

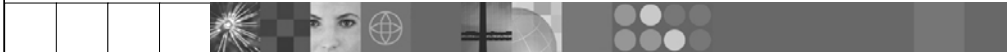


IBM Software Group

iSeries WebFacing IBM eServer iSeries Application Development

iSeries AD Team

WebSphere. software



iSeries
WebFacing

2003 | WDS V5R2 and WDSC 5.0, 5.1

© 2003 IBM Corporation

This presentation reviews the components of a Web application, the WebFacing Tool and how it is packaged, how the WebFacing Tool development time and runtime works and the steps to Web-enable a 5250 application using the WebFacing Tool.

Table of contents

□ □ e-business Primer

What is WebFacing?

WebSphere Development Studio Client

WebFacing Tools

Customizing WebFacing

What's new in WebFacing 5.0 and 5.1.0

Summary

In the first section of this presentation we review what e-business is all about. Then we look at WebFacing and what it is. Next we introduce Development Studio Client. The WebFacing Tools are described in more detail. WebFacing customization is described followed by a review of the new WebFacing Tool Version 5.0 features.



e-business Primer

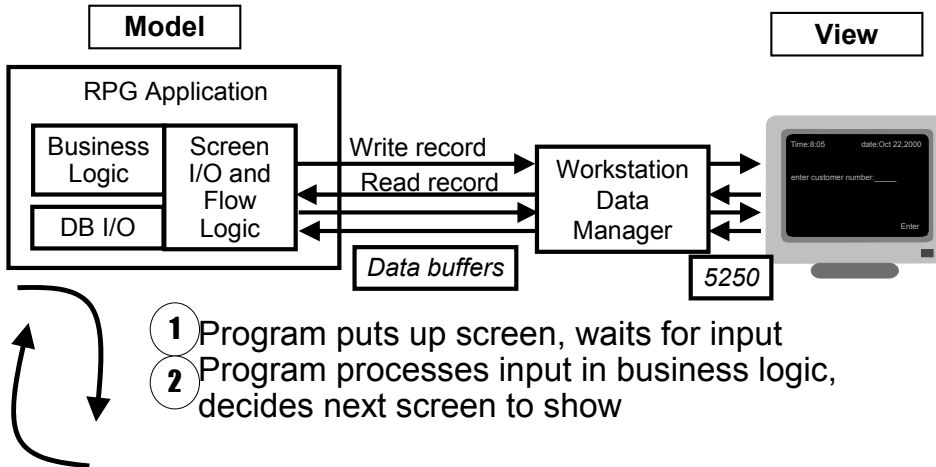
- Program Models

- Servlets

- JSPs

In this section we review the programming models and their associative technology, servlets and JSPs.

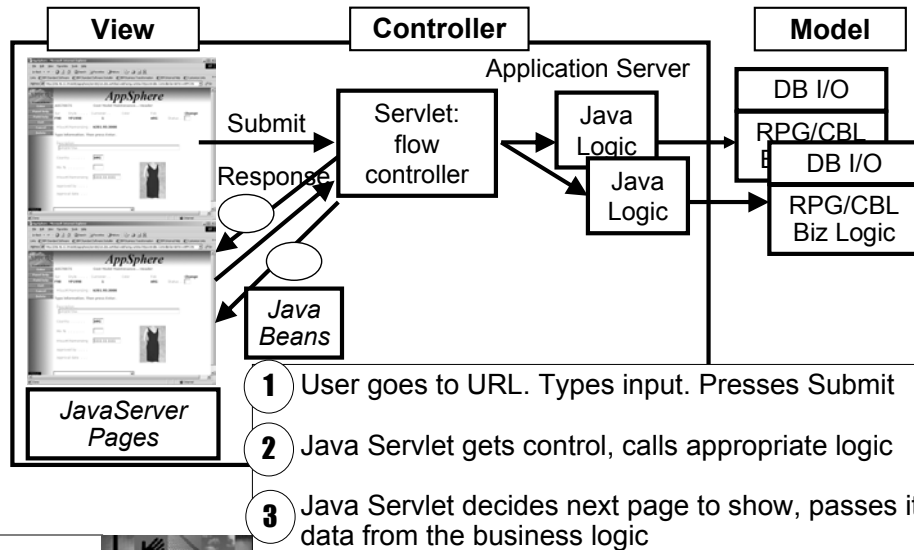
iSeries Classic Program Model



In the green-screen environment, the application performs READs and WRITEs to the workstation. The application data is sent to Workstation Data Manager. The Workstation Data Manager merges application data with the display file. The Workstation Data Manager generates a 5250 datastream that is sent to the display.

In this model, you typically have one model object and multiple views on that object. A view is a window onto the model. Presentation logic typically goes in the view. Business logic goes in the model.

E-business Program Model



The e-business program model follows the Model-View-Controller paradigm, organizing the application into three separate components:

- Model: the application model with corresponding data representation and business logic
- View: data presentation, providing views for user input
- Controller: to dispatch requests and control data flow

Here you can see the controller is added. The controller handles the interactions between the view and the model. When the model changes it updates the view, when the user does something with the view the controller informs the model.

First the application is converted. This creates JSPs for each record format as well as Java beans. When the program performs a READ on a record format, control as well as the application data is sent to the controller.

Because the job was started by the WebFacing server, the controller knows this is a WebFacing request and passes the data and control to the WebFacing server which runs on the iSeries. Control returns to the Webfacing runtime servlet that runs in WebSphere Application Server. The WebFacing servlet locates the appropriate JSPs and Java beans. The WebFacing servlet tells WebSphere Application Server to return the JSP back to the browser. The JSP is compiled and the resultant HTML is returned to the browser. Note that no 5250 datastream is generated in this flow.

