastream—it simply never happens. Instead, the data comes to the workstation data manager, which passes it immediately to the IBM WebFacing Tool for iSeries servlet. This servlet, in turn, passes the data to the generated JSP for that record format.

One important benefit of using this tool is that all the components—JSPs and JavaBeans—are created at development time. Nothing is generated at runtime, so the user interface is already waiting for the application. At runtime, the buffers are intercepted to drive the JSPs versus the Display File. The result is, in many cases, good performance. Other tools that produce a Web-enabled front end for 5250 applications require two conversions. First, workstation data management merges the application-written data with a display file to produce a 5250 datastream, which in turn must be converted to something else—for example, HTML. However, with the IBM WebFacing Tool for iSeries, creation of the 5250 datastream is skipped.

The second important benefit of this tool is that since it utilizes a source-to-source development time conversion, you can iterate the output source at development time to ensure that is presented exactly as expected—before deployment to production.

1st Mini Quiz

1. In yesterday's commonly-used RPG design model, the RPG program would include code to:
   ☐ a. Write the screen
   ☐ b. Read the screen
   ☐ c. Process the screen
   ☑ d. All of the above—Correct. They also had to make decisions about what to show the user next, as well as write code for the business logic, the database access logic, and the report logic.

2. The IBM WebFacing Tool for iSeries does NOT allow the RPG "blended" program to continue running as is, thinking it is talking to a 5250 user interface.
   ☐ True
   ☑ False —Correct. The tool DOES allow the RPG program to continue to run as is, thinking it is talking to a 5250 user interface. However, the green-screen is replaced!

3. The IBM WebFacing Tool for iSeries tool consists of two pieces.
   ☑ True—Correct. It consists of a conversion tool for the DSPF-to-JSP conversion, and a runtime intercept that enables an iSeries application to be run as a Web-faced application for the Web. The tool also includes a wizard to facilitate the process of Web-facing an application.
   ☐ False

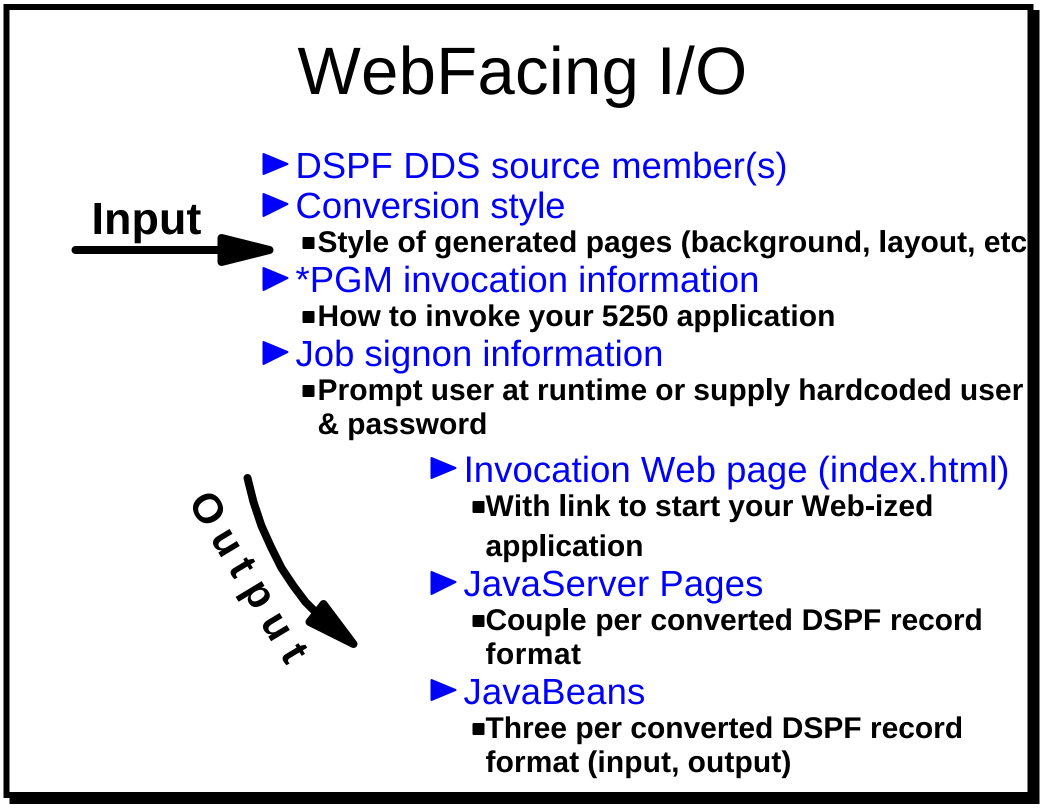
Goals of IBM WebFacing Tool for iSeries

- ►Quickly Web-enable many applications
- ►Allow iterative optimization for the Web
- ►Introduce no new costs to application stack price

Goals of WebFacing

The IBM WebFacing Tool for iSeries was announced April 23, 2001 and has been steadily enhanced since. IBM developed it so that developers could quickly Web-enable many of their strategic, in-house applications. The tool permits reliable business applications to run on the Web without introducing any new application stack costs. The runtime support is available in the form of PTFs for OS/400® V5R1 and V5R2.

Please note, however, that WebSphere Application Server is required to run a Web application generated by this tool. For more information on current versions of WebSphere Application Server, visit the Web site referenced in the "Hotlinks" section of this course.



WebFacing I/O

- DSPF DDS source member(s)
- Conversion style
  - Style of generated pages (background, layout, etc
- *PGM invocation information
  - How to invoke your 5250 application
- Job signon information
  - Prompt user at runtime or supply hardcoded user & password

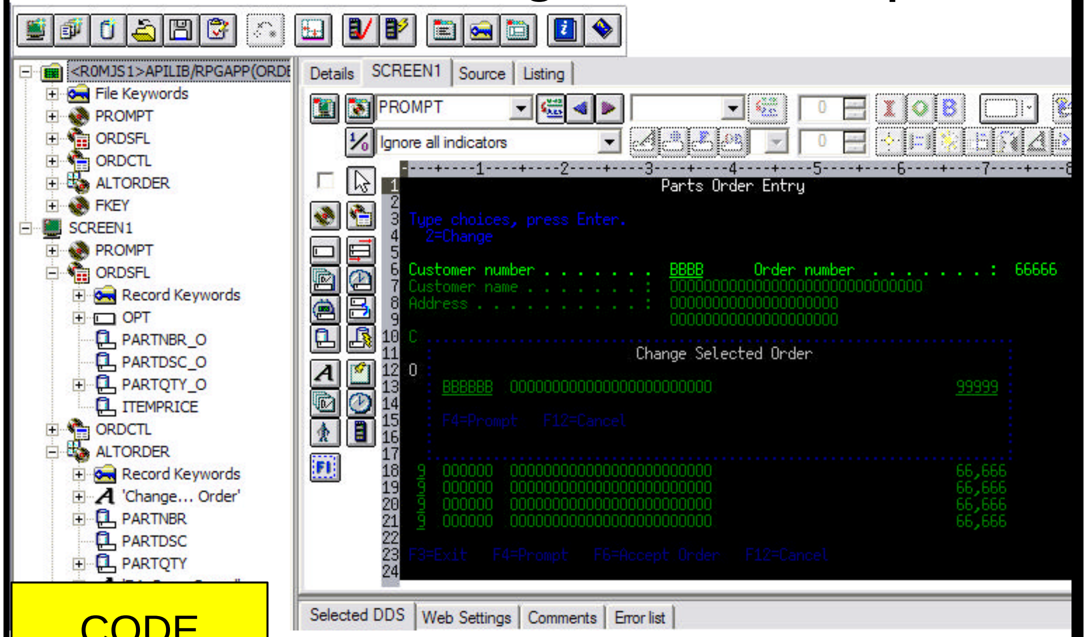
- Invocation Web page (index.html)
  - With link to start your Web-ized application
- JavaServer Pages
  - Couple per converted DSPF record format
- JavaBeans
  - Three per converted DSPF record format (input, output)

WebFacing I/O

The input to the IBM WebFacing Tool for iSeries development-time tool is the display file DDS source member and the conversion style. Additionally, you will need to specify how to invoke the 5250 application and how the user should be prompted for sign-on. There are two sign-on choices—prompt the user at runtime or hardcode the user ID and password at development time.

The output will consist of several items. First of all, it produces JSPs for actual user interfaces and secondly, it produces JavaBeans to hold the data. The data for the JavaBeans is the data passed from the RPG program and the data returned to the RPG program from the user interface. Optionally, this tool also generates the Web invocation page that users see when starting the Web-faced application. The invocation Web page provides a link to initiate the program that was “Web-faced.” Users click on the link for the program they want to use and the program begins. The expectation is that you, as the developer, can copy this link to an initial page of your own design.

IBM WebFacing Tool Example



```
                              Parts Order Entry

Type choices, press Enter.
  2=Change

Customer number . . . . . . . .  BBBB       Order number  . . . . . . . :  66666
Customer name . . . . . . . . :  000000000000000000000000000000
Address . . . . . . . . . . . :  00000000000000000000
                                 00000000000000000000
C  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   :                       Change Selected Order                             :
0  :                                                                          :
   :  BBBBBB  000000000000000000000000                                99999  :
   :                                                                          :
   :  F4=Prompt   F12=Cancel                                                  :
   :                                                                          :
   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 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   F12=Cancel
```

CODE Designer

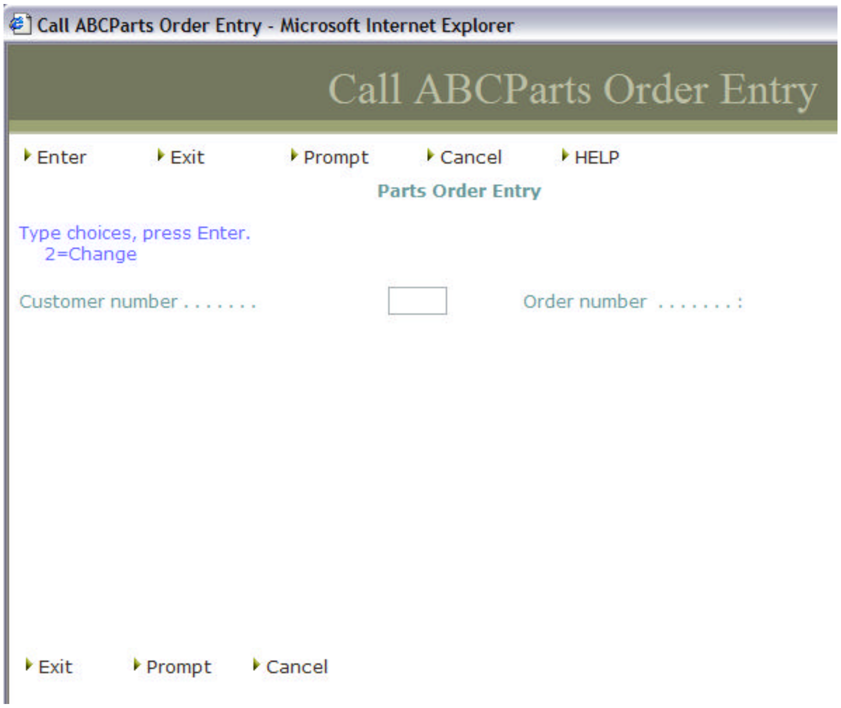
IBM WebFacing Tool for iSeries Example

Let’s contrast how a 5250-type screen interface might be transformed through the use of IBM WebFacing Tool for iSeries.

First, notice the screen capture shown here of a green screen. This is shown in CODE Designer, the SDA-follow-on tool that is also part of WebSphere Development Studio Client for iSeries.

The screen is part of a simple 5250 application built to illustrate this example.

IBM WebFacing Tool for iSeries Format


Call ABCParts Order Entry - Microsoft Internet Explorer
Call ABCParts Order Entry
Enter
Exit
Prompt
Cancel
HELP
Exit
Prompt
Cancel

Parts Order Entry

Type choices, press Enter.
2=Change

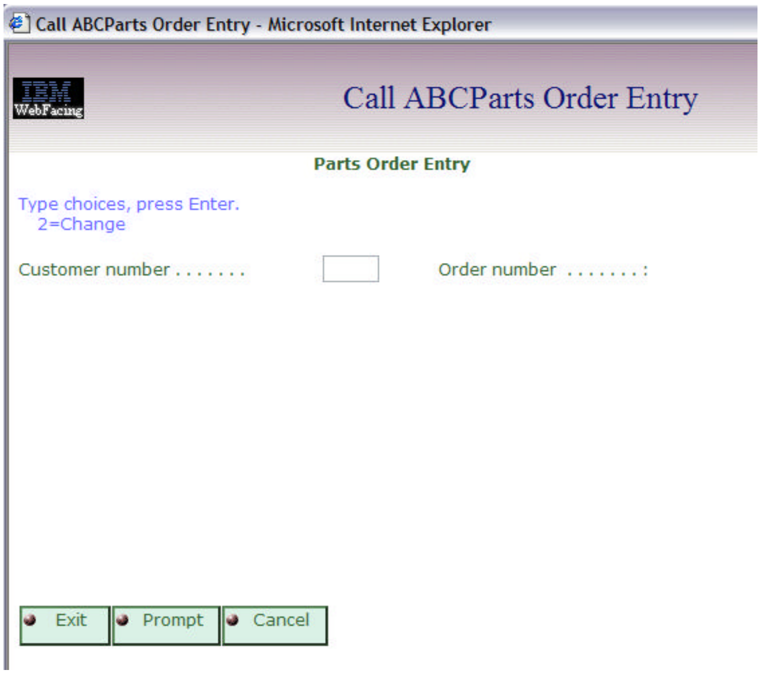
Customer number . . . .

Order number . . . . :

IBM WebFacing Tool for iSeries Format

Here is our previous example as it has been “Web-faced” with the “Sage” visual style. Notice that you only see the function keys that are actually enabled for the record formats that make up this screen. With other Web-facing solutions—because the 5250 datastream does not capture which function keys are enabled—all 24 function keys (that might potentially have been programmed for use) are typically shown in Web-enabled applications. This example was done on WebSphere Development Studio Client 5.0 Advanced using the IBM WebFacing Tool for iSeries—absolutely no improvements have been made to the resulting output.

Another Style . . . TextButtons



Call ABCParts Order Entry

Parts Order Entry

Type choices, press Enter.
2=Change

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

Another Web-facing Style—TextButtons

Here is another example of the various styles that can be selected when using IBM WebFacing Tool for iSeries. Here, we simply reconverted—but chose the “TextButtons” style—and again nothing has been done to improve the visual results. IBM supplies a number of styles with the tool. Further, a wizard is provided for creating your own styles. Or, existing cascading style sheets can be used.

There are two ways you can improve the looks of a “Web-faced” screen—one is utilized prior to conversion, while the other is an after-conversion process. For example, when designing a Web screen, you may not want a user to see the date, system name, or user ID. So, the tool offers a way of non-displaying fields for the Web version, even though these fields are displayed in the 5250 version. The easy method of doing this is to use CODE Designer and select the fields to be hidden. A Web Setting property sheet (per field) is used to specify options that only affect the conversion tool—options such as: “Visibility,” “Appearance and text,” or “Create graphic.” These options are stored as DDS comments in the source and picked up by the conversion tool.

The alternative would be to edit the generated JSP source file and delete these fields. However, that puts an onus on you to merge the changes back in after every subsequent reconversion.

IBM WebFacing Tool for iSeries Pros and Cons

Pros:

- Application no longer runs as an interactive job on new hardware
  - No changes needed to program
- Interactive *PGM can run either 5250 or "WebFaced" UI (user interface
  - No changes need to program
- Conversion done at development time
  - Runtime is fast and efficient
    - No 5250 datastream is ever created
  - Means generated JSP output can be enhanced... 'til cows come home' before deploying into production
- Cheap! Free runtime. Tooling is part of WebSphere Development Tools

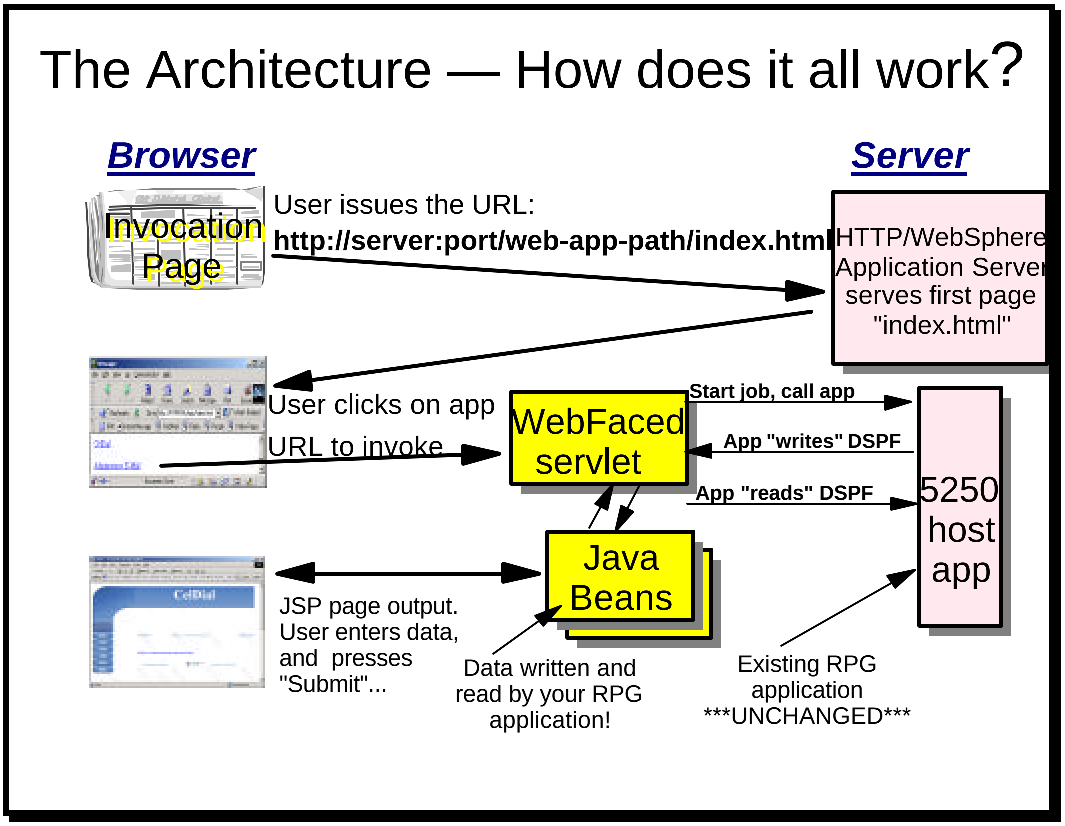
Cons:

- Must have access to DDS source
- Must convert EVERY screen
  - No support for UIM or system screens or 3270 or ...