What Are Servlets?

J2EE
Servlet
Spec 2.3

- **Servlets are . . .**
Java classes (programs written in Java)
- **Servlets run . . .**
On an application server such as WebSphere
- **Servlets are called . . .**
By a user pressing Submit, or clicking a link, that is mapped to the servlet
- **The input to Servlets are . . .**
User-entered data from a Web page
- **The output of a Servlet is . . .**
Control passed to a JavaServer Page

Servlets are server-side Java programs that use the *Sun Microsystems Java Servlet API* and its associated classes and methods, as defined in the *Sun Microsystems Java Servlet 2.3 Specification*. These Java programs extend the functionality of a Web server by generating dynamic content and responding to Web client requests. When a browser sends a request to the server, the server can send the request information to a servlet, so that the servlet construct the response that is sent back to the browser.

Just as applets run on a Web browser and extend the browser's capabilities, servlets run on a Java-enabled Web server, such as the WebSphere Application Server, to extend the server's capabilities. Servlets are commonly used to allow businesses to connect databases to the Web, due to their flexibility, scalability, and the their processing economy when developed in the WebSphere Studio Web development environment.

What Are JSPs?

J2EE
JSP Spec
1.2

Just
like a
*DSPF

- **JavaServer Pages (JSPs) are . . .**
.jsp files containing html tags plus JSP tags
- **JSP tags . . .**
Allow dynamic data to be inserted into the static HTML
Data is extracted from Java Beans passed to the JSP
- **JSPs are called by . . .**
Your servlet
- **The input to JSPs are . . .**
Java Beans passed from your Servlet
- **The output of a JSP is . . .**
A fully resolved Web page, displayed to user
Can be just HTML, or DHTML = HTML 4.0+CSS+JavaScript

JavaServer Pages enable you to generate dynamic web content, such as HTML, DHTML, XHTML, and XML files, to include in a Web application. JSP files are one way that the WebSphere Studio implements server-side dynamic page content. JSP files allow a Web server, such as WebSphere Application Server or Apache Tomcat, to dynamically add content to your HTML pages before they are sent to a requesting browser.

When you deploy a JSP file to a Web server that provides a servlet engine, it is preprocessed into a servlet that executes on the Web server. This is in contrast with client-side JavaScript (within <SCRIPT> tags), which is executed in a browser. A JSP page is ideal for tasks that are better suited to execution on the server, such as accessing databases or calling Enterprise Java beans.

Table of contents

e-business Primer

□ □  **What is WebFacing?**

WebSphere Development Studio Client

WebFacing Tools

Customizing WebFacing

What's new in WebFacing 5.0 and 5.1.0

Summary

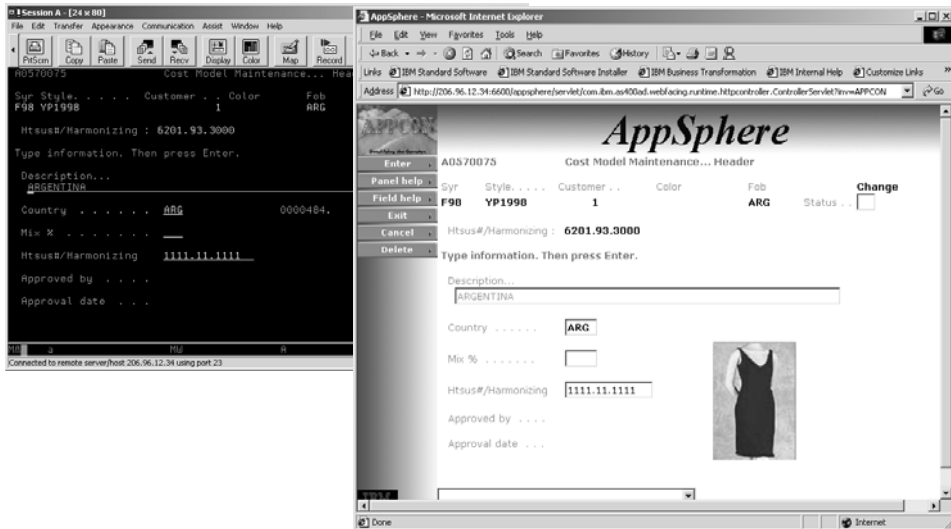
Now you know what an e-business application is and what technology it uses. Now we introduce what WebFacing is.

What is WebFacing?

- Overview
- Goals
- History

In this section we first give you an overview of the WebFacing Tool followed by goals and history.

WebFacing



This slide shows a sample "before and after" of a green screen that has been WebFaced. These screens are courtesy of APPCON, an IBM iSeries business partner. See more at www.appcon4.com.

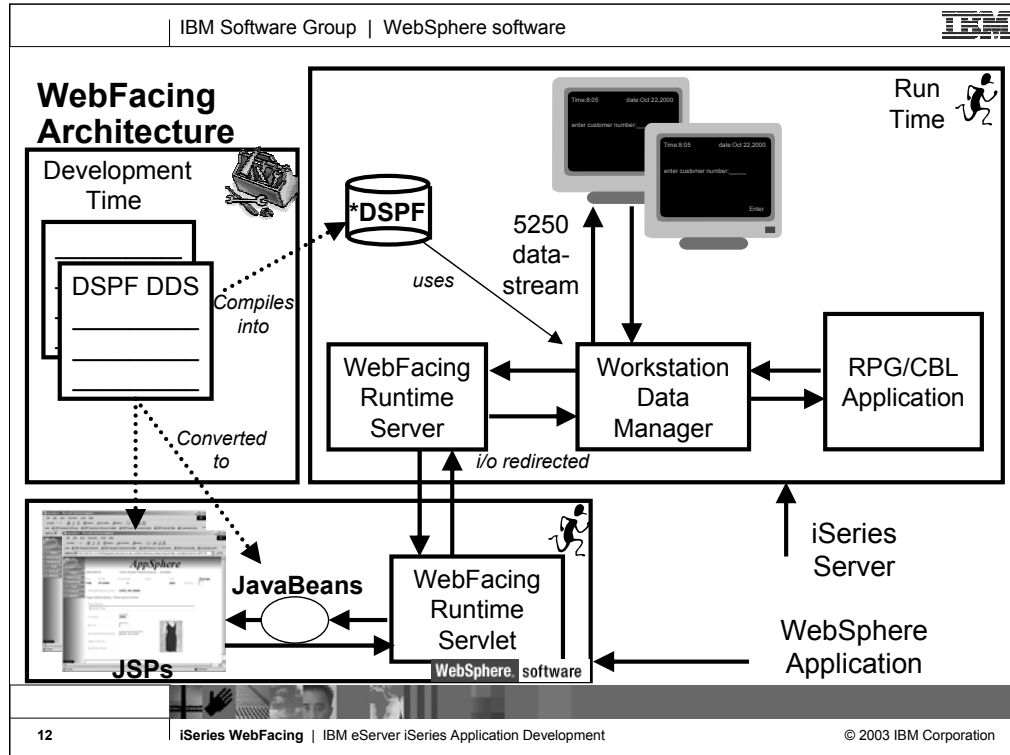
As you can see IBM's new WebFacing Tool converts existing 5250 interfaces to browser-based graphical user interfaces. With little or no modification to your original iSeries applications, you can extend the use of your programs to the Internet or an Intranet. Whether your applications are new or were written before the Internet became a viable platform for conducting business, with the WebFacing Tool, your applications can be available anywhere that users have access to a browser. You can use the WebFacing Tool with applications where DDS source code was used to create 5250 display screens. The tool has user-friendly wizards that facilitate selecting your original application's DDS source, converting the source, and deploying the new browser-based interface to your program as a WebSphere application. The conversion creates JavaServer Pages and JavaBeans that substitute for your DDS code and make Web access possible. After your DDS code has been converted, you can access the application through a browser or continue to use 5250 displays. Having the interface to your applications based on JavaServer Pages allows for more flexibility in customizing their appearance. Before your DDS code is converted, you can use the Style properties pages to change the look and feel of the pages that will be generated for you. Styles allow you to define attributes in your Web pages such as graphics, fonts, colors, and layouts. You can use one of the supplied styles or create your own. If you would like to update the appearance of a previously converted project, simply run the WebFacing Tool again and select a new style.

What is WebFacing?

- A Re-Facing **tool** and **runtime** for iSeries 5250 applications
- **Development-time conversion**
 - Of 5250 display file DDS source into Web page source (JavaServer Pages or JSPs)
- **Run-time intercept**
 - Of workstation data manager I/O to pass application data to generated Web application, prior to creation of 5250 data stream
 - Enables existing applications to run without change, indeed without even knowing they have been WebFaced

With the WebFacing Tool, you can quickly convert your DDS display file source members so that the user interface of your iSeries programs can run in a browser. When you convert your DDS display files, JSPs and Java beans are generated for you that substitute for the DDS code and make Web access possible. In the WebFacing Project wizard, you can select one or more DDS source members to convert, and select a Web look and feel from one of several predefined styles, or you can design your own Web style for use with your applications. The tool creates three Java beans and one JSP per record format; the Java beans hold the data for the record format, or control its appearance or other characteristics, and the JSP handles displaying the Web version of the screen, prompting for data, and handling input errors. The wizard generates an application home page to launch the Web-enabled version of your program.

When a user invokes a converted application from the browser, the WebFacing server on the iSeries system starts the host program. The server intercepts all calls to READ, WRITE, and EXFMT operations to DSPFs, so that in many cases your program (*PGM) can run without modifications, and without even detecting that it is being accessed using WebFacing. You might need to make coding changes if your application uses DDS keywords that are not supported by WebFacing, or if you want to modify the DDS screens so that the conversion to Web format produces a more attractive or consistent result.



First the application is converted. This creates JSPs for each record format as well as Java beans. When the program performs a READ on a record format, control as well as the application data is sent to the Workstation Data Manager.. Because the job was started by the WebFacing Runtime Server, Workstation Data Manager knows this is a WebFacing request and passes the data and control to the WebFacing Runtime Server which runs on the iSeries. Control returns to the Webfacing Runtime Servlet that runs in WebSphere Application Server. The WebFacing Runtime Servlet locates the appropriate JSPs and Java beans. The WebFacing Runtime Servlet tells WebSphere Application Server to return the JSP back to the browser. The JSP is compiled and the resultant HTML is returned to the browser. Note that no 5250 datastream is generated in this flow.