Pros and Cons

To summarize, the benefit of using the IBM WebFacing Tool for iSeries is that users can run interactive programs in either a 5250 or “Web-faced” UI mode. The application no longer runs as an interactive job on new hardware, so it does not tie up as much system resource. Program changes are typically not required. Also, the conversion is done at development time so runtime can always be fast and efficient. With this tool, a datastream is never created and the generated JSP output can be enhanced until you are satisfied to deploy the application into production. The Web-facing effort is also inexpensive because there are no runtime costs and it is included, at no additional charge, in WebSphere Development Studio Client.

On the other hand, let’s discuss some of the drawbacks, should you use this tool. You must have access to the DDS source of the program, and you must convert EVERY screen in the program. IBM WebFacing Tool for iSeries only supports screens created with DDS and UIM help—there is no support at this time for other user interface management (UIM) panels, including: system screens, 3270 emulation, or any other type of user interface coding.



The Architecture — How does it all work?

Browser

Server

Invocation Page

User issues the URL:
http://server:port/web-app-path/index.html

HTTP/WebSphere Application Server serves first page "index.html"

User clicks on app URL to invoke

WebFaced servlet

Start job, call app

App "writes" DSPF

App "reads" DSPF

5250 host app

Java Beans

JSP page output. User enters data, and presses "Submit"...

Data written and read by your RPG application!

Existing RPG application ***UNCHANGED***

The Architecture—How does it all work?

Here is an overview of the IBM WebFacing Tool for iSeries architecture. To follow the application flow, start in the upper left corner.

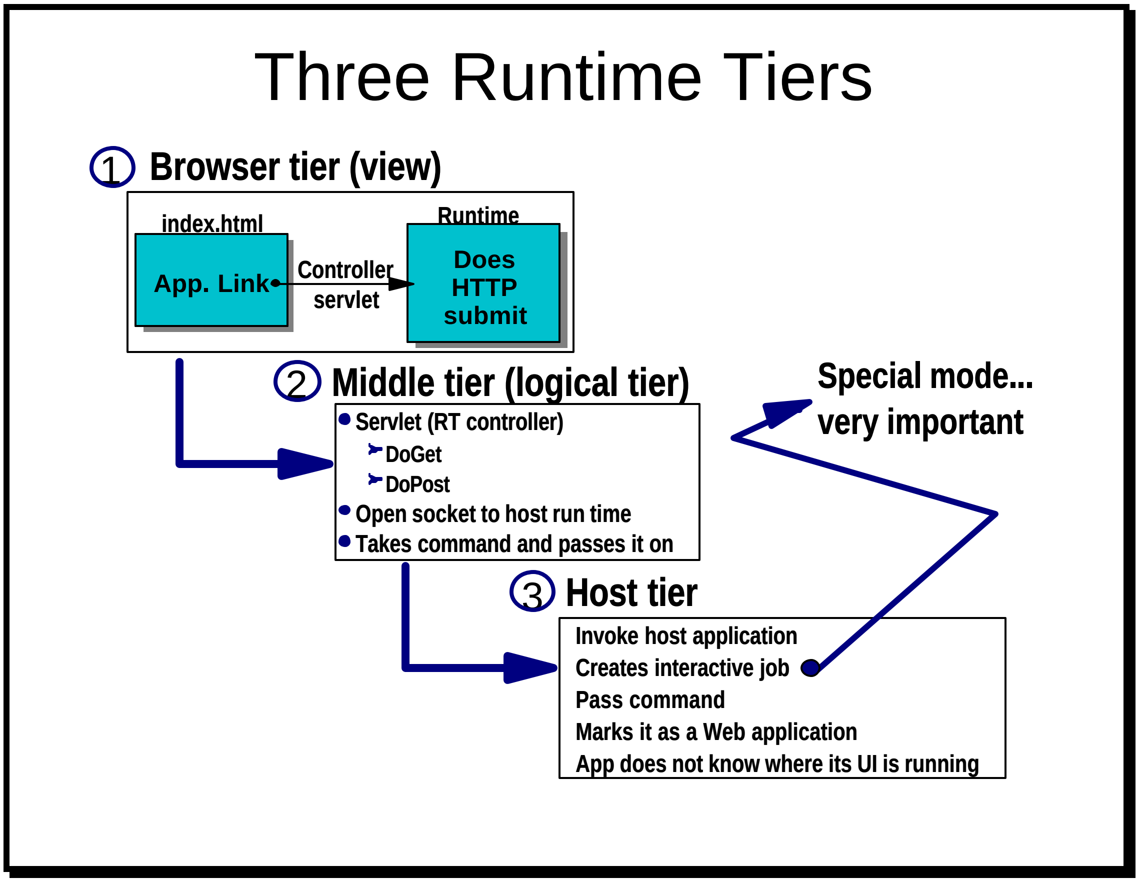
First, an invocation page is either generated by the tool (or has been created by you). On an invocation page, the user clicks on a link to start a Web-faced application. In this example, the URL associated with the link retrieves index.html. The index.html page is sent from the HTTP Server (or the WebSphere Application Server). The user (at his or her browser) receives the index.html page containing the “Web-faced” application links.

Next, the user clicks on the link that represents the needed application; and the URL associated with the link invokes the appropriate “Web-facing” servlet. The servlet starts the job and will either prompt the user for a user ID and password or will use the hardcoded password and user ID supplied by the programmer at development time.

The servlet starts the job the user requested by calling the appropriate 5250 program (RPG, COBOL. etc.). When the program starts, the iSeries server detects the invocation method and sets a switch allowing the workstation manager to identify the source of the application invocation—that is, either a Web browser or a 5250 display device type.

Once started by the servlet, the program executes to the point at which a screen is displayed (for example, in “write” or “execute” format). The servlet then sends the buffer to the workstation data management. At this point, since the user is on a Web browser, the data is sent to the IBM WebFacing Tool for iSeries runtime servlet, which converts the output program data into a Java Data Bean and sends it to the correct JSP for viewing in the user’s Web page. As mentioned previously, the tool’s servlet does not actually send the JSP to the user until the entire screen is built. So actually, the “writes” are queued and sent to the browser when the 5250 program issues a “read.”

When the user presses "Submit," the servlet regains control, thanks to a patented method used by the IBM WebFacing Tool for iSeries that blocks your RPG code. The servlet is actually waiting on a “read,” so it is already synched up when the user presses “Submit.” But what if the user does not press “Submit” for 30 minutes because of a sudden urge to get a cup of coffee? You can control how long to wait for user input with WebSphere Application Server. After 30 minutes, or whatever time you set, you can shut the application down.



Three Runtime Tiers

① **Browser tier (view)**

index.html

App. Link

Controller servlet

Runtime

Does HTTP submit

② **Middle tier (logical tier)**

- Servlet (RT controller)
  - DoGet
  - DoPost
- Open socket to host run time
- Takes command and passes it on

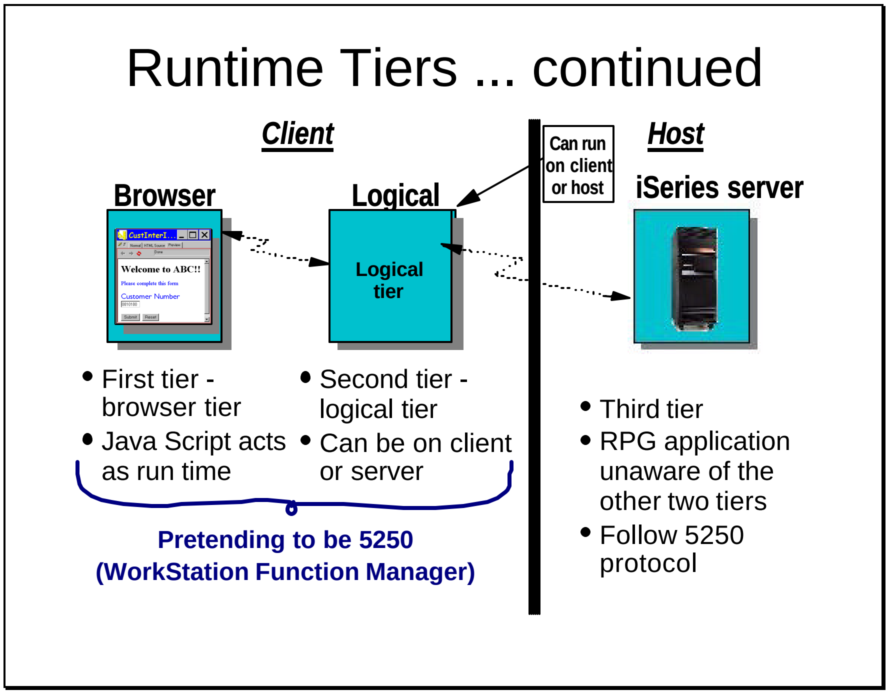
③ **Host tier**

Invoke host application
Creates interactive job
Pass command
Marks it as a Web application
App does not know where its UI is running

Special mode... very important

Three Runtime Tiers

There are three runtime tiers in a Web-faced application. First is the view, which is represented on this chart by the browser tier. Next is the middle tier (or logical tier) where the runtime servlet is managing the JSPs and JavaBeans. The last tier (the host tier) is where the 5250 application thinks it is still talking to a green screen.



Runtime Tiers ... continued

Client

Host

Browser

Logical

Can run on client or host

iSeries server

Logical tier

- First tier - browser tier
- Java Script acts as run time

- Second tier - logical tier
- Can be on client or server

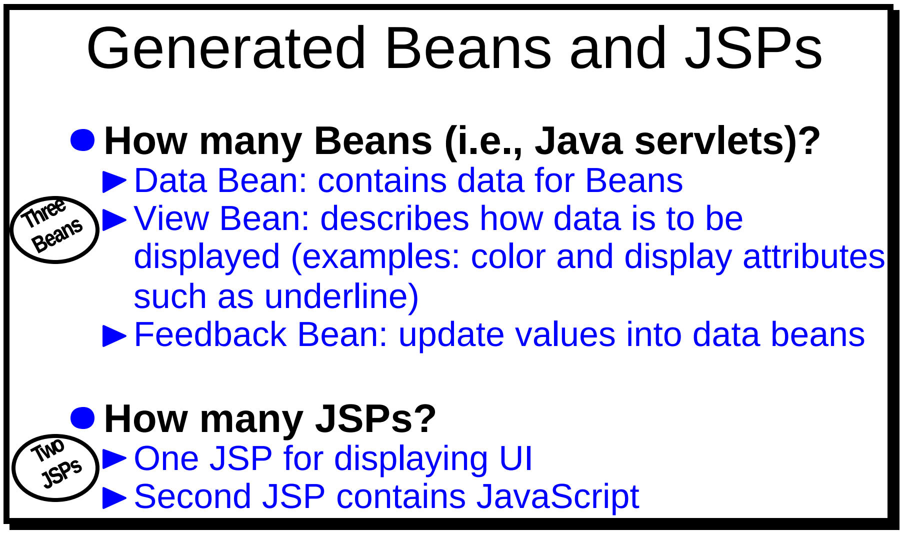
- Third tier
- RPG application unaware of the other two tiers
- Follow 5250 protocol

Pretending to be 5250
(WorkStation Function Manager)

Runtime Tier(s)—Continued

The middle tier—the "Web-faced" servlet plus the output of the development time transform (that is, the JSPs and JavaBeans)—can run on the same iSeries server as the RPG or COBOL application, or can alternatively be run on a different iSeries server. In fact, the middle tier does not have to be on an iSeries server at all. However, the middle tier must be on the same server as the WebSphere Application Server that is hosting the JSPs and servlets. There are three tiers in this environment, even though the user is running in a Web browser. The first tier is the browser. The second tier is WebSphere Application Server (and its prerequisite HTTP Server Powered by Apache). The third tier is the RPG or COBOL program. The Web-faced RPG or COBOL application runs wherever it normally executes.

It is up to you to decide whether the second and third tiers are located on the same iSeries server or on multiple systems. You may prefer to put the WebSphere Application Server and HTTP server combination on a separate iSeries server from your RPG or COBOL applications. In this case, you would front-end the iSeries server that runs the applications, coupled with another iSeries server that is running WebSphere only—especially if it is for public use. Remember, the IBM WebFacing Tool for iSeries does not have to be used for a public Web site; you might want to Web-face a back-office payroll application for a private intranet.



Generated Beans and JSPs

- **How many Beans (i.e., Java servlets)?**
  - Data Bean: contains data for Beans
  - View Bean: describes how data is to be displayed (examples: color and display attributes such as underline)
  - Feedback Bean: update values into data beans

Three Beans

- **How many JSPs?**
  - One JSP for displaying UI
  - Second JSP contains JavaScript

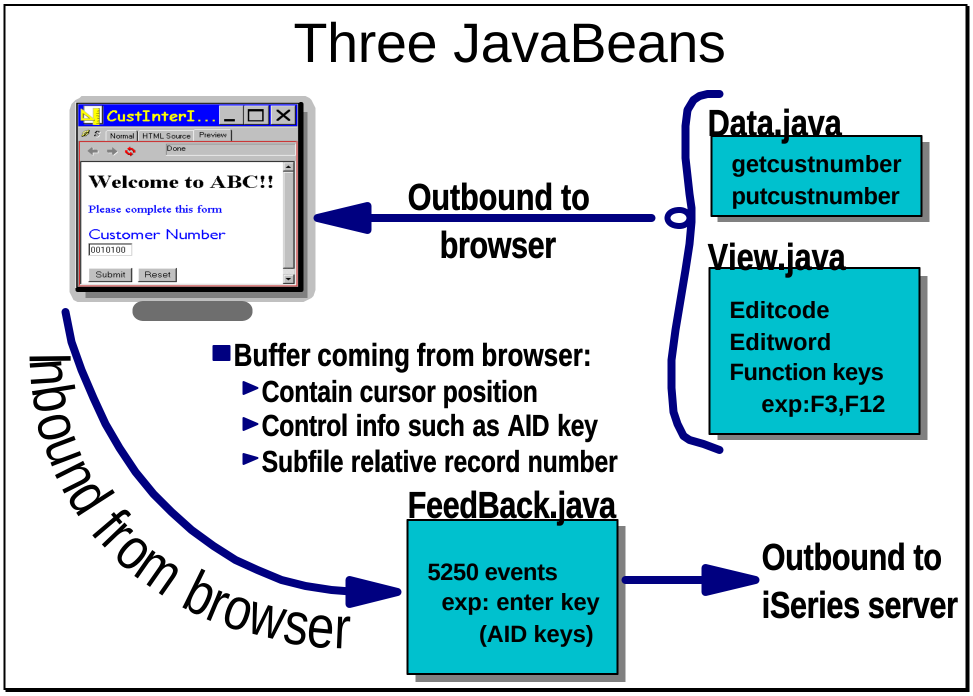
Two JSPs

Generated Beans and JSPs

How many JavaBeans are generated when the IBM WebFacing Tool for iSeries is used? For every record format, there will be three beans—a Data Bean, a View Bean, and a Feedback Bean. The Data Bean describes the data—the data in and out for that record. Of course, the View Bean describes how to display the data—for instance, edit code, validity checking keywords, color, and display attributes such as underline. There is no information in the View Bean about the raw data. Basically, each piece of data is pulled through the View Bean which decorates it with colors and attributes according to the indicator settings. Finally, a Feedback Bean tracks which key the user has pressed. The Feedback Bean is used to update values into Data Beans on a read request.

How many JSPs are created by this tool? Two—one JSP is used to display the user interface, including constants and holes for the dynamic data. The second JSP is used to contain JavaScript which deals with issues such as error checking. The tool will ensure that DDS validity-checking keywords (e.g., range) are honored. This type of error checking occurs on the client, not on the server. Because this JavaScript can be extremely lengthy and does not change while the application is running, the tool stores the JavaScript in a separate JSP that will come down to the browser only once and is cached automatically.

We will now go into more detail about each of the three types of beans: the Data Bean, the View Bean, and the Feedback Bean.



Three JavaBeans

Data.java

getcustnumber
putcustnumber

View.java

Editcode
Editword
Function keys
exp:F3,F12

Outbound to browser

- **Buffer coming from browser:**
  - **Contain cursor position**
  - **Control info such as AID key**
  - **Subfile relative record number**

Inbound from browser

FeedBack.java

5250 events
exp: enter key
(AID keys)

Outbound to iSeries server

Three JavaBeans

They say "a picture is worth a thousand words," so here is a picture of how three JavaBeans in an application work together to provide a "Web-faced" version of an RPG or COBOL program.

The Data Bean (Data.java) is used to store data coming in and out of the Web-faced application. The user (at a browser) inputs a customer number. This input actually issues a command to get information about a particular customer number (getcustnumber). The user input and key-pressed information is captured in the feedback bean and sent to the Workstation Data Manager which passes it on to the application. The existing application logic reads the database, writes out the format with the customer data, and does a read for the next request. When the customer data is returned to the servlet, the Data Bean and View Bean are used to insert that data into the generated JSP for that record format. Then the fully-realized Web page is sent to the browser. The Data Bean contains the customer data and the View Bean contains instructions on how to display that data. The JSP pulls the data through the View Bean and inserts the decorated results into the Web page.