Goals of WebFacing

- **Quick conversion**

Using existing skills

- **Unlimited refinement**

Using existing SDA skills, or using Web skills

- **Cost effective**

Tool part of ubiquitous tool set

Runtime part of operating system

Only pre-req is WebSphere Application Server Express or higher

You can get your RPG/COBOL programs running on the Web in a short time. Conversion is straight forward. After creating a project and specifying the DDS to convert, just choose Convert from the pop-up menu. Typically there is no change required by the host program. After the initial conversion, the new style can be applied to give the application a new look. With knowledge of SDA or JSPs and Cascading style sheets (CSS), you can modify shipped styles to create your own. You don't require other skills such as Java to WebFace an application. There are no additional costs from a previous version.

History of WebFacing

- **WDT V5R1 (May 2001)**
 - First release
 - Five fix packs
- **Development Studio Client V4.0 (June 2002)**
 - Second release
 - Four fix packs
- **Development Studio Client 5.0 (April 2003)**
 - Third release
 - One fix pack
- **Development Studio Client 5.1 (Oct 2003)**
 - Fourth release
 - One fix pack (second due out Feb 2004)

In May 2001 WebSphere Development Studio (WDS) for iSeries—a single package containing RPG, COBOL, C, C++, ADTS and unlimited licenses of WDT was introduced. Further, it introduced WebFacing, the technology for converting green-screens to a Web interface. This phase marked the beginning of a new world where all developers had all tools for modern application development.

The long-term goal has been to collapse the many tools in WDT into a single, integrated tool that can be used for all development. The first phase of that tool was introduced when IBM announced and delivered Version 4.0 of its client tool suite for iSeries, formerly known as WDT in June 2002. With Version 4.0, the suite received a new name—WebSphere Development Studio Client (Development Studio Client). The IBM WebFacing Tool was now integrated into the Development Studio Client workbench.

In version 5.0 and 5.1 of Development Studio Client, additional enhancements were made to the IBM WebFacing Tool.

Table of contents

e-business Primer

What is WebFacing?

□ □  **WebSphere Development Studio Client**

WebFacing Tools

Customizing WebFacing

What's new in WebFacing 5.0 and 5.1.0

Summary

Previously we introduced WebFacing and reviewed the goals and history of WebFacing. Now lets introduce Development Studio Client.

WebSphere Development Studio Client

- Strategy
- Packaging
- Tools

This section introduces the strategy behind Development Studio Client including the new packaging and new tools.

iSeries AD Strategy

New Packaging

WebSphere Development Studio

New Technology

Eclipse, Web, XML, WebServices, etc

New Tools

WebSphere Development Studio Client


New User Interface

Web

Here we see IBM's strategy is to reach its goals of building community and building excitement through 4 areas: New Packaging, New Technology, New Tools and New User Interface. The new packaging was actually done in May 2001, with the introduction of WebSphere Development Studio for iSeries, which as we will see offers a single product with all the host and client tools needed for all application development needs, from traditional to e-business.

With the 4.0 release of the client tools in June 2002, we introduced new Eclipse technology and totally new Eclipse-based tools. Further, for technology we also plan to keep up with technology both in the compilers and the tools, so iSeries programmers are very current. This was evident in the V5R2 release of the compilers in Sept 2002, and will be again in the V5R3 release of the compilers. With the April 2003 new 5.0 release of the client tools, we are in the second release of the eclipse-based technology and continue to improve the tools and introduce new technology.

The preferred user interface for iSeries applications is now a Web browser, versus a 5250 green screen or even a client/server GUI. There are exciting new and enhanced tools to make it easy to convert 5250 user interfaces into Web user interfaces, and to build new Web user interfaces on top of new or existing business logic.

IBM Software Group | WebSphere software 

WebSphere Development Studio V5

www.ibm.com/software/awdtools/wds400

RPG	COBOL
C/C++	PDM, SEU SDA, RLU

V5R1 or V5R2
5722-Development Studio
No-cost V4 Upgrade: 5905 (V5R1) or 5913 (V5R2)

Unlimited Licenses

iSeries	iSeries	iSeries	iSeries	Web Facing	iSeries Projects
Java	Debug	Struts Web	Web Service	App Server	RSE
Trace	Profiling	DB	XML		

+CODE
+VisualAge RPG

WebSphere Development Studio Client V5= WSSD 5.1.0 + iSeries Extensions

18 | **iSeries WebFacing** | IBM eServer iSeries Application Development | © 2003 IBM Corporation

There is now only one application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and unlimited licenses of the workstation-based toolset named WebSphere Development Studio Client (formerly WebSphere Development Tools).

If you are an existing customer who has a subscription, you can upgrade to Development Studio free of charge. Without a Software Subscription, there is an upgrade fee. New licenses of Development Studio are priced very competitive compared to the combined prices of all constituent products. As of V5R1, there is no way to purchase the compilers or tools individually. So if you have RPG at V5R1 or higher, you must have Development Studio and hence are entitled to Development Studio Client.

For consultants who do not have an iSeries of their own, but still wish to have the client tools, Development Studio Client is also made available as a passport advantage product so it can be purchased "off the shelf" from IBM Direct.

Development Studio has been a huge success, with over 80,000 licenses sold. Just as every development machine used to have PDM and SEU, every development machine will now have all the modern Application Development tools from IBM. This ubiquity is especially important for business partners who build and sell software. These Business Partners are now free to build software using any of the technologies or tools in Development Studio, and can assume their customers will have the tools required to tailor everything from RPG to Java and Web user interfaces. This effectively raises the lowest common denominator to a level unparalleled by any other operating system.



WebSphere Development Studio Advanced V5

www.ibm.com/software/awdtools/wds400

NEW!