The Feedback Bean (FeedBack.java) passes data and status information from the browser to the WorkStation Data Manager. For example, when the user keys in a customer number and presses the "Submit" button, the Feedback Bean updates the Data Bean with the new customer number and passes the "function key pressed" information to the RPG or COBOL program as if the user had pressed "Enter" on a 5250 terminal.

Additional Comments

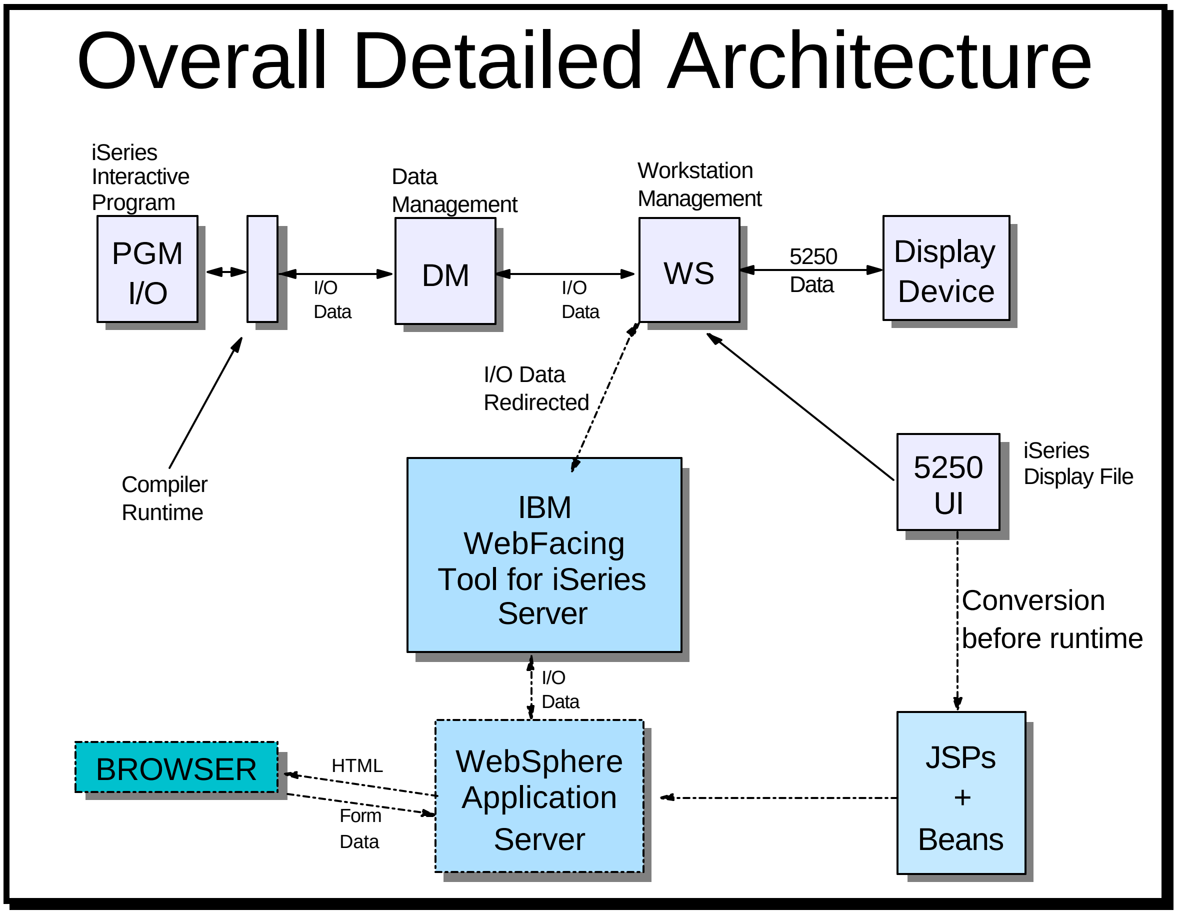
- **Two kinds of data in this application model**
  - Application data
    - **Coming out**
  - User data
    - **Typed by user**
- **Little change required to *PGM object**
  - Except for those DDS keywords not supported which impacts program
  - Except when decision is made to optimize flow for the Web
- **Multiple writes are merged into single JSP**

Additional Comments

A major point in this application model is that there are two kinds of data. There is application data coming out of your application and there is user data (typed by the user) coming in.

When you use the IBM WebFacing Tool for iSeries, no changes are required to the way the application handles input and output—the RPG or COBOL program is writing out via the usual RPG opcodes or COBOL statements it is receiving data back. The only situation that *requires* changes to the program is when the program uses DDS keywords that the tool does not support yet. You can also, optionally, modify the program to optimize flow for the Web. This is accomplished through the use of an API that detects when the user is on a browser. You may choose to leverage this API to offer an alternative screen flow optimized for the Web.

Another point is that multiple “writes” are merged into a single JSP. Many times, a programmer takes three record formats to create one screen. In this case, the program would write each format and then, on the last format, use either an “execute” format operation or a “write” immediately followed by a "read." This technique was a challenge for IBM development because the tool had to merge multiple formats into a single JSP so that only one screen would be sent to the client—but it has been handled, and we think you will appreciate this support.



Overall Detailed Architecture

Overall Detailed Architecture

Here is an overall detailed architecture picture. The top, horizontal flow depicts the traditional 5250 program which a company may want to continue using because it is an excellent, full-function business application that addresses what the company needs precisely. The lower portion of the graphic shows how the IBM WebFacing Tool for iSeries integrates this existing flow with new components.

The program continues to function as if writing to a 5250 terminal and OS/400 data management sends input and output data to workstation management. However, when the application is started from a browser session, a switch is set indicating that workstation management should redirect all data to the IBM WebFacing Tool for iSeries server instead of generating a 5250 datastream and directing it to a 5250 device.

The IBM WebFacing Tool for iSeries server works in concert with the WebSphere Application Server or the HTTP server, which in turn interfaces directly with the user's browser. As discussed previously, JSPs and JavaBeans are used to pass information back and forth between the user and the application.

A Web-faced application and its parts are generated at development time so they are all in place when invoked at runtime. Thus, runtime will be more efficient than with 5250 intercept products that create a Web interface on the fly.

2nd Mini Quiz

1. There are two ways you can improve the looks of a "Web-faced" screen.
   ☑ True—Correct. One method is utilized prior to conversion, while the other is an after-conversion process.
   ☐ False

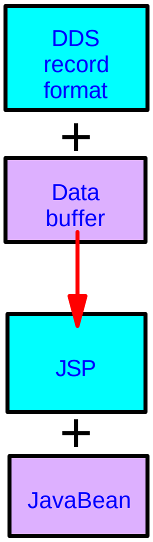
2. UI stands for:
   ☐ a. User Instance
   ☐ b. Unstable Interface
   ☑ c. User Interface—Correct. The benefits of using the IBM WebFacing Tool for iSeries are that users can run interactive programs in either a 5250 or “Web-faced” UI (user interface) mode.
   ☐ d. None of the above

3. The IBM WebFacing Tool for iSeries is strictly used for public Web sites:
   ☐ True
   ☑ False —Correct. This tool does not have to be used for a public Web site; you might want to Web-face a back-office payroll application for a private intranet.

DDS → JSP Conversion

IBM WebFacing Tool for iSeries Wizard

- Converts DDS to JSPs:
  - Select one or more members to convert
  - Creates 3 JavaBeans & 2 JSPs per record format
    - Beans holds data for record format
    - JSPs display/prompts data
  - Select from various look-and-feels
- Creates an "invocation page" per conversion:
  - Web page with links
  - Each link results in starting or reusing a job, running the user-supplied CL command which starts the application that uses the converted DSPFs



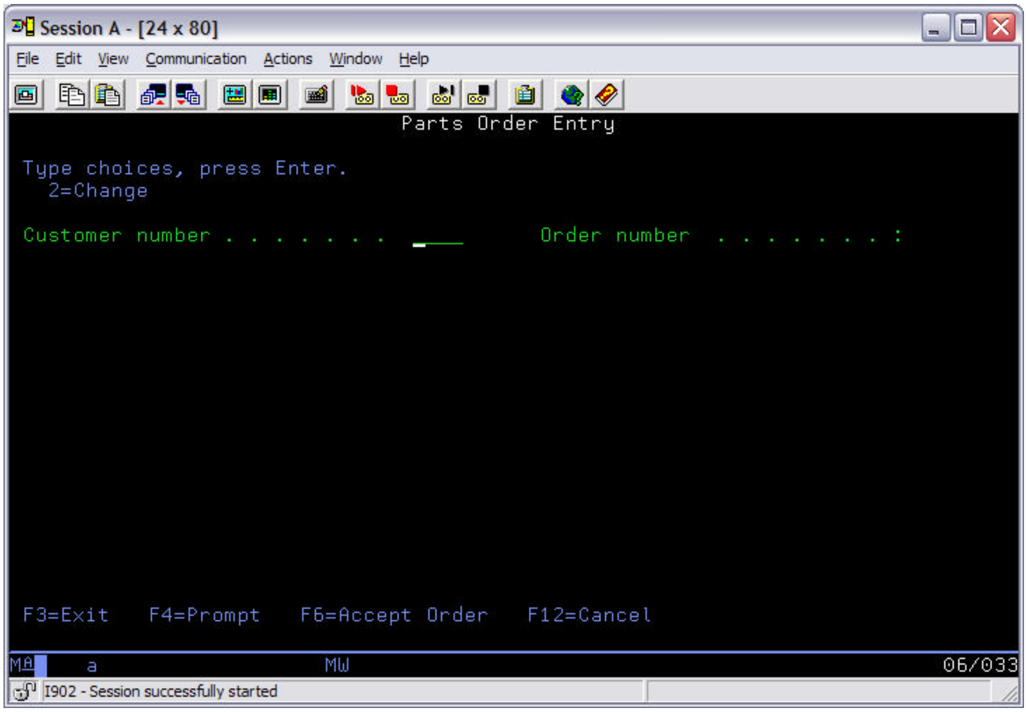
DDS- to-JSP Conversion

In this section, we are going to review converting DDS—a task where several tools are available to help.

One tool for this purpose, not surprisingly, is via the use of the IBM WebFacing Tool for iSeries wizard, which easily converts DDS to JSPs. First, you select one or more DDS members to convert, and then, for each record format, the wizard creates three JavaBeans and two JSPs, as described earlier in this course. The Beans hold dynamic data and formatting information for that data, for the record format, as well as function key and input data on a read request. The first JSP handles the constant part of the user interface as well as logic to insert the formatted dynamic data. The second JSP holds cache-able JavaScript™ logic for error checking the user input. You can select a presentation style for the JSP that displays the UI. As already mentioned, many are supplied by IBM, and the existing styles can be modified to create your own styles.

The IBM WebFacing Tool for iSeries wizard also generates an invocation page for each conversion. An invocation page—index.html—is a Web page with links. Selecting a link on the invocation page runs the associated program or command to start the application that was converted using this wizard. You would typically embellish this page or copy the links to a page of their own authoring.

First Screen (Full Example)



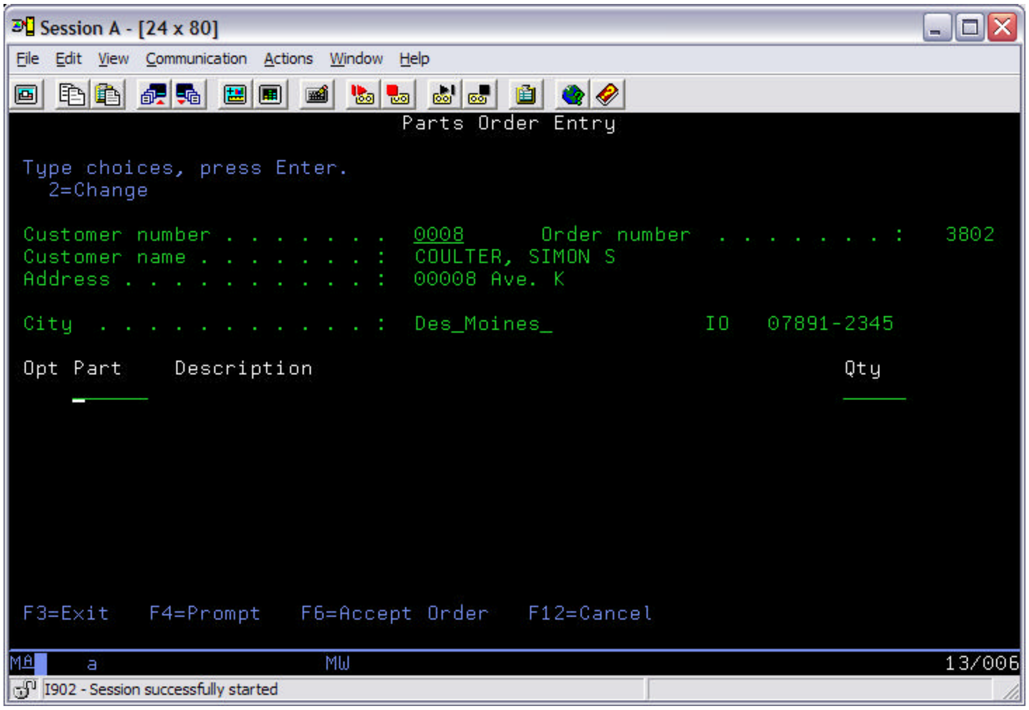
```
                              Parts Order Entry

 Type choices, press Enter.
   2=Change

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












 F3=Exit   F4=Prompt   F6=Accept Order   F12=Cancel
MA   a                 MW                                        06/033
```

I902 - Session successfully started

First Screen—Full Example

Here is a 5250 application that we are using as an example program that needs to be Web-faced. This first screen depicts a typical, small slice of an application. The program prompts for a customer number. If the user types in a customer number and presses ENTER, the Customer Detail Screen is displayed.

Customer Detail



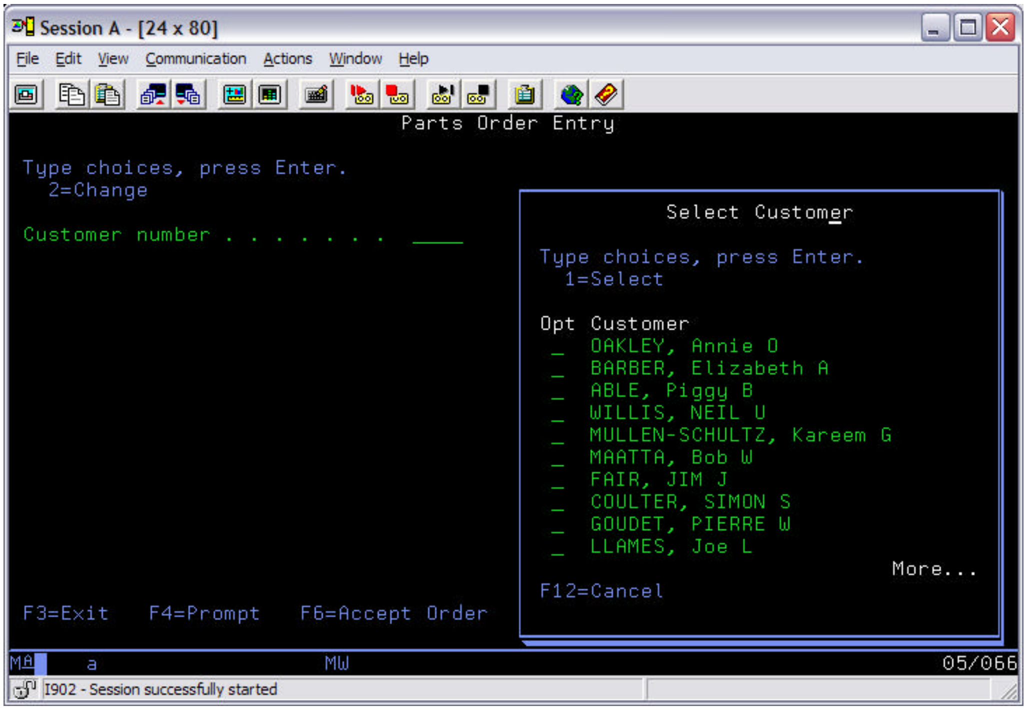
```
                                   Parts Order Entry

 Type choices, press Enter.
   2=Change

 Customer number . . . . . . .  0008      Order number . . . . . . . :   3802
 Customer name . . . . . . . :  COULTER, SIMON S
 Address . . . . . . . . . . :  00008 Ave. K

 City  . . . . . . . . . . . :  Des_Moines_            IO   07891-2345

 Opt Part     Description                                              Qty
     ______                                                            _____






 F3=Exit    F4=Prompt    F6=Accept Order    F12=Cancel

MA     a                        MW                                       13/006
```

I902 - Session successfully started

Customer Detail

Customer information for the selected customer number is returned, as shown here.

Customer List



```
                            Parts Order Entry

 Type choices, press Enter.
   2=Change                              Select Customer

 Customer number . . . . . . .  _____   Type choices, press Enter.
                                          1=Select

                                        Opt  Customer
                                         _   OAKLEY, Annie O
                                         _   BARBER, Elizabeth A
                                         _   ABLE, Piggy B
                                         _   WILLIS, NEIL U
                                         _   MULLEN-SCHULTZ, Kareem G
                                         _   MAATTA, Bob W
                                         _   FAIR, JIM J
                                         _   COULTER, SIMON S
                                         _   GOUDET, PIERRE W
                                         _   LLAMES, Joe L
                                                              More...
                                        F12=Cancel
 F3=Exit   F4=Prompt   F6=Accept Order

MA    a                    MW                                   05/066
I902 - Session successfully started
```

Customer List

If the user pressed F4 to see a list of customers—instead of typing in a specific customer number—then a selection pop-up window, similar to the one shown on this screen, is displayed.

What are the steps to using the IBM WebFacing Tool for iSeries?