Unlimited Licenses

RPG	COBOL
C/C++	PDM, SEU SDA, RLU

V5R1 or V5R2
5722-Development Studio
Cost Upgrade: 5906
(V5R1) or 5914 (V5R2)

iSeries	iSeries	iSeries*	iSeries	Web Facing*	iSeries Projects
Java	Debug	Struts Web	Web Service		RSE
Trace	Profiling	DB	XML	App Server	EJB* J2EE*
					Test Cases*

+CODE
+VisualAge RPG

WebSphere Development Studio Client Advanced V5 = WSAD 5.1.0 + iSeries Exts

The Advanced edition of Development Studio Client, and Development Studio, is new as of April 25th 2003. Development Studio Advanced is currently the same as Development Studio, except the customer is entitled to unlimited licenses of Development Studio Client Advanced versus just Development Studio Client.

The difference of Development Studio Client Advanced over Development Studio Client is that it has additional tools (blue boxes with asterisks) and some enhancements to existing iSeries tools (green boxes with asterisks) which will be discussed later in this presentation. The majority of the new tools are related to Enterprise Java Bean (EJB) development. You'll see later that Development Studio Client Advanced is based on WebSphere Application Developer (Application Developer) versus WebSphere Studio Site Developer (Site Developer).

WebSphere Development Studio Client

- *Tools for ebusiness and your business*

For **RPG and COBOL**

Remote System Explorer and iSeries Projects perspectives

For **WebFacing**

WebFacing perspective

For **Java**

Java perspective and iSeries enhancements

For **Web Services**

Web Services perspective and iSeries enhancements

For **Web**

Web perspective and iSeries enhancements

For **XML**

XML perspective

For **Database**

Database perspective

You will most likely be working in the **WebFacing perspective** to quickly convert classic application's UI components like DDS, help, and message files into Web GUI components. A perspective is a specific arrangement of views and tools in the workbench, depending on what role you have, you will use a different perspective. The **Remote Systems Explorer (RSE)** perspective in the workbench is for PDM-like access to OS/400® resources, and iSeries Projects for disciplined team development that leverages the excellent team capabilities of Eclipse. A Java developer who wants to develop Java applications or applets with Interfaces to iSeries classic applications will use the **Java perspective**. A Web developer who wants to use the Web tools iSeries Extensions to develop and maintain Web GUIs to iSeries applications will use the **Web perspective**. An XML developer who wants to work in the Visual XML Development Environment will use the **XML perspective** and the Database developer who wants access database tools will use the **Database perspective**.

Table of contents

e-business Primer

What is WebFacing?

WebSphere Development Studio Client

□ □  **WebFacing Tools**

Customizing WebFacing

What's new in WebFacing 5.0 and 5.1.0

Summary

Previously we introduced Development Studio Client. Now lets look at how you create a WebFaced application, test it and deploy it.

WebFacing Scenario

1 Create new WebFacing project

Specify members to convert, look-and-feel style, how to call/invoke the application

2 Convert the project

Creates Web stuff from the DDS and UIM source

3 Run it!

Select "Run on Server" to run in built-in WebSphere

4 Refine it

Using Web Settings in CODE Designer (launched from WebFacing)

Using project's Properties dialog

5 Repeat

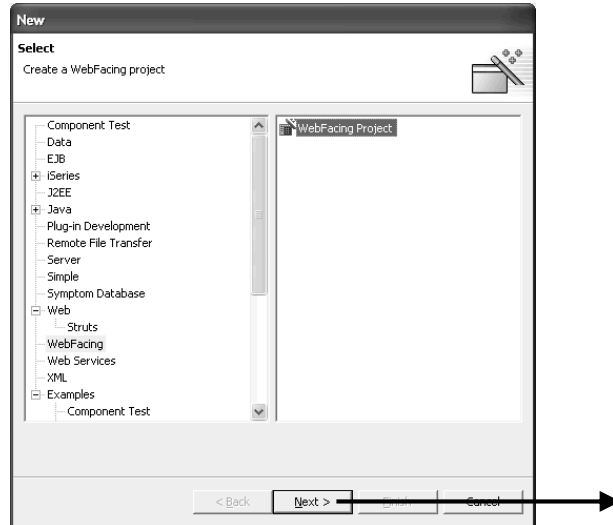
6 Export as EAR file, import into WebSphere Application Server on iSeries

Or use built-in server-tools support to publish to WebSphere Application Server on iSeries

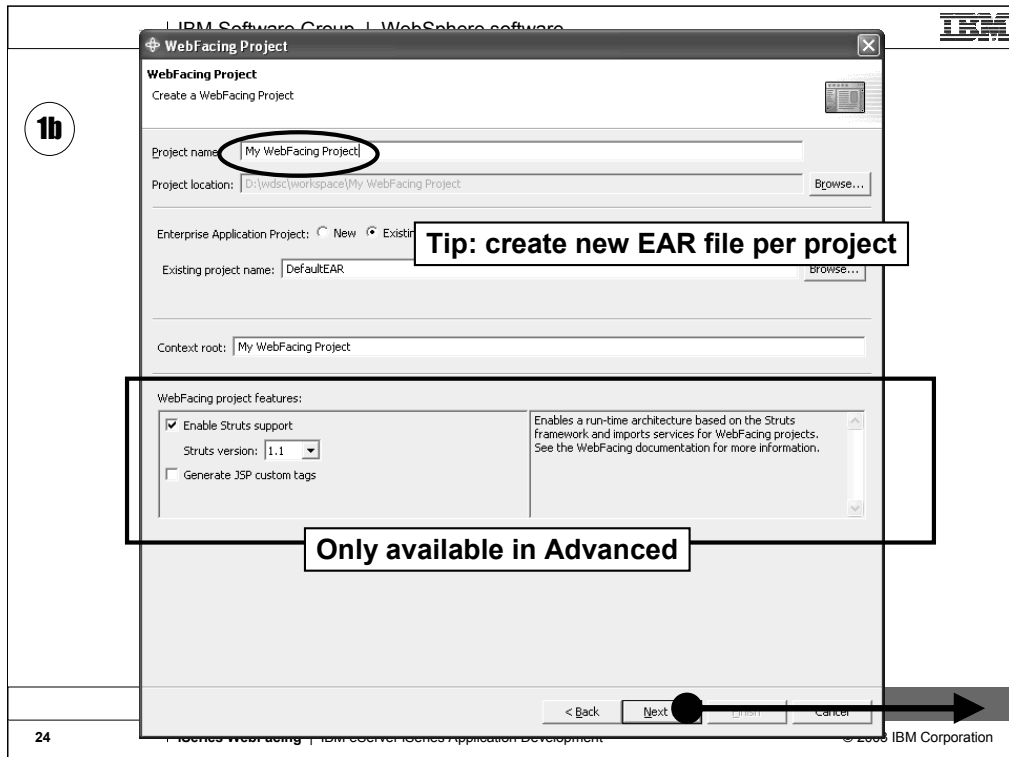
Here is a typical WebFacing scenario. After you start the workbench and open the WebFacing perspective, you create a WebFacing project. When you create this project you select the DDS source on the server that you want to convert, the CL command to start the application and the text for the Web page link and the style you want to use for your new Web pages. Next you convert the DDS to JSPs, beans. After conversion you are ready to run your JSPs using the Run on Server option which starts the WebSphere Test Environment, a local copy of the built-in WebSphere Application Server. Now you can iteratively improve the look of your new Web pages by using the Web Settings in CODE Designer and the Web Project's Properties dialog. You can then re-test your changed WebFaced application. You can continue to refine it and retest your new Web pages until you have a design that you like. Finally you can move the new WebFaced application to a production server.