- **Start the Integrated Development Environment (new IDE)**
- **Import DDS source from the server**
- **Convert the DDS source**
- **Generate publishing information**
- **Ensure WebSphere Application Server instance is up and running**
- **Deploy the files (FTP, net use, ...etc.)**
- **Invoke the first page**
  (Invocation page = index.html)

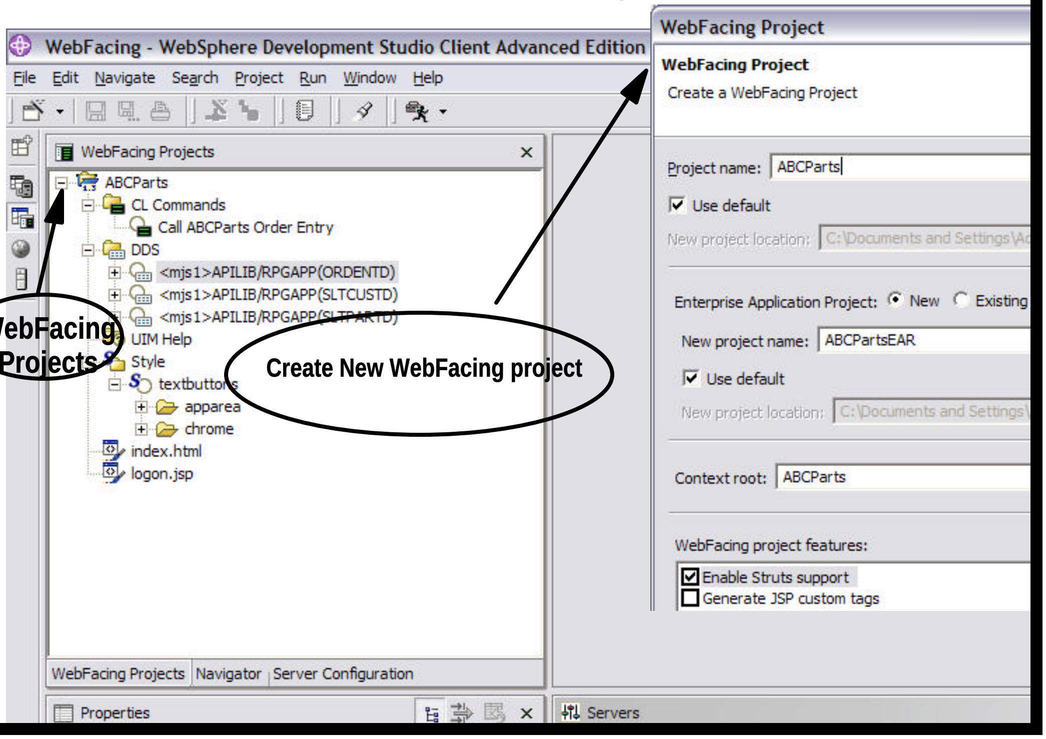
Steps for Using the Tool

You can now start WebSphere Development Studio Client for iSeries and open the Web-facing perspective. Next, run the IBM WebFacing Tool for iSeries wizard to create a new project which prompts for the DDS source to be converted. You can simply create the project first and do the conversion later, or do the conversion as the last step of creating the project. The project can then be tested in the built-in WebSphere Application Server test environment.

An important part of this whole process is the successful installation and running of WebSphere Application Server on the iSeries server. Before the files can be deployed, you need to ensure that the WebSphere Application Server instance is up and running, and that a Web application also running.

Once WebSphere Application Server is running and the Web application is configured, the files created by the wizard (JSPs and JavaBeans) can be deployed and the first page can be invoked from a Web browser.

New IBM Web-facing Tool IDE


WebFacing - WebSphere Development Studio Client Advanced Edition
File Edit Navigate Search Project Run Window Help
WebFacing Projects
ABCParts
CL Commands
Call ABCParts Order Entry
DDS
<mjs1>APILIB/RPGAPP(ORDENTD)
<mjs1>APILIB/RPGAPP(SLTCUSTD)
<mjs1>APILIB/RPGAPP(SLTPARTD)
UIM Help
Style
textbuttons
apparea
chrome
index.html
logon.jsp
WebFacing Projects
Navigator
Server Configuration
Properties
Servers
WebFacing Projects
Create New WebFacing project
WebFacing Project

WebFacing Project

Create a WebFacing Project

Project name: ABCParts

☑ Use default

New project location: C:\Documents and Settings\

Enterprise Application Project: ● New ○ Existing

New project name: ABCPartsEAR

☑ Use default

New project location: C:\Documents and Settings\

Context root: ABCParts

WebFacing project features:

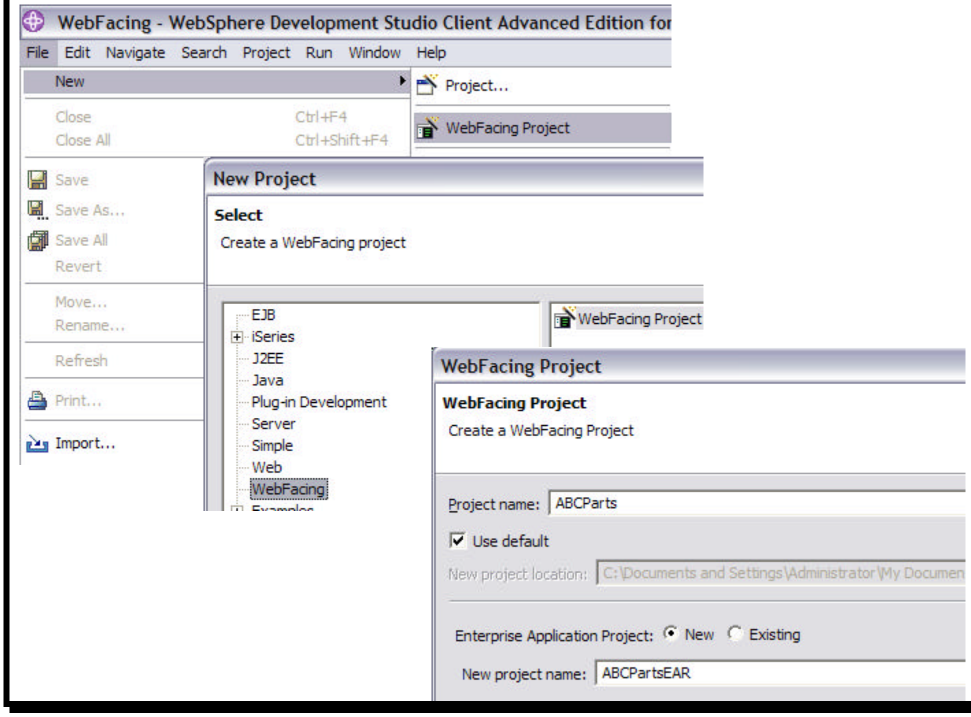
☑ Enable Struts support

☐ Generate JSP custom tags

New Web-facing Tool IDE

Here are some of the screens you might see when Web-facing an application. You begin by responding to the Quick Start menu of the IBM WebFacing Tool for iSeries toolbox shown on the right in this graphic. If you choose to go to the desktop, then you are taken to the screen shown on the left side of this graphic.

Create a New Project!



Select

Create a WebFacing project

WebFacing Project

Create a WebFacing Project

Project name: ABCParts

☑ Use default

New project location: C:\Documents and Settings\Administrator\My Documen

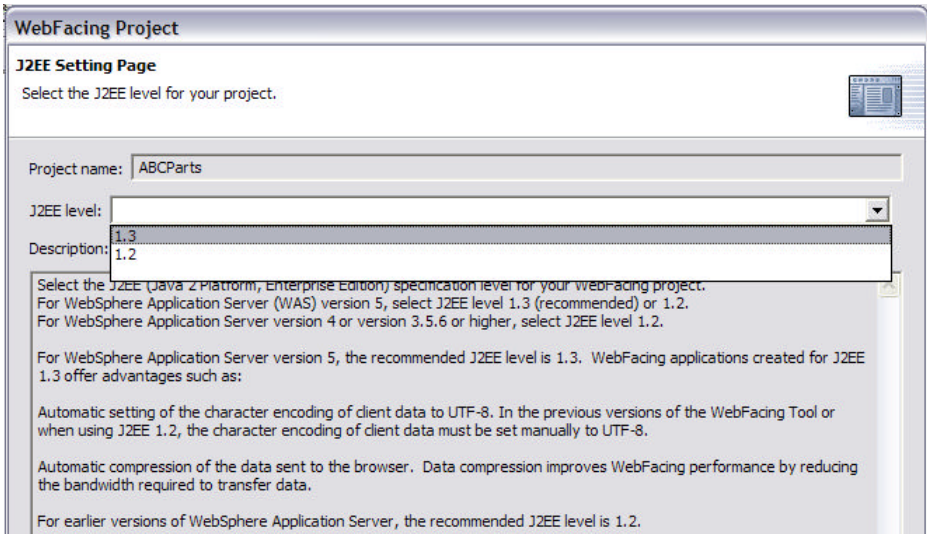
Enterprise Application Project: ● New ○ Existing

New project name: ABCPartsEAR

Create a New Project!

If you decide to create a new Web-facing project, then the wizard prompts shown on this graphic will be invoked. Not surprisingly, the first thing you are prompted for is the name of the project.

Pick a J2EE level



Select the J2EE level for your project.

Project name: ABCParts

J2EE level:

Description:

Select the J2EE (Java 2 Platform, Enterprise Edition) specification level for your WebFacing project.
For WebSphere Application Server (WAS) version 5, select J2EE level 1.3 (recommended) or 1.2.
For WebSphere Application Server version 4 or version 3.5.6 or higher, select J2EE level 1.2.

For WebSphere Application Server version 5, the recommended J2EE level is 1.3. WebFacing applications created for J2EE 1.3 offer advantages such as:

Automatic setting of the character encoding of client data to UTF-8. In the previous versions of the WebFacing Tool or when using J2EE 1.2, the character encoding of client data must be set manually to UTF-8.

Automatic compression of the data sent to the browser. Data compression improves WebFacing performance by reducing the bandwidth required to transfer data.

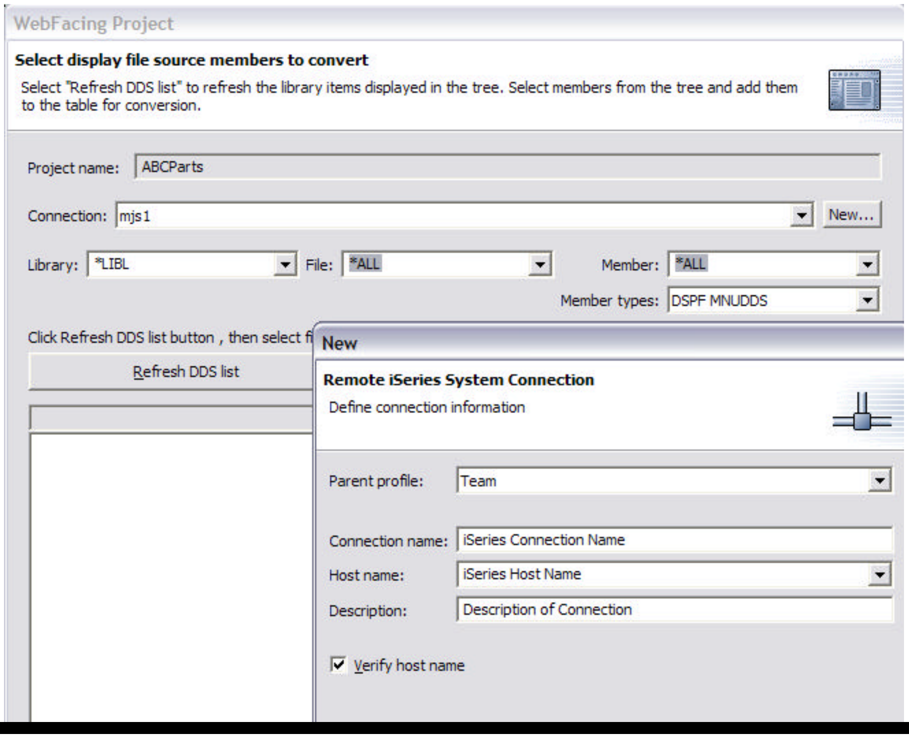
For earlier versions of WebSphere Application Server, the recommended J2EE level is 1.2.

Pick A J2EE Level

On this screen, we chose between Java 2 Enterprise Edition™ (J2EE™) version 1.2 and 1.3.

J2EE 1.2 is required for versions of WebSphere Application Server Version 5. By using J2EE 1.3 with WebSphere Application Server 5 or later, UTF-8 no longer needs to be set in the application server. For this example, we will be selecting J2EE 1.3.

Pick a Server



WebFacing 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: ABCParts

Connection: mjs1

Library: *LIBL

File: *ALL

Member: *ALL

Member types: DSPF MNUDDS

Click Refresh DDS list button , then select fi

New

Remote iSeries System Connection

Define connection information

Parent profile: Team

Connection name: iSeries Connection Name

Host name: iSeries Host Name

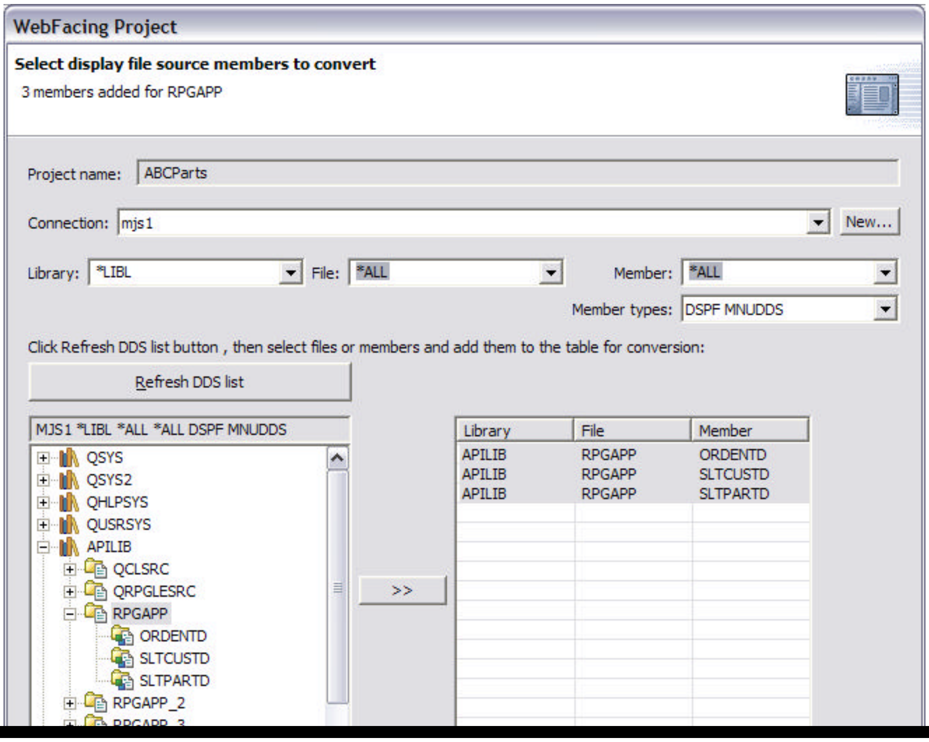
Description: Description of Connection

☑ Verify host name

Pick a Server

The next step is to select the iSeries server where the DDS and UIM help is stored and to sign on to that server. Select the DDS and UIM help source members that you want to convert. On this next display, the libraries and library list are shown.

Select DDS File Source



WebFacing Project

Select display file source members to convert

3 members added for RPGAPP

Project name: ABCParts

Connection: mjs1

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:

MJS1 *LIBL *ALL *ALL DSPF MNUDDS

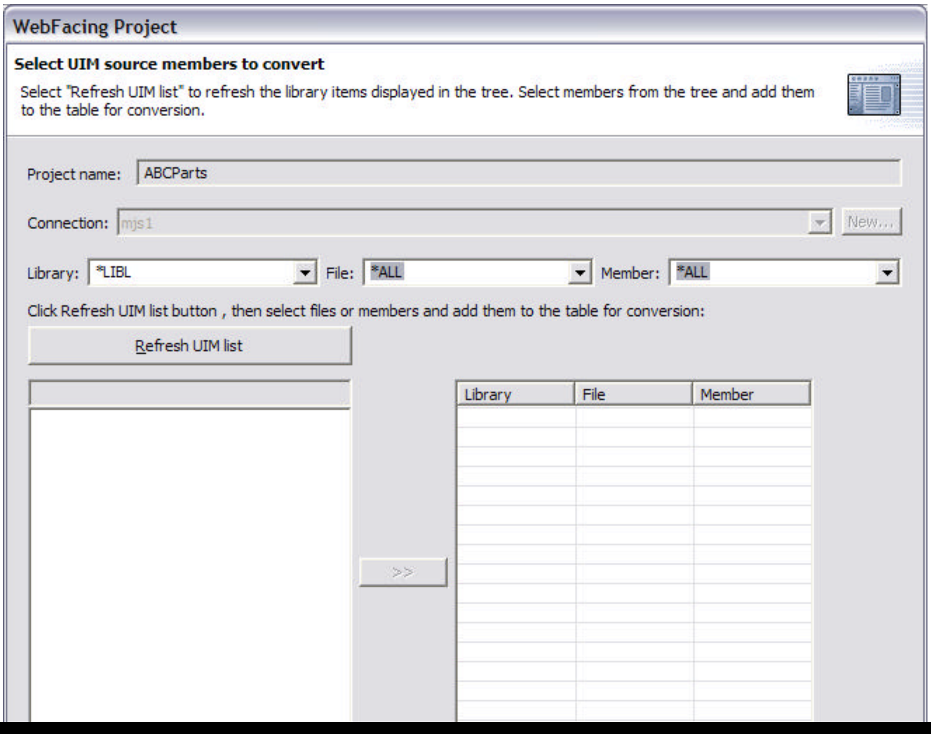
- QSYS
- QSYS2
- QHLPSYS
- QUSRSYS
- APILIB
  - QCLSRC
  - QRPGLESRC
  - RPGAPP
    - ORDENTD
    - SLTCUSTD
    - SLTPARTD
  - RPGAPP_2

Library	File	Member
APILIB	RPGAPP	ORDENTD
APILIB	RPGAPP	SLTCUSTD
APILIB	RPGAPP	SLTPARTD

Select DDS File Source

The source selection screen is populated based on your requested library list. To change the display, type in a different library name on this display and the list will be repopulated. You can expand a source file and list the source members. Selecting a source member in the left-hand list and pressing the arrow between the two lists adds the source member to the right-hand list, thus indicating the source members to be converted.