1a Create a New WebFacing Project

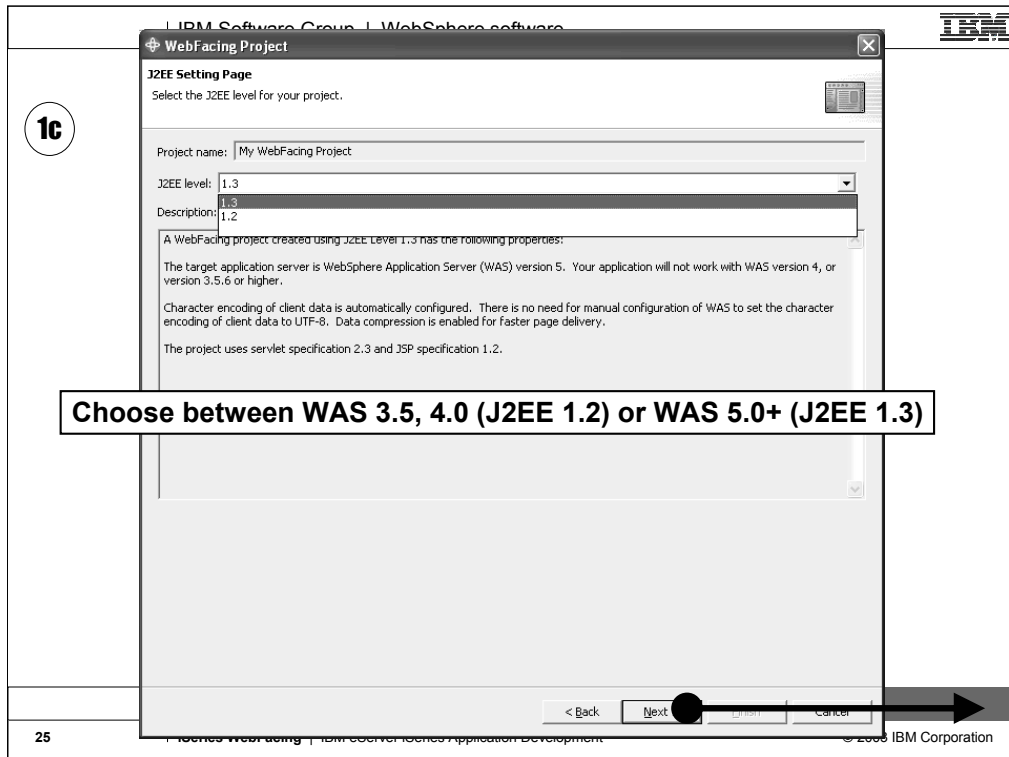
Use File->New->Other



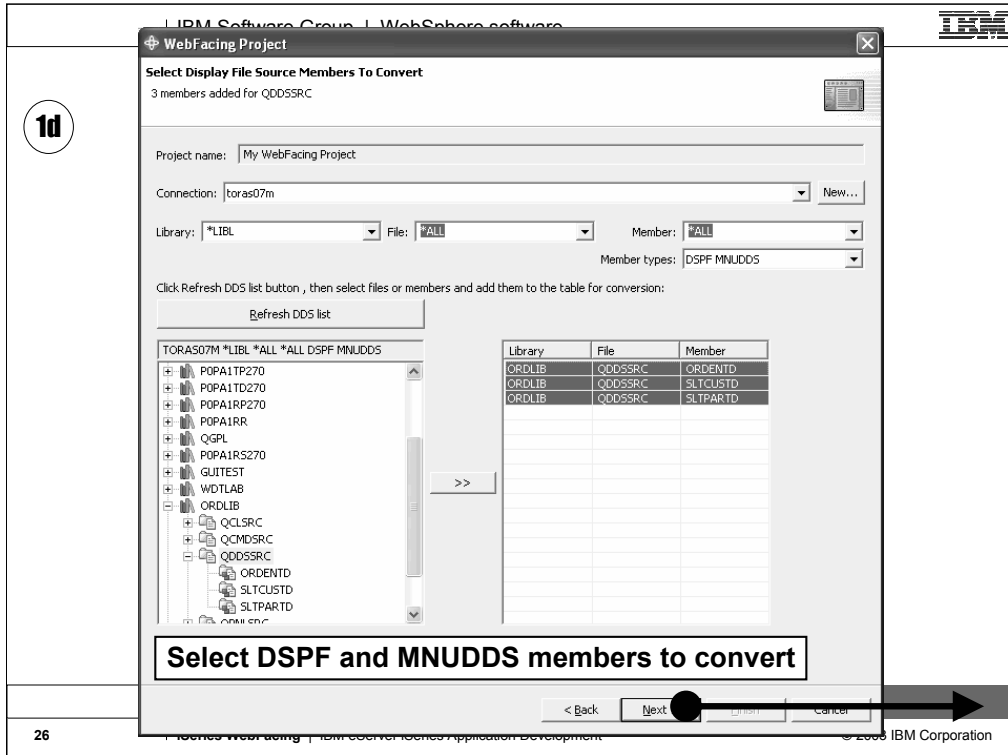
To begin, you create a new project. If WebFacing doesn't appear in the menu you select Other and then select WebFacing from the next dialog.



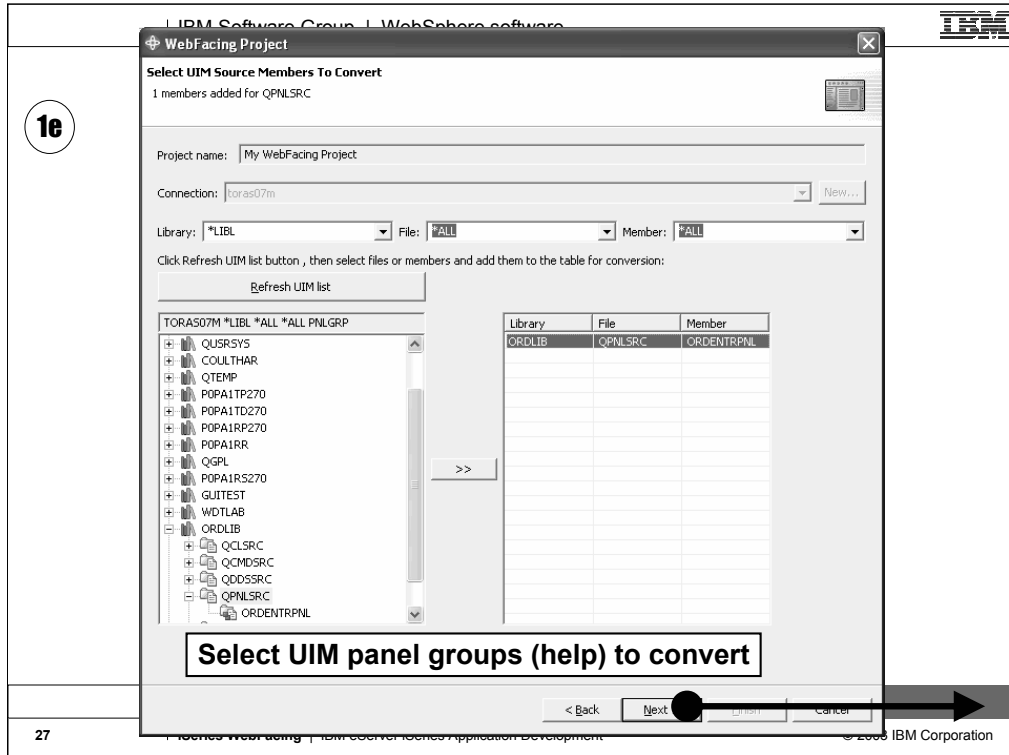
On this page of the wizard you give the project a name and specify where the files are to be stored. The default is to store them in the workspace directory. Web applications (which WebFACING is based on) are part of an EAR (Enterprise Archive) file, so we need to specify the EAR file name. The Context root is what you would specify as part of the URL in the browser to invoke the application. In Development Studio Client Advanced Edition you also can Enable Struts support. In doing so you comply with the Model-View-Controller paradigm. If you want to write a custom tag library for use with WebFACING, you would use the Generate JSP custom tags selection. This is an advanced edition feature only. What this means is that at conversion time the process of converting the DDS into JSPs can be influenced by user-written classes to change the JSP source that is emitted for fields.