Select UIM Source



WebFacing Project

Select UIM source members to convert

Select "Refresh UIM 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: ABCParts

Connection: mjs1

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

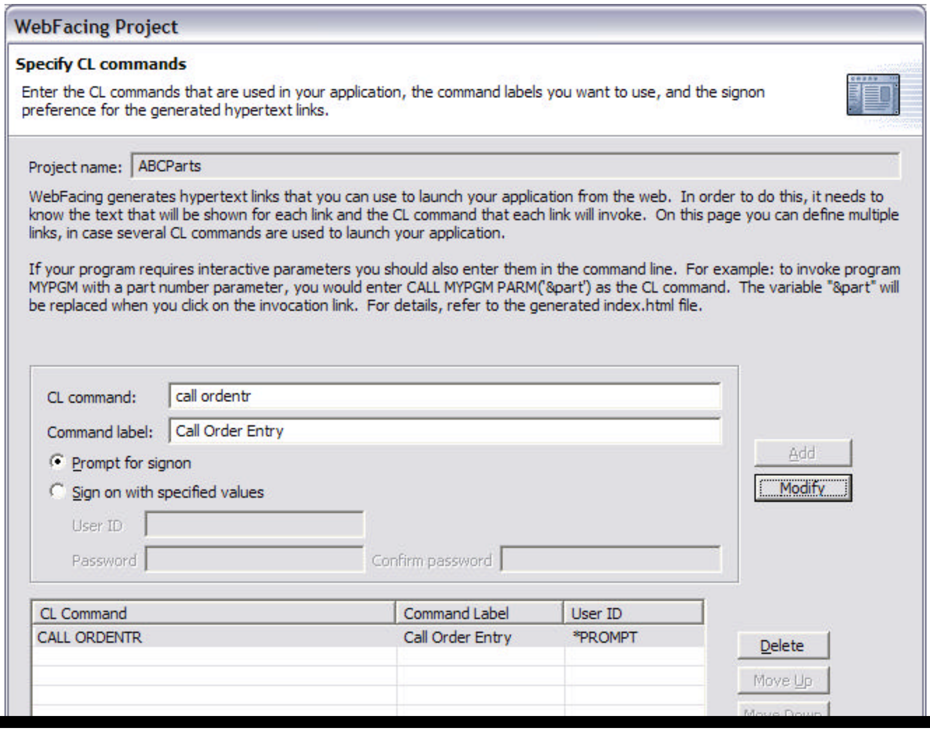
Click Refresh UIM list button , then select files or members and add them to the table for conversion:

Library	File	Member

Select UIM Source

As we just discussed, the source selection screen is populated based on your requested library list. To change the display, you can type in a different library name on this display and the list will be repopulated. Our example does not contain any UIM Help so none are selected.

CMD to Call Application



WebFacing 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: ABCParts

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 interactive parameters you should also enter them in the command line. 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.html file.

CL command: call ordentr

Command label: Call Order Entry

● Prompt for signon

○ Sign on with specified values

User ID

Password

Confirm password

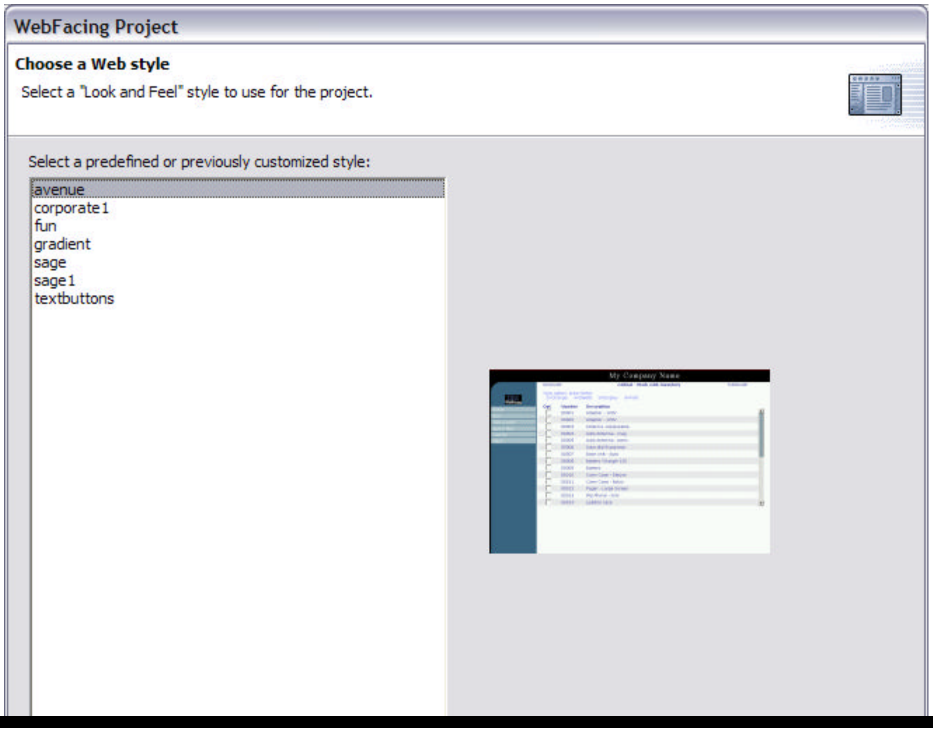
CL Command	Command Label	User ID
CALL ORDENTR	Call Order Entry	*PROMPT

CMD to Call Application

Next, type in the command used to invoke the application with all of its parameters. Multiple commands may be entered if you want to generate multiple links, each of which can call the application by using different parameters. As you can see, you have the ability to indicate exactly what should be generated in terms of the caption shown on the invocation page link.

A common mistake on this page is to forget to press the “Add” button after typing in the label and command.

Select Presentation Style



WebFacing Project

Choose a Web style

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

Select a predefined or previously customized style:

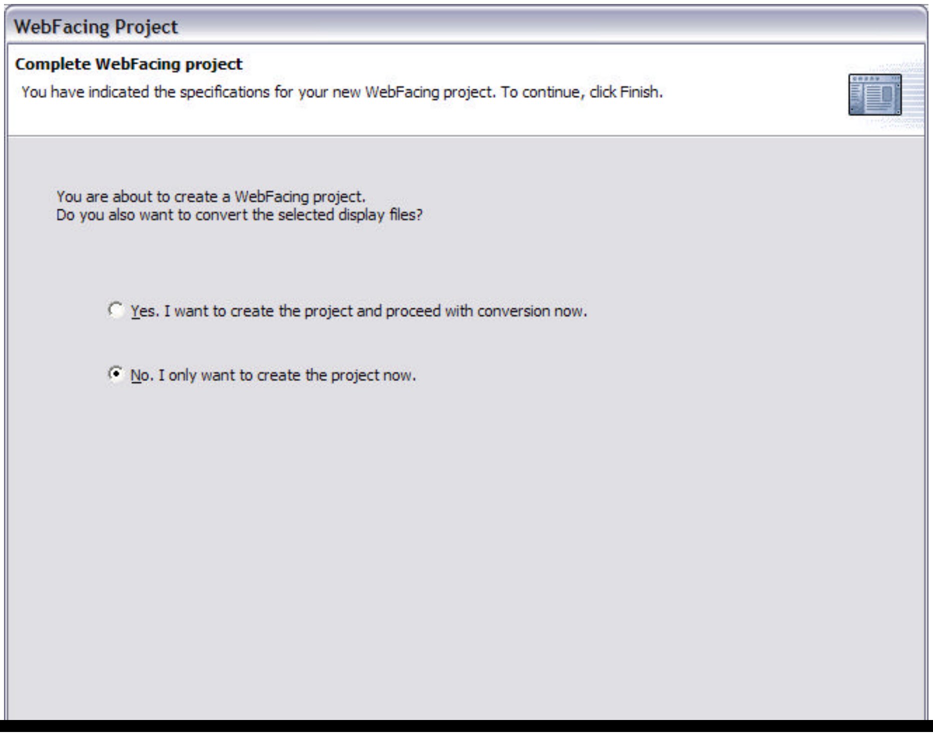
- avenue
- corporate1
- fun
- gradient
- sage
- sage1
- textbuttons

Select Presentation Style

Once you decide which members to convert, a prompt is displayed to select a presentation style that determines how the converted output should look.

You can also create your own styles by modifying an existing style. Existing Cascading Style Sheets can also be used to customize the look of the application.

Done!



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.

Done!

At this point, you can create the “WebFacing” project. The DDS can be converted immediately or you can choose to defer the conversion, in which case all the conversion information will be recorded. Once the project has been created, right-click on the project at any time and select “Convert” to actually create the Web-enabling elements—that is, the JSPs, the JavaBeans, and the HTML invocation page—from the saved information. As the DDS source evolves, it is expected that the project will be reconverted many times.

The conversion takes up to a couple of minutes per source member, and will add the library containing the source to the library list. But, if your referenced fields are in another library, please ensure that library is on your initial library list as specified by the job description specified in your user profile.

A report is generated that indicates the level of success of the conversion, and pointing out any keywords encountered that are not yet supported by the conversion tool. Some of these can be successfully ignored, while others will have a major impact. The report includes an Overview, Referenced Keywords, DSPF Conversion Log, and UIM Help Conversion Log.

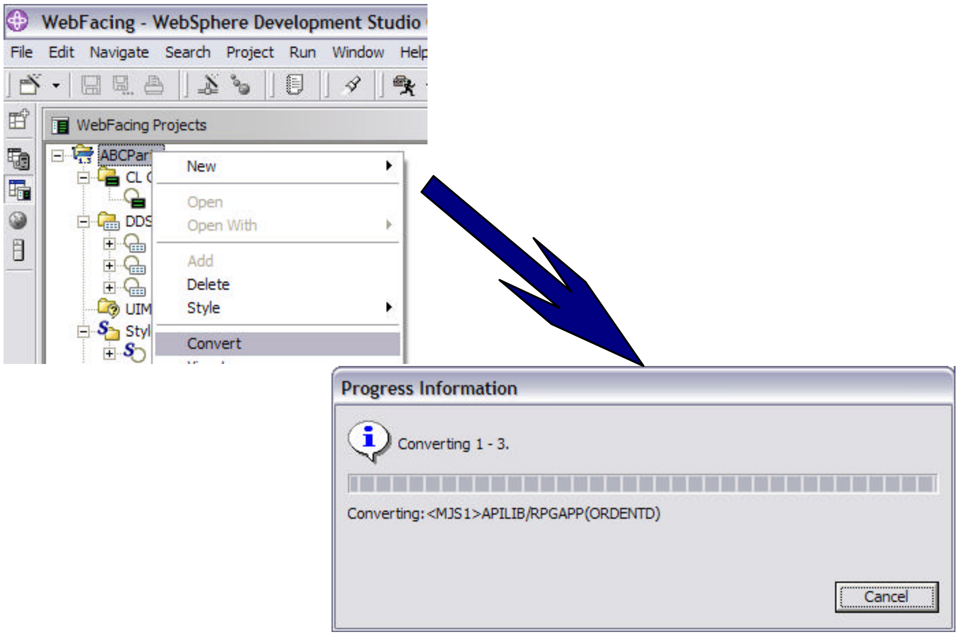
3rd Mini Quiz

1. An invocation page is a Web page with links.
   ☑ True—Correct. The IBM WebFacing Tool for iSeries wizard generates an invocation page for each conversion. Selecting a link on the invocation page runs the associated CL command to start the application that was converted using this wizard. The IBM WebFacing Tool for iSeries SmartGuide also creates an invocation page—index.html—for each conversion project.
   ☐ False

2. The successful installation and running of WebSphere Application Server is not a necessary step.
   ☐ True
   ☑ False—Correct. WebSphere Application Server is an important part of this whole process. Before the files can be deployed, you need to ensure that the WebSphere Application Server instance is up and running. In some installations, this task may take some time.

3. Programmers are not allowed to create their own styles.
   ☐ True
   ☑ False—Correct. Programmers can also create their own styles.

Convert DDS to JSPs



Converting 1 - 3.

Converting: <MJS1>APILIB/RPGAPP(ORDENTD)

Convert DDS to JSPs

Let's review some of the screens you might see when converting DDS to JSPs.

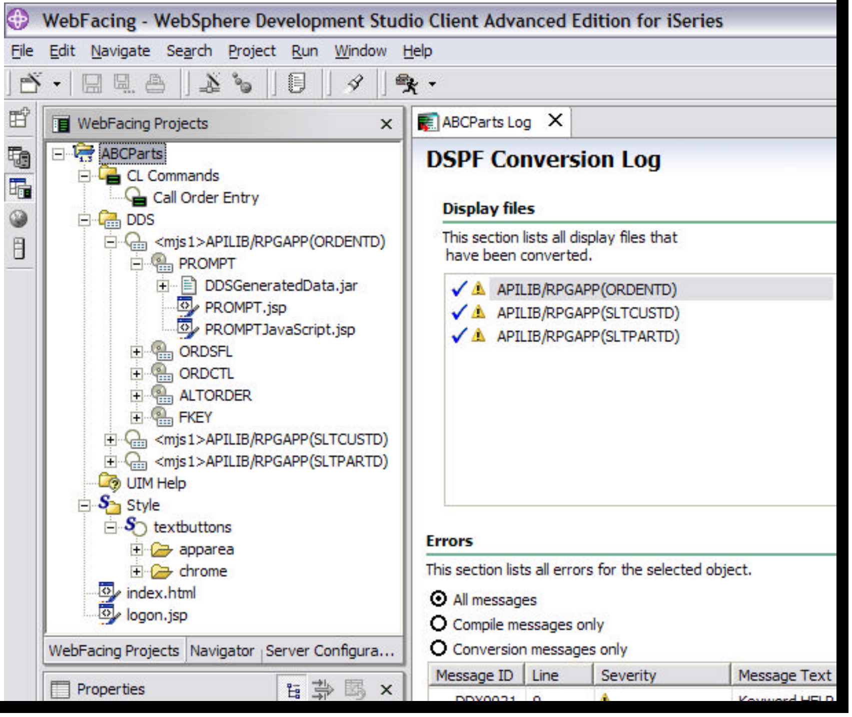
To do the actual conversion, you have to be signed onto the iSeries server to get the DDS source. Therefore, you will be prompted for the log-on, if not already signed on.

Everything else required for the conversion would have already been entered when you went through the IBM WebFacing Tool for iSeries SmartGuide. So, you simply wait for the conversion to complete.

Typically, a conversion takes less than a minute per display file source member. A report will be generated and shown in your Web browser, indicating any keywords found that are not yet supported by the conversion, and the severity of those missing keywords.

Please note the project will only record the names and locations of the DDS source members. It does not maintain a local copy of those members. The conversion tool will connect and retrieve the source as it runs.

Completed "IBM WebFacing Tool for iSeries" Project



DSPF Conversion Log

Display files

This section lists all display files that have been converted.

✓ ⚠ APILIB/RPGAPP(ORDENTD)
✓ ⚠ APILIB/RPGAPP(SLTCUSTD)
✓ ⚠ APILIB/RPGAPP(SLTPARTD)

Errors

This section lists all errors for the selected object.

● All messages
○ Compile messages only
○ Conversion messages only

Message ID	Line	Severity	Message Text

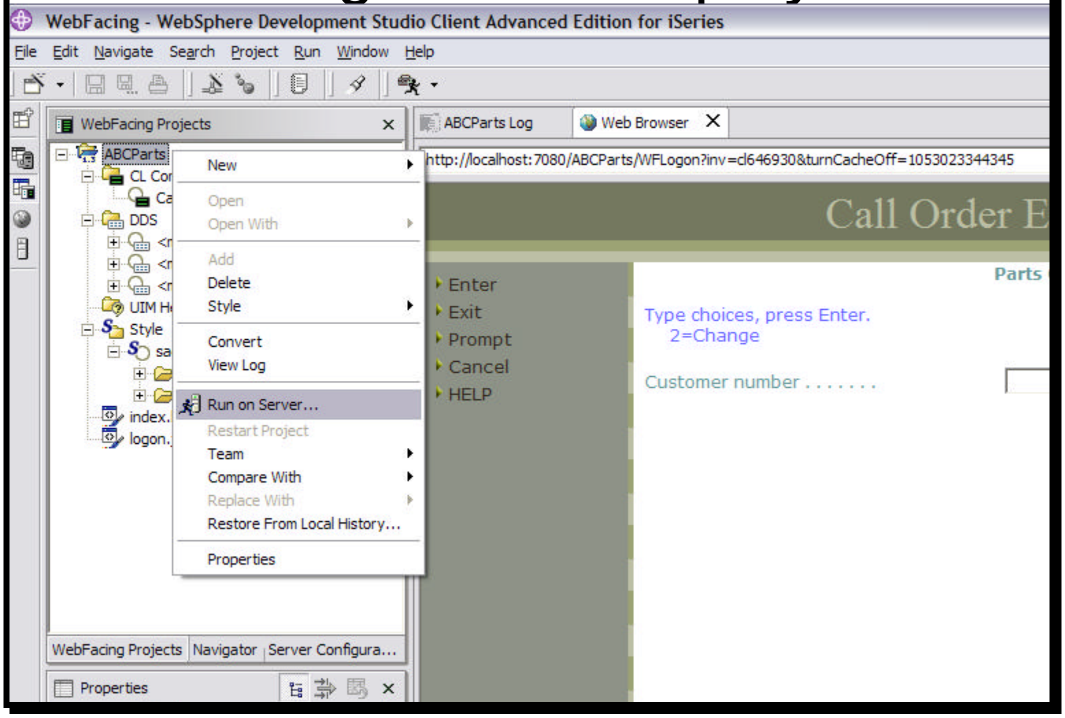
Right-click on any folder or file for actions

Completed "IBM WebFacing Tool for iSeries" Project

The conversion automatically generates the JSPs, the JavaBeans, the servlets, the publishing information, and the invocation Web page—everything you see encircled on this graphic. When the conversion is complete, all the output information will be displayed within the project, as shown on this “WebFacing Perspective.”

Your next step is to verify the exporting information.

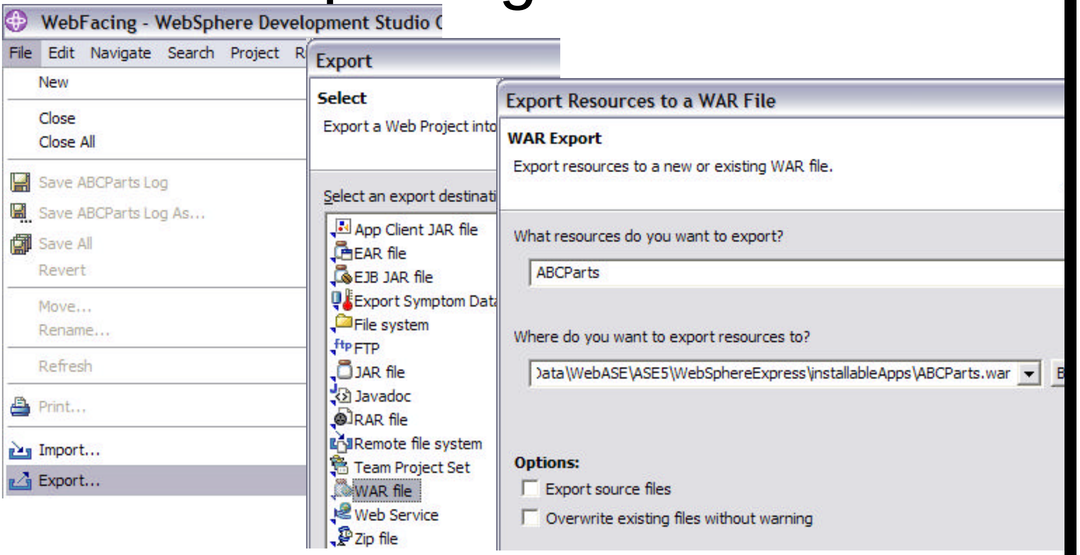
Testing before Deployment



Testing in WebSphere Development Studio Client before Deployment

Your Web-facing project can be tested quickly and easily inside WebSphere Development Studio Client for iSeries in the integrated WebSphere Application Server test environment. By right-clicking on your project and selecting "Run on Server…" you can select the version on which you want to test. This feature allows you to quickly test and update your WebFacing project until you are satisfied and ready to deploy to an iSeries server. For your project to work in the test environment, it needs to talk to the iSeries server running the green screen application and the WebFacing Runtime Server by issuing the following command: (STRTCPSVR SERVER(*WEBFACING))

Exporting Information



WAR Export

Export resources to a new or existing WAR file.

What resources do you want to export?

ABCParts

Where do you want to export resources to?

)ata\WebASE\ASE5\WebSphereExpress\installableApps\ABCParts.war

Options:

☐ Export source files

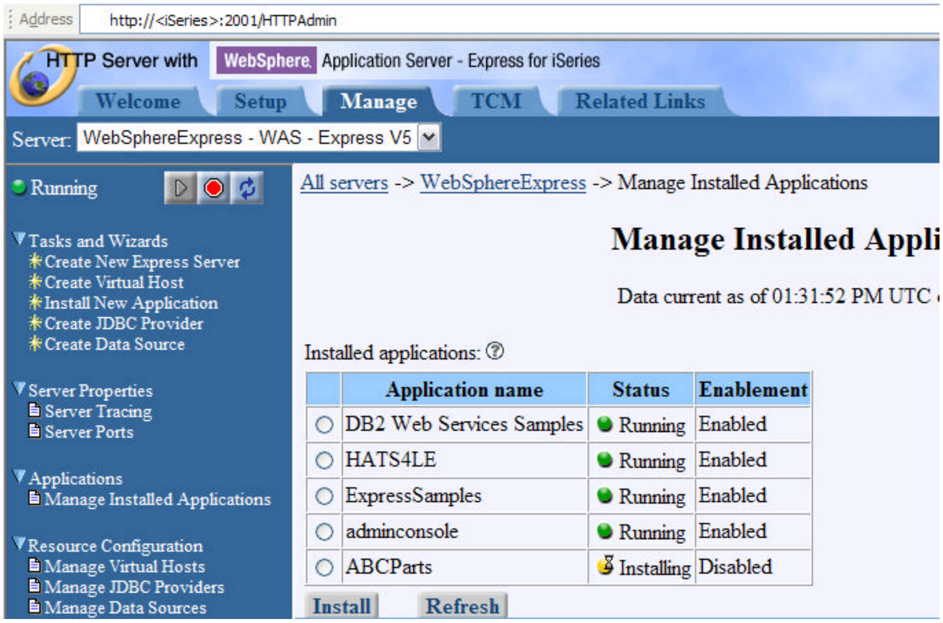
☐ Overwrite existing files without warning

Congratulations! You are done and ready to deploy to the server !

Exporting Information

To export your information, click “File” and select “Export…” from the menu. Once this is done, either select “WAR file” (Web Archive file) or “EAR file” [Enterprise (J2EE) Archive file] on the next window. Both will work with WebSphere Application Server. The EAR format contains more information including things like context root. In this example, we selected a WAR file for export. On the Export Resources to a WAR file window, select your Web-facing project and give a destination to which it will be exported. Here we exported to the installableApps directory in our WebSphere Application Server 5.0 - Express instance.

Installing into WebSphere Application Server



Address http://<iSeries>:2001/HTTPAdmin

Server: WebSphereExpress - WAS - Express V5

Data current as of 01:31:52 PM UTC

Installed applications:

	Application name	Status	Enablement
○	DB2 Web Services Samples	● Running	Enabled
○	HATS4LE	● Running	Enabled
○	ExpressSamples	● Running	Enabled
○	adminconsole	● Running	Enabled
○	ABCParts	Installing	Disabled

Deploy to Server into WebSphere Application Server

To publish the Web-faced application to an iSeries server, there needs to already be a configured instance of WebSphere Application Server. In this slide, we are showing the installation using WebSphere Application Server— Express V5.0 for iSeries. By clicking on “Install” under “Manage Installed Applications” and following the wizard, ABCParts can be installed easily–as long as you know where you exported the EAR or WAR file. Applications can also be installed using the Express Console which is launched from the same menu under Tools -> Launch Express Console. Or, it can be launched through a command line interface in Qshell.

A Few Comments

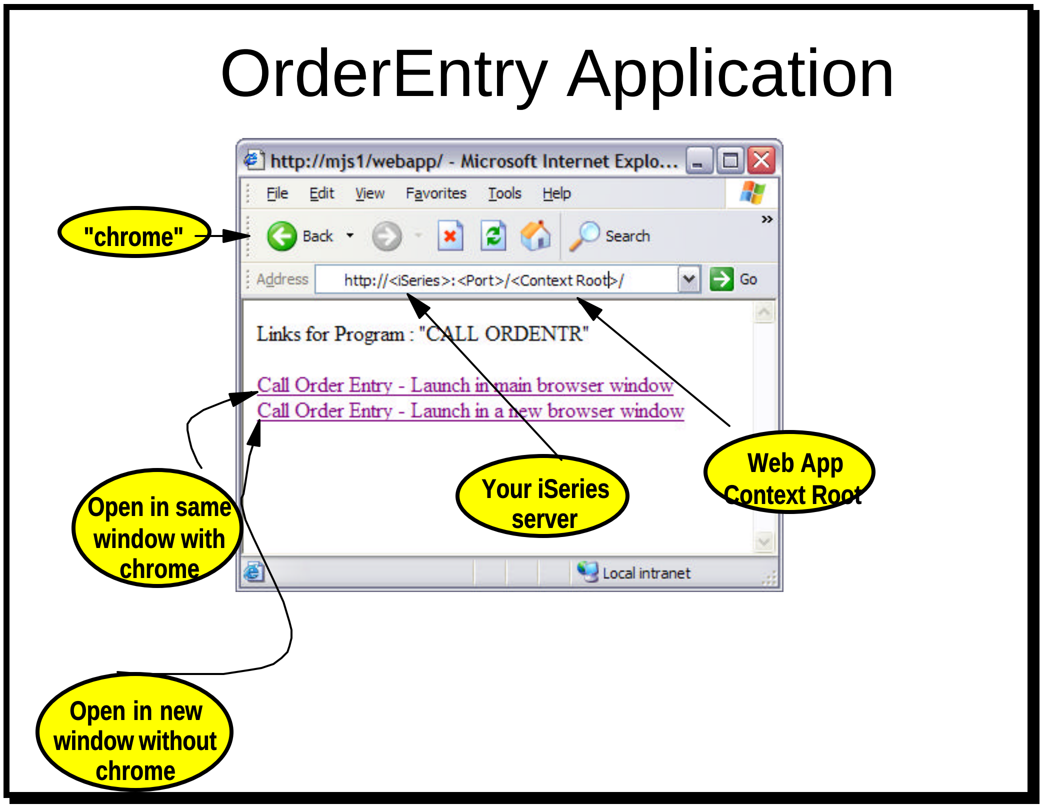
- **Before deploying, is WebSphere Application Server properly configured?**
- **Verify deployment of files by checking:**
  - Within IFS directory on iSeries server, or
  - Locally, on PC specified as the server
- **Invoke the first page**

A Few Comments

Just a few comments here… before attempting to export an IBM WebFacing Tool for iSeries project. Make sure WebSphere Application Server is up and running properly. It is necessary to configure WebSphere Application Server first. Installing WebSphere Application Server is getting easier with each release. For information pertaining to related redbooks, refer to the "Hotlinks" and “Additional Information” sections of this course.

In addition to the WebSphere Application Server graphical user interface, you can also verify that Web-faced files have been deployed by checking the IFS directories on the server.

Next, invoke the first page of the Web-faced application.



OrderEntry Application

http://mjs1/webapp/ - Microsoft Internet Explo...

Address http://<iSeries>:<Port>/<Context Root>/

Links for Program : "CALL ORDENTR"

Call Order Entry - Launch in main browser window

Call Order Entry - Launch in a new browser window

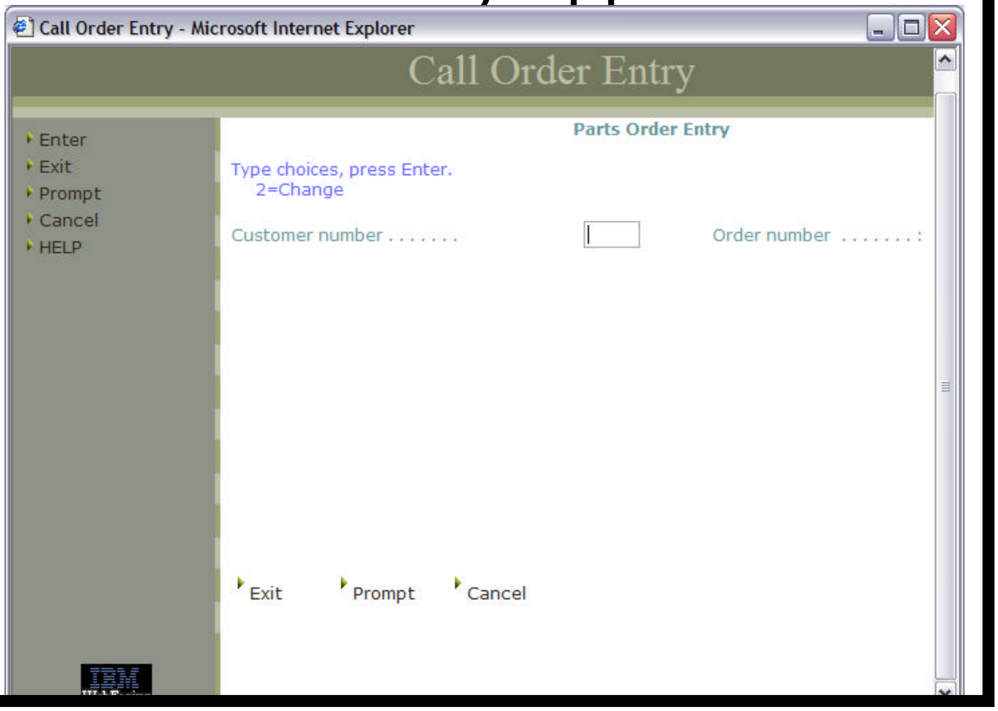
OrderEntry Application

Once WebSphere Application Server is running on your system, access your desktop browser to type in the URL of the invocation page for the Web-faced application. On this screen, you can see the location (http://<iSeries>:<Port>/<ContextRoot>) that has been entered—this is the invocation page that was generated in the example used for this course. The first part of the URL is the domain name of the Web (substitute the name of your server). The next part of the URL is the port configured to your HTTP server to listen on. If you configure the default (port 80), this can be left out. The final part of the URL is the name of the context root configured during deployment for WAR files or during development for EAR files.

Next, you will receive a Web page with two links to the Web-faced application. The first link starts the application in the current browser window. The second link opens a second browser window, without any “chrome” such as the toolbar. Choose which is preferred and copy that HTML anchor tag to your own invocation page. When either link is clicked, the program will start.

When the user clicks on the link, a call is automatically made to the appropriate Web-facing-generated servlet (and includes any needed parameters to tell the system which RPG or COBOL program should be executed). That application begins to run, and when it first does a read request on the display file, you will see the first Web-faced screen.

OrderEntry Application



Parts Order Entry

Type choices, press Enter.
2=Change

Customer number . . . .

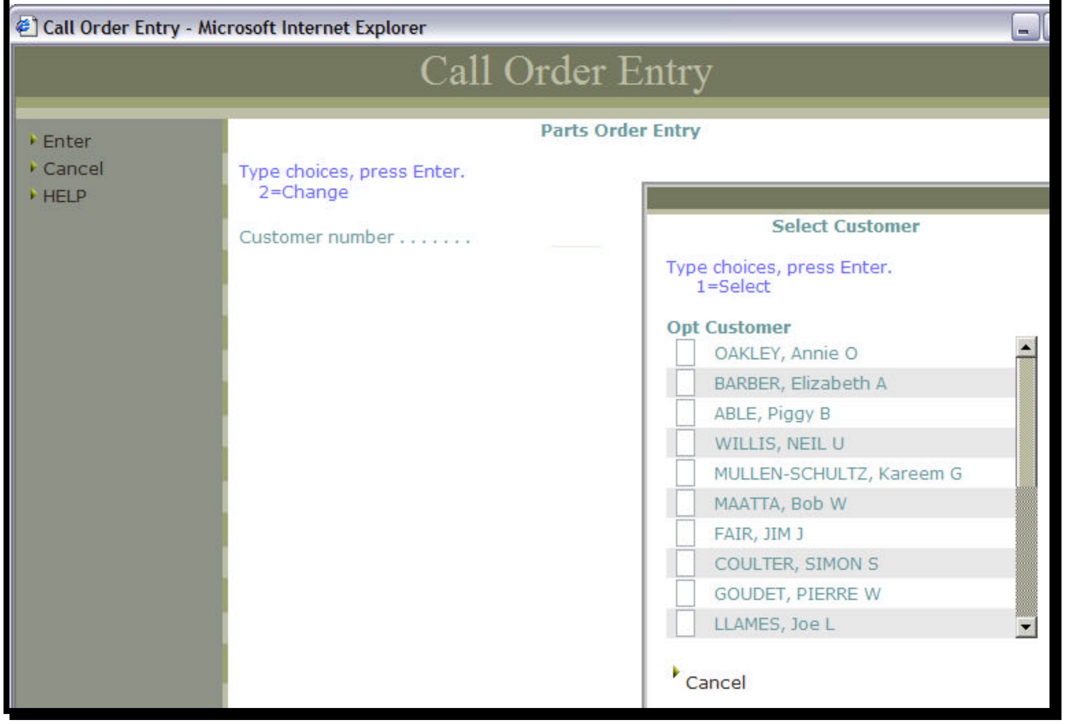
Order number . . . . :

OrderEntry Application (continued)

If a user ID and password were not imbedded into the project, the user will be prompted to enter a valid user ID and password for the iSeries server. The user ID must have the application on its library list. The next screen the user sees (after having clicked on the "Call Order Entry - Launch in a new browser window" link) is the first screen, as defined by the first record format "read" request in the Web-faced application. All record formats written to that point are displayed, just as with green-screen processing. Here, you see the sample screen generated by our practice example in this course.

The user will probably enter a specific customer number to view details related to that particular customer. Or, as is true with the 5250-based interface (and as shown on the next screen), the user can hit the F4 function key or click on the "Prompt" link in the browser to review the entire customer list.

Select a customer



Parts Order Entry

Type choices, press Enter.
  2=Change

Customer number . . . .

Select Customer

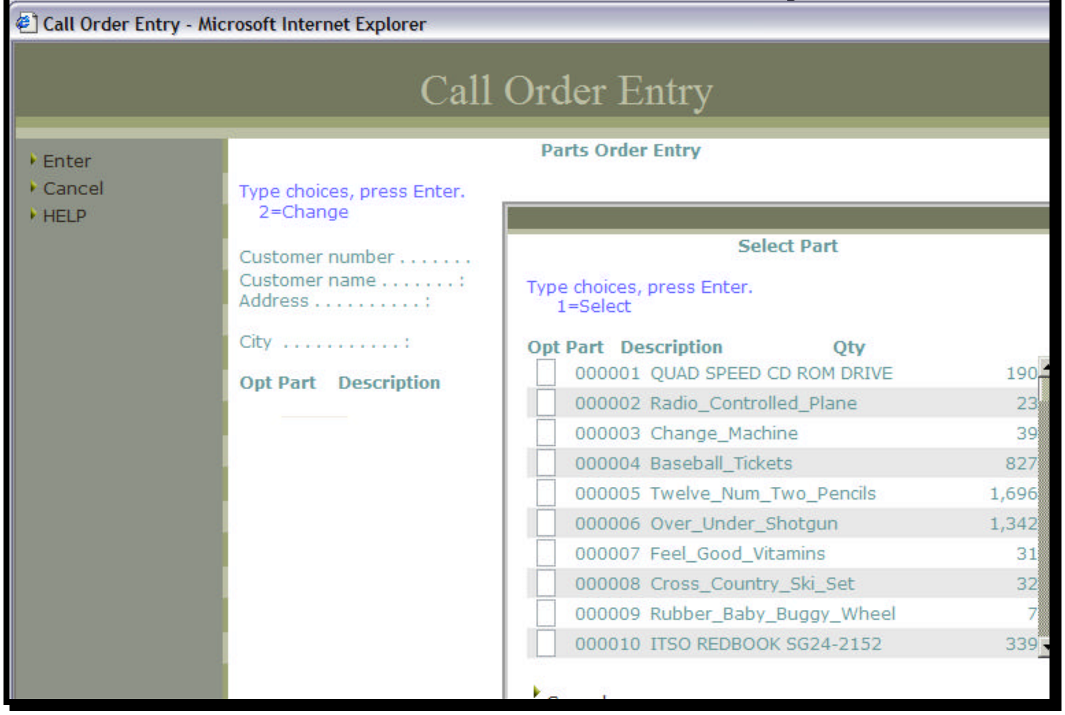
Type choices, press Enter.
  1=Select

Opt	Customer
☐	OAKLEY, Annie O
☐	BARBER, Elizabeth A
☐	ABLE, Piggy B
☐	WILLIS, NEIL U
☐	MULLEN-SCHULTZ, Kareem G
☐	MAATTA, Bob W
☐	FAIR, JIM J
☐	COULTER, SIMON S
☐	GOUDET, PIERRE W
☐	LLAMES, Joe L

Select a customer

This is a list of customers from which individual “viewing” selections can be made. As mentioned on the previous screen, this list is achieved by hitting the F4 button or clicking on the “Prompt” link. This is a conversion with no customization. If desired, the Select Customer pop-up window could have the Opt column removed and have the customer names as a clickable link that, when clicked, set and submit 1 in the opt field.

Select Inventory



Call Order Entry

Parts Order Entry

Type choices, press Enter.
2=Change

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

City . . . . :

Opt Part Description

Select Part

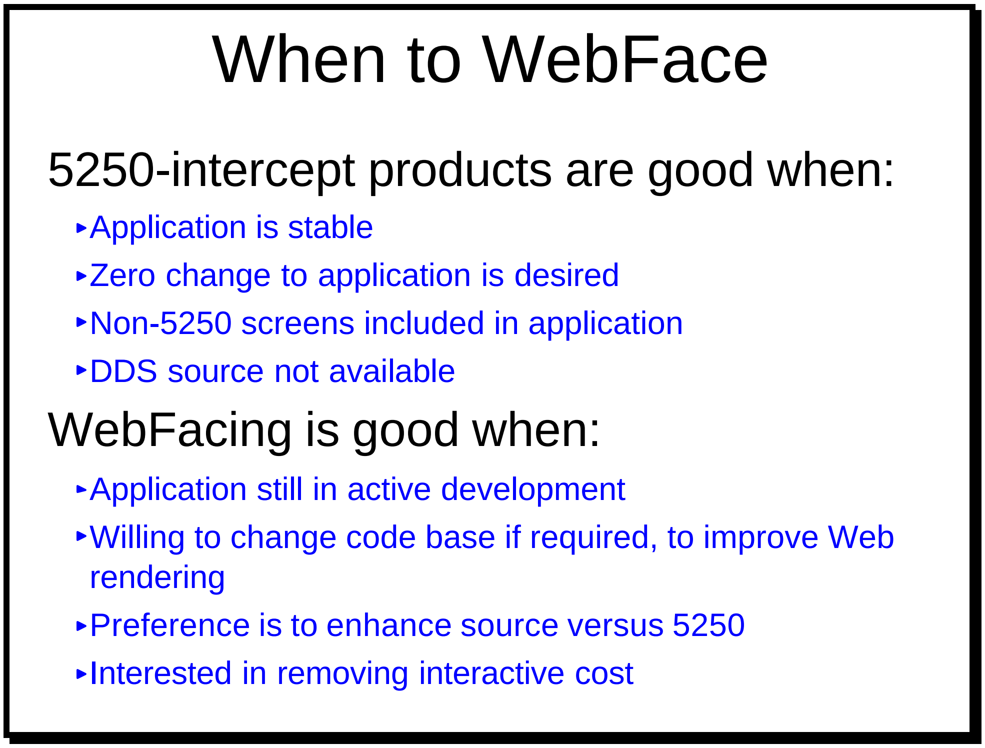
Type choices, press Enter.
1=Select

Opt	Part	Description	Qty
☐	000001	QUAD SPEED CD ROM DRIVE	190
☐	000002	Radio_Controlled_Plane	23
☐	000003	Change_Machine	39
☐	000004	Baseball_Tickets	827
☐	000005	Twelve_Num_Two_Pencils	1,696
☐	000006	Over_Under_Shotgun	1,342
☐	000007	Feel_Good_Vitamins	31
☐	000008	Cross_Country_Ski_Set	32
☐	000009	Rubber_Baby_Buggy_Wheel	7
☐	000010	ITSO REDBOOK SG24-2152	339

Select Inventory

Here is an example of what a Web-faced subfile looks like, by default. In this case, the user is viewing an inventory file from which individual line items can be ordered. If the primary input column uses a VALUES keyword, then a dropdown is generated which displays all the legitimate "value" choices from which an individual keyword can be selected.

It is also relatively easy, for simple selection lists, to hide the input column altogether, and to instead make that input column a link. In this case, when the user clicks on that link, the link itself will map the user to a location where he or she can enter a pre-defined number in the first column. The user can then press "Enter" to actually view the desired subfile. This type of enhancement is easily done using CODE Designer and its Web Settings property sheet for each field you wish to enhance for the Web.



When to WebFace

5250-intercept products are good when:

- Application is stable
- Zero change to application is desired
- Non-5250 screens included in application
- DDS source not available

WebFacing is good when:

- Application still in active development
- Willing to change code base if required, to improve Web rendering
- Preference is to enhance source versus 5250
- Interested in removing interactive cost

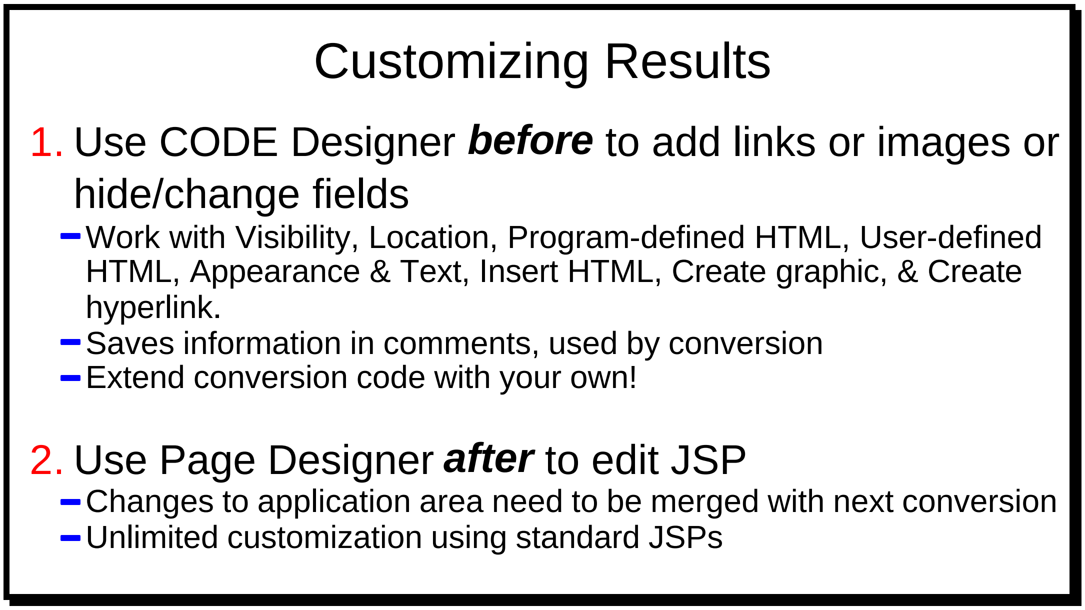
When to Web-face

In summary, when would you want to use the IBM WebFacing Tool for iSeries and when would the use of 5250-intercept products (such as screen-rendering products from Jacada® or Seagull®) be appropriate?

5250-intercept products are beneficial when the application being considered is stable and no changes are desired. Also, 5250-intercept products are good when your preference is to enhance runtime output versus development time source. Further, since the IBM WebFacing Tool for iSeries requires every displayed screen to be converted, an intercept product is preferable when every screen of every application and even of the operating system itself, is to be Web-enabled. There are also times when non-5250 screens, such as 3270 and UNIX screens, need to be Web-enabled as well. The IBM WebFacing Tool for iSeries does not address these.

On the other hand, IBM WebFacing Tool for iSeries is a good choice when the application is still in active development and you are willing to change the DDS source, and possibly the application code base as required, to improve Web rendering. The tool is also most desirable if your preference is to maintain and enhance the DDS source as the single definition for both the 5250 and the Web user interfaces, rather than customizing the Web-enabled interface independent of the 5250 interface.

Sometimes both options will be used… an intercept product to generically Web-enable every application and operating system screen, and the IBM WebFacing Tool for iSeries to optimize specific applications for the Web.



Customizing Results

1. Use CODE Designer ***before*** to add links or images or hide/change fields
    - Work with Visibility, Location, Program-defined HTML, User-defined HTML, Appearance & Text, Insert HTML, Create graphic, & Create hyperlink.
    - Saves information in comments, used by conversion
    - Extend conversion code with your own!
2. Use Page Designer ***after*** to edit JSP
    - Changes to application area need to be merged with next conversion
    - Unlimited customization using standard JSPs

Customizing Results

One of the last things to discuss is the process of customization. As stated earlier, you can customize code before the Web-facing conversion or afterward.

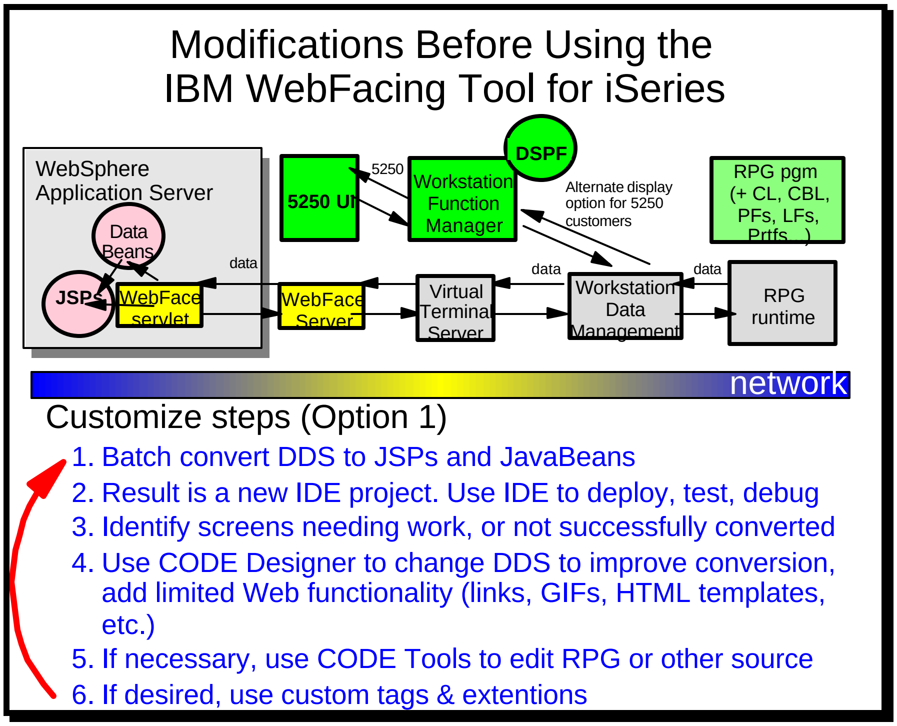
You may prefer to perform code customization before conversion because it allows you to maintain both the 5250 interface and a Web-enabled interface easily—in one step. Use CODE Designer to make modifications that affect the Web-faced interface.

CODE Designer is the SDA of the future. It is one of the components of WebSphere Development Studio Client for iSeries. CODE Designer has a rich graphical interface for designing DDS displays, as well as printer and physical file output. It is such a fantastic product; you might actually want to create new green screens with it!

You can use CODE Designer before the conversion to add links or images, or to hide or change fields. By using CODE Designer to customize code information, the changes that impact the IBM WebFacing Tool for iSeries interface are automatically saved in the comments. This information is used during the conversion, but is also available for modifications or if the Web-facing interface must be regenerated.

It is also possible for you to extend this IBM conversion code with your own conversion code. The conversion code is written in Java and has well-defined extension points for intercepting the logic and inserting custom generation code. This option is mostly of interest to Business Partners looking to offer alternative generation algorithms.

The alternative to using CODE Designer to customize the user interface prior to conversion is to use the built-in source editor (Page Designer) in WebSphere Development Studio Client tool to edit the generated output source directly. The drawback to editing after the conversion is that the changes must be merged in after a subsequent conversion. It is ultimately a manual process to reincorporate post-conversion changes. The following pages look at these two alternatives in more detail.



Modifications Before Using the IBM WebFacing Tool for iSeries

WebSphere Application Server
Data Beans
JSPs
WebFace servlet
data
5250 UI
5250
Workstation Function Manager
DSPF
Alternate display option for 5250 customers
RPG pgm (+ CL, CBL, PFs, LFs, Prtfs...)
WebFace Server
Virtual Terminal Server
data
Workstation Data Management
data
RPG runtime
network

Customize steps (Option 1)

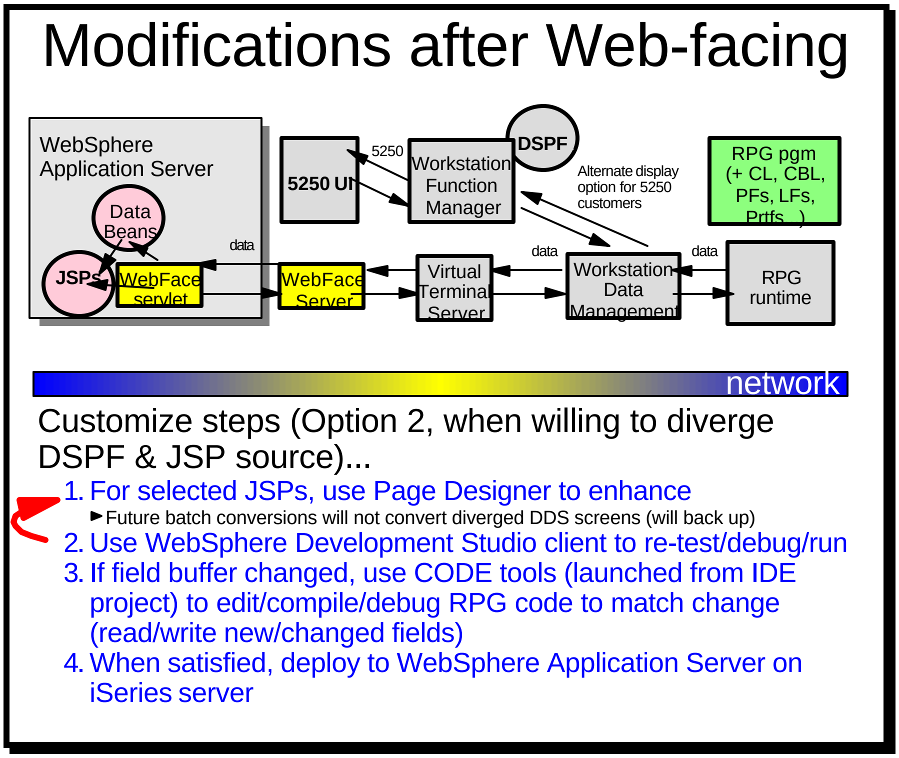
1. Batch convert DDS to JSPs and JavaBeans
2. Result is a new IDE project. Use IDE to deploy, test, debug
3. Identify screens needing work, or not successfully converted
4. Use CODE Designer to change DDS to improve conversion, add limited Web functionality (links, GIFs, HTML templates, etc.)
5. If necessary, use CODE Tools to edit RPG or other source
6. If desired, use custom tags & extentions

Modifications before Using the Web-facing Tool

Conversion can occur during development of the application (that is, before the use of IBM WebFacing Tool for iSeries). With the use of CODE Designer, these customization steps can be taken before conversion takes place:

1. You need to batch convert the DDS to JSPs and JavaBeans.
2. The result is a new IDE project. The IDE is used to deploy, test, and debug.
3. Identify the screens needing work or those screens not successfully converted.
4. Use CODE Designer to change the DDS to improve the conversion. Some specific, relatively-limited Web functionality; such as links, GIFs, HTML templates, etc. may be added.
5. If necessary, use CODE Edit Tools, another component of CODE Designer, to edit RPG or other source.
6. If you desire, a unique extension to the IBM WebFacing Tool for iSeries conversion tool could be written.

[NOTE: Now you are ready to convert the DDS to JSPs and JavaBeans for real. So, execute step 1 again.]



Modifications after Web-facing

WebSphere Application Server
Data Beans
JSPs
WebFace servlet
data
WebFace Server
5250 UI
5250
Workstation Function Manager
DSPF
Alternate display option for 5250 customers
Virtual Terminal Server
data
Workstation Data Management
data
RPG pgm (+ CL, CBL, PFs, LFs, Prtfs...)
RPG runtime
network

Customize steps (Option 2, when willing to diverge DSPF & JSP source)...

1. For selected JSPs, use Page Designer to enhance
   - Future batch conversions will not convert diverged DDS screens (will back up)
2. Use WebSphere Development Studio client to re-test/debug/run
3. If field buffer changed, use CODE tools (launched from IDE project) to edit/compile/debug RPG code to match change (read/write new/changed fields)
4. When satisfied, deploy to WebSphere Application Server on iSeries server

Modifications after "WebFacing"

With the use of the WebSphere Development Studio Client Page Designer or any editor, these customization steps can be taken after conversion takes place—as long as you are willing to diverge the DSPF and JSP source.

1. For the selected JSPs, use an editor to enhance the JSPs.
2. WebSphere Development Studio Client for iSeries is then used to redeploy, re-test, debug, and run. If additional changes need to be made to enhance a JSP further, perform step 1 again, then step 2 again, as many times as necessary to be pleased with the enhancements. If satisfied, proceed to step 3.
3. If the field buffer is changed, such as by the addition of named fields, use CODE Edit Tools (launched from the IDE project) to modify the RPG or COBOL code to match the new buffer layouts.
4. After testing the Web-faced application in the integrated test environment in WebSphere Development Studio Client, the application is deployed to WebSphere Application Server on the iSeries server.

[NOTE: The display file object must also contain the named fields for the RPG or COBOL code in order to read and write them. This can be accomplished by adding hidden named fields into the display file that are used when targeting a Web user interface, but ignored when targeting a 5250 user interface.]

Now you can WebFace your iSeries Applications
in your sleep!

Conclusion

To meet the goals of 21st century businesses, IBM developed the IBM WebFacing Tool for iSeries to be included with WebSphere Development Studio Client for iSeries. It helps you to quickly build the Web interfaces you need for existing, strategic 5250 applications. The IBM WebFacing Tool for iSeries will continue to evolve in the depth of its support to iSeries users who need to convert applications to a Web interface.

Some of the main requirements to utilize this tool are:

- A supported version of WebSphere Application Server must be configured, up, and running. Since it pre-requires the HTTP powered by Apache, this too must be configured and running. Each Web-facing project requires a Web application to be configured for WebSphere Application Server, via the WebSphere Application Server Administrative Console.
- OS/400 must be at V5R1 or later. You must also have the latest Service Packs for the tool (available at the Web site referenced each time the IBM WebFacing Tool for iSeries is started.) There are also required PTFs for the runtime. (Be sure to reference the "Hotlinks" section of the course for a link to these PTFs.)

When you Web-face 5250 applications, existing display files are converted into JavaServer Pages and JavaBeans—both are industry standards. Chances are good that no changes will have to be made to stable 5250 applications.

Another IBM WebFacing Tool for iSeries advantage is that Web-faced applications can still run in a 5250 environment or the application can run via its newly created Web interface. When a Web-faced application program starts on an iSeries server, a switch is set on the system. This switch indicates which interface the program will be using—a 5250 interface or a Web interface. Since a Web-faced application is one program supporting a Web interface and a 5250 interface, there is no dual program maintenance required.

Not only is the IBM WebFacing Tool for iSeries a “home run” for those who are fretting about how to get their applications up and running on the Web successfully, but the product is easy to use with lots of built-in wizards. IBM has had a very positive reaction already from users who are excited to move forward with many application conversions they have put off until the availability of this excellent Web-facing application tool.

Good luck as you Web-face your iSeries applications!

Final Test

1. In yesterday's commonly-used RPG design model, the RPG program would include code to:
   ☐ a. Write the screen
   ☐ b. Read the screen
   ☐ c. Process the screen
   ☑ d. All of the above—Correct. They also had to make decisions about what to show the user next, as well as write code for the business logic, the database access logic, and the report logic.

2. The problem with the old RPG approach is the difficulty a programmer faced in substituting or augmenting any one piece of a program—such as the user interface.
   ☑ True—Correct. The user interface is difficult to change because the code is married to the 5250 user interface.
   ☐ False

3. IBM WebFacing Tool for iSeries does NOT allow the RPG "blended" program to continue running as is, thinking it is talking to a 5250 user interface.
   ☐ True
   ☑ False—Correct. The IBM WebFacing Tool for iSeries DOES allow the RPG program to continue to run as is, thinking it is talking to a 5250 UI. However, the tool replaces that green-screen!

4. The IBM WebFacing Tool for iSeries does not address all the requirements for developing new, Web-enabled applications?
   ☑ True—Correct. It is a tool with a very specific role. If an application is strategic and there are long term needs for another interface, you should go into the code, rip out the 5250 interface from the business logic, and build a new application that separates the user interface from the business logic. However, it is hard to rip out the 5250 interface and that is not what this product is about.
   ☐ False

5. The IBM WebFacing Tool for iSeries buys time while rewriting an entire application or program from scratch.
   ☑ True—Correct. It is to be used as a short-term bridge to help programmers get an application up and running in a Web browser when that application already exists as an application for a 5250 device. This tool could be a good, long-term option for an application where you want to maintain both the 5250 interface and a Web interface.
   ☐ False

6. The buffer going back and forth between the RPG model and the controller would be language dependent.
   ☑ True—Correct. For example, using current technology to create a Web application, the controller would be a Java servlet, the buffer would be a JavaBean, and the view would be a Web page—perhaps a JavaServer Page (JSP).
   ☐ False

7. The three application components are:
   ☑ a. Controller, Buffer, and program model—Correct. These three application components — controller, buffer, and program model (which includes business logic as well as database access)—are separate pieces of code. These "pieces of code" can more easily move into the future and embrace ever-changing technologies without the need to rewrite the entire application.
   ☐ b. Business Logic, UI Processing, and DB Access
   ☐ c. View, Logical Display, DB Processing
   ☐ d. None of the above

8. The IBM WebFacing Tool for iSeries allows you to emulate the controller-buffer-OPM environment, without rewriting existing "blended" applications.
   ☑ True—Correct. This is because, while the final answer today for user interfaces is the Web, the final answer for tomorrow will be different. In light of these facts, you need to be able to easily modify each code component independently.
   ☐ False

9. What are the "Ways to Web"?
   ☐ a. Through the use of Domino or WebSphere Commerce
   ☐ b. Through the use of Java Applets or CGI-Bin programs
   ☐ c. Through the use of WebSphere Development Studio Client for iSeries or IBM WebFacing Tool for iSeries
   ☑ d. All of the above—Correct. Domino is a fantastic way to get an application up on the Web. Domino can serve document databases to the Web. WebSphere Commerce is useful if the programmer does not want the Web application to get into all the details related to money-based Web transactions. The features of WebSphere Commerce are ideal if selling over the Internet. Java applets are great if there is an in-house application where everyone is connected via a T1 line. However, for code deployed over the Internet, applets can be a performance concern because you cannot predict how long it will take for the applets to come down to the browser. In general, most Java developers avoid applets today, because of the download time.

10. CGI stands for:
    ☐ a. Correctable Gateway Interface
    ☐ b. Corrupt Gateway Instance
    ☑ c. Common Gateway Interface—Correct. CGI is a standard method used by a Web server to pass browser requests to an application program. CGI is also the standard vehicle for the Web server to receive data back from the application program to forward to the user.
    ☐ d. All of the above

11. The IBM WebFacing Tool for iSeries is exciting because it is relatively painless to implement and it also incorporates IT industry standards in its approach.
    ☑ True—Correct. Because servlets and JSPs are in demand by programmers who develop Web interfaces for iSeries applications, IBM now offers the "IBM WebFacing Tool for iSeries" (a part of the WebSphere Development Studio Client for iSeries) as a new alternative to get applications up and running on the Web.
    ☐ False

12. Although a "blended" RPG program contains both business logic and interface logic, the act of Web-facing is achieved by taking the entire RPG "blended" program and treating it as "business logic only."
    ☑ True—Correct. The IBM WebFacing Tool for iSeries allows you to easily generate JSPs and servlets from existing 5250 programs. The RPG program, by virtue of its structured nature, determines which screen to display next and what to include on that screen.
    ☐ False

13. Based on the 5250 display file, DDS that is used to create a Web interface for existing 5250 applications, the IBM WebFacing Tool for iSeries generates:
    ☐ a. JavaServer Pages
    ☐ b. JavaBeans
    ☐ c. Servlets
    ☑ d. All of the above—Correct. This newly generated Web interface code works in concert with the RPG program, thereby quickly allowing strategic business applications to be presented on the Web. Generally, using the Web user interface does not require any changes to the RPG program, and it can still be accessed through a 5250 interface if required.

14. The IBM WebFacing Tool for iSeries is part of:
    ☑ a. WebSphere Development Studio Client for iSeries—Correct.
    ☐ b. WebSphere Commerce
    ☐ c. WebSphere Design Tools Suite
    ☐ d. All of the above

15. The IBM WebFacing Tool for iSeries consists of two pieces.
    ☑ True—Correct. It contains a conversion tool for the DSPF-to-JSP conversion, and a runtime intercept that enables an iSeries application to be run as a Web-faced application for the Web. The tool also includes many wizards to facilitate the process of Web-facing an application.
    ☐ False

16. During the IBM WebFacing Tool for iSeries development time, there is NOT a source-to-source conversion.
    ☐ True
    ☑ False —Correct. The development-time conversion window provided by this tool is its key differentiator when compared to other Web-facing options. During development time, there is a source-to-source conversion (DSPF DDS is converted to JSPs and JavaBeans).

17. The IBM WebFacing Tool for iSeries creates all the components—JSPs and JavaBeans—at development time.
    ☑ True—Correct. Nothing is generated at runtime, so the user interface is already waiting for the application. At runtime, the buffers are intercepted to drive the JSPs versus the Display File. The result is, in many cases, good performance. Since IBM WebFacing Tool for iSeries utilizes a source-to-source development time conversion, you can iterate the output at development time to ensure that is presented exactly as expected—before deployment to production.
    ☐ False

18. There are two ways you can improve the looks of a "Web-faced" screen.
    ☑ True—Correct. One method is utilized prior to conversion, while the other is an after-conversion process.
    ☐ False

19. UI stands for:
    ☐ a. User Instance
    ☐ b. Unstable Interface
    ☑ c. User Interface—Correct. The benefit of using the IBM WebFacing Tool for iSeries is that users can run interactive programs in either a 5250 or “Web-faced” UI mode.
    ☐ d. None of the above

20. There are three runtime tiers in an IBM WebFacing Tool for iSeries application.
    ☑ True—Correct. First is the view, which is represented by the browser tier. Next is the middle tier (or logical tier) where the runtime servlet is managing the JSPs and JavaBeans. The last tier (the host tier) is where the 5250 application thinks it is still talking to a green screen.
    ☐ False

21. The IBM WebFacing Tool for iSeries is strictly used for public Web sites:
    ☐ True
    ☑ False —Correct. The tool does not have to be used for a public Web site; you might want to Web-face a back-office payroll application for a private intranet.

22. How many JavaBeans are generated when the IBM WebFacing Tool for iSeries is used?
    ☐ a. 2
    ☑ b. 3—Correct. For every record format, there will be three beans—a Data Bean, a View Bean, and a Feedback Bean. The Data Bean (Data.java), is used to store data coming in and out of the Web-faced application. The Data Bean contains the customer data and the View Bean contains instructions on how to display that data. The FeedBack Bean assists in the application flow by updating the values in the Data Beans, as well as passing data and status information from the browser to the Data Bean.
    ☐ c. 4
    ☐ d. None

23. How many JSPs are created by the IBM WebFacing Tool for iSeries?
    ☑ a. 2—Correct. Two—one JSP is used to display the user interface and the second JSP is used to contain JavaScript which deals with issues such as error checking.
    ☐ b. 3
    ☐ c. 4
    ☐ d. None

24. An invocation page is a Web page with links.
    ☑ True—Correct. The IBM WebFacing Tool for iSeries wizard generates an invocation page for each conversion. Selecting a link on the invocation page runs the associated CL command to start the application that was converted using this wizard.
    ☐ False

25. The successful installation and running of WebSphere Application Server is not a necessary step.
    ☐ True
    ☑ False —Correct. WebSphere Application Server is an important part of this whole process. Before the files can be deployed, the programmer needs to ensure that the WebSphere Application Server instance is up and running. In some installations, this task may take some time. Further, a Web Application must be configured once for each Web-facing project.

26. Programmers are not allowed to create their own styles.
    ☐ True
    ☑ False—Correct. Programmers can also create their own styles.

27. When converting DDS to JSPs, you have to be signed on to an iSeries server to get the DDS source.
☑ True—Correct. To do the actual conversion, you have to be signed on to iSeries server to get the DDS source. You will be prompted for the log-on if not already signed on to the iSeries server. Then, you simply wait for the conversion to complete. Typically, a conversion takes a few minutes per display file.
☐ False

28. What is needed to generate publishing files?
☐ a. Runtime iSeries name
☐ b. Administrative e-mail address
☐ c. Password
☑ d. All of the above—Correct. To generate publishing files, the Runtime iSeries name, user ID, password, and administrative e-mail address is needed. The information entered is used every time the Web-faced application is called. The administrative e-mail address is important because, if there is a problem, a message describing the problem and requesting help can be generated by the user, and transmitted to this e-mail address for action.

29. How many times would you go through the conversion process?
☐ a. Once every three months
☐ b. Once a day
☑ c. Multiple times—Correct. You go through the conversion process each time the DDS source changes.
☐ d. None of the above

Hotlinks Related to this On-line Course

These Hotlinks are referenced from the speaker notes. They provide useful Web sites and reference materials to supplement the information in this course.

- WebSphere Development Studio Client for iSeries
  ibm.com/software/awdtools/wdt400/
- IBM WebFacing Tool for iSeries Resource Site
  ibm.com/software/awdtools/wdt400/about/webfacing.html
- IBM WebFacing Tool for iSeries PTF information
  ibm.com/support/docview.wss?rs=0&uid=swg27002213
- IBM WebSphere Web site
  ibm.com/eserver/iseries/websphere

Other References and Additional Information

Useful Web sites:

- IBM WebFacing Tool site from the IBM eServer enablement organization
  ibm.com/servers/enable/site/ebiz/webfacing/
- IBM eServer iSeries
  ibm.com/eserver/iseries/
- iSeries for e-business technologies
  ibm.com/servers/solutions/e-business/iseries/
- alphaWorks (lots of free Java stuff, such as JavaBeans, XML tools, etc.)
  ibm.com/alphaworks
- WebSphere for iSeries site
  ibm.com/eserver/iseries/websphere
- IBM WebFacing Tool for iSeries Enablement Web site
  ibm.com/servers/enable/site/ebiz/webfacing/hints.html

Redbooks & Redpapers

At the IBM Redbooks Web site (**ibm.com**/redbooks) type the "SG" number into the search engine. Or, search for WebSphere, servlets, JavaServer Pages, etc.

- WebSphere Development Tools for iSeries—Generate Web front ends to existing applications (redpaper REDP0516)
- WebSphere Development Studio Client for iSeries: Bringing New Life into 5250 Applications SG24-6600-00
- Design and Implement Servlets, JSPs, and EJBs for IBM WebSphere Application Server (SG24-5754-00)
- Domino and WebSphere Together (SG24-5955-00)

Books

These books are referenced as additional information resources. The referenced Web sites are given as a suggested starting place only.

- “Java for RPG Programmers,” “Java for COBOL Programmers,” “Student Workbook for Java for RPG and COBOL Programmers,” and related books
  www.amazon.com
- Various books on Java and WebSphere
  www.mc-store.com/java3.html
- Tons of books on Java, VisualAge, WebSphere, etc.
  www.amazon/borders/fatbrain.com

Frequently Asked Questions

Question: Why do I get a blank page when I invoke my Web-faced application?

Answer: If the WebSphere Application Server instance is not running, it is possible to get a blank page when invoking the application. This can also occur if the instance is starting but has not come up completely. To verify that the instance is running, run the WRKACTJOB command and locate the subsystem under which your WebSphere Application Server instance is running, such as QASE5 for WebSphere Application Server — Express V5. Locate the administrative job if it is WebSphere Application Server V4 or WebSphere Application Server V5 Base or Network Deploy for the instance in question. If it is a WebSphere Application Server — Express V5 instance, then it is the job using the instance name. Specify option 5 and press 'enter.' From the menu that appears, choose option 10 to view the job log for the administrative job. Look for the message 'WebSphere administration server instance_name ready'. If you see this message, the instance is up and running.

Question: How do I change the user ID and password for a Web-faced application?

Answer: When you create a new Web-face project, you are prompted for a user ID and password when selecting which DDS members to convert. It is this user ID and password that are used during runtime when executing the Web-faced application. If you wish to use a different user ID and password during runtime, you must change them in WebSphere Development Studio Client for iSeries. To do so, rightmouse-click on the Web-face project in WebSphere Development Studio Client. From the pop-up menu, choose "Properties." In the Application Properties dialog, change the user ID and password to the desired values. You must re-export your project after making these changes.

[NOTE: A good additional source for information that is similar to FAQs is maintained at the IBM WebFacing Tool for iSeries Enablement Web site, which is found at: ***ibm.com****/servers/enable/site/ebiz/webfacing/hints.html .]*

Trademarks

IBM, the IBM Logo, e-business logo, eServer, iSeries, OS/400, WebSphere, Lotus, Notes, and Domino are trademarks or registered trademarks of the International Business Machines Corporation in the United States or other countries or both.

Java, Java 2 Enterprise Edition, J2EE, JavaServer Pages, JSPs, Enterprise JavaBeans, EJBs, JavaScript, and all Java-related trademarks or logos are trademarks or registered trademarks of Sun Microsystems, Inc in the United States and other countries.

Jacada is a trademark or registered trademark of Jacada.

Seagull is a trademark or registered trademark of Seagull software.

Other company, product, and service names listed throughout this presentation are the trademarks of their respected providers.