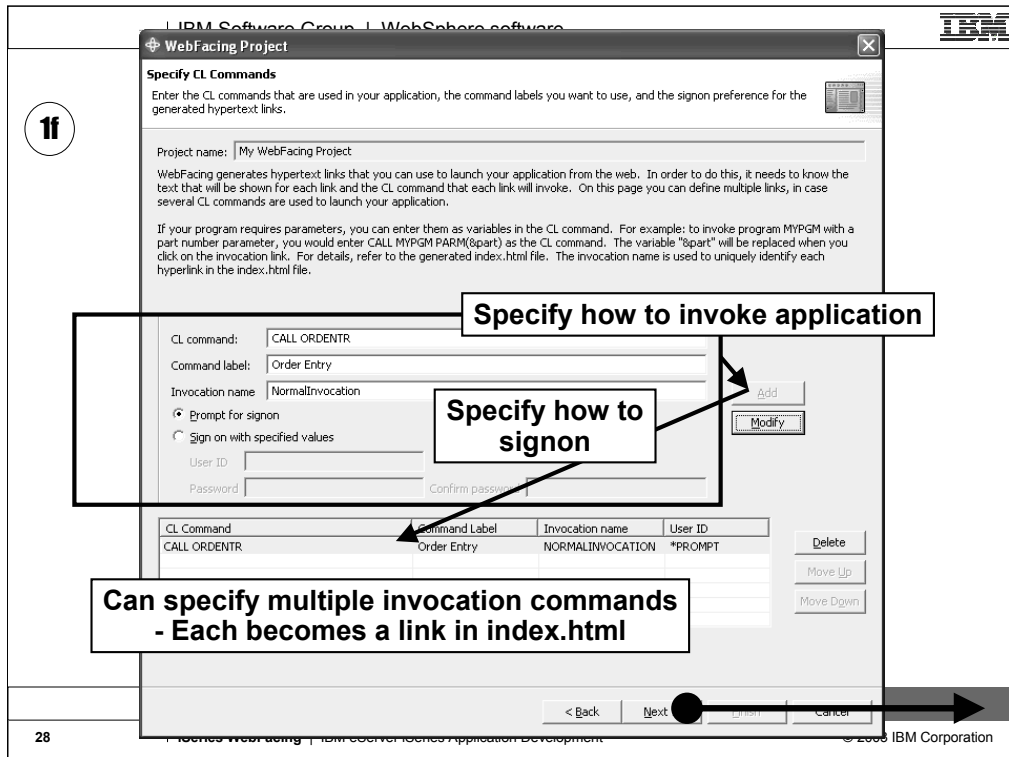
On this page of the wizard you to specify the J2EE level you want for the Web project to be created. You might wonder what that means to you. If you plan to use the WebSphere Application Server version 5 only then you should select J2EE level 1.3. If you are using previous versions of WebSphere Application Server in your environment and are planning to deploy the Web application to these versions then you should use J2EE level 1.2 since these older versions of WebSphere Application Server don't support the J2EE level 1.3. In this exercise you will use the Web application in the WebSphere Test Environment, which is version 5, so you will use J2EE level 1.3.



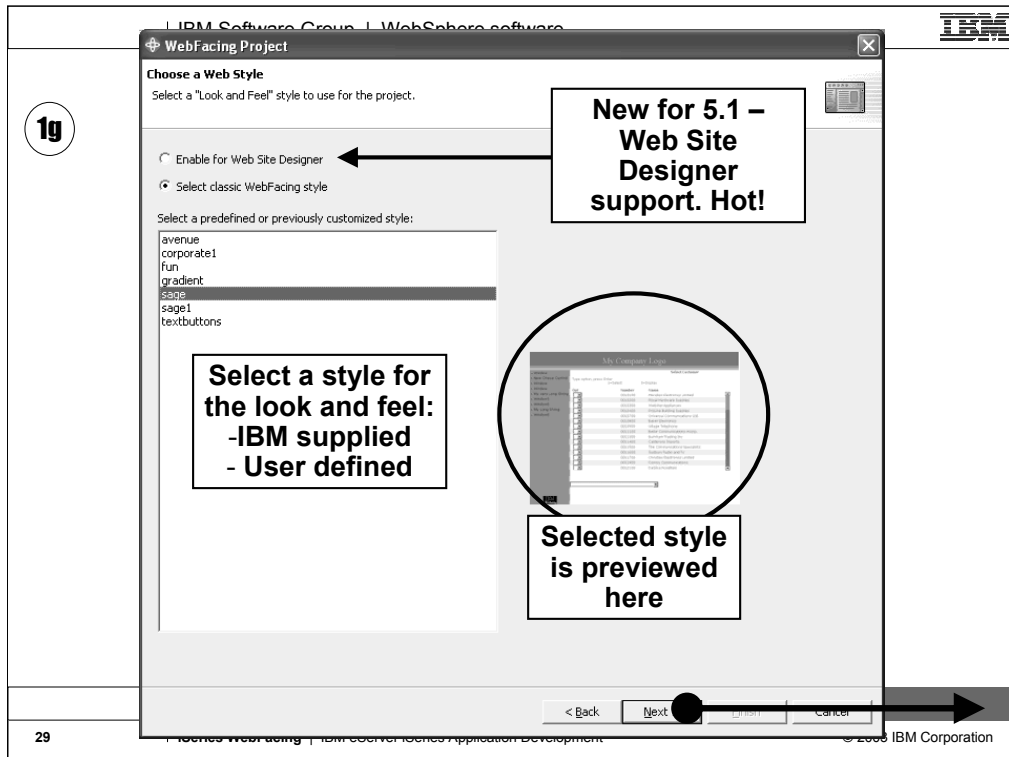
On this page of the wizard you select which connection (iSeries) contains the DDS. The member types allows you to select DSPF and/or MNUDDS (for menus). WebFacing supports menus. If you click Refresh list you will retrieve the library list, after you have signed on. From the library list you drill down to find the DDS source members you want to convert. When you click the >> push button, the selected members are moved to the list of members to be converted. If you click the folder all the members in the folder are moved to the list of members to be converted. In this case you don't need to select each member.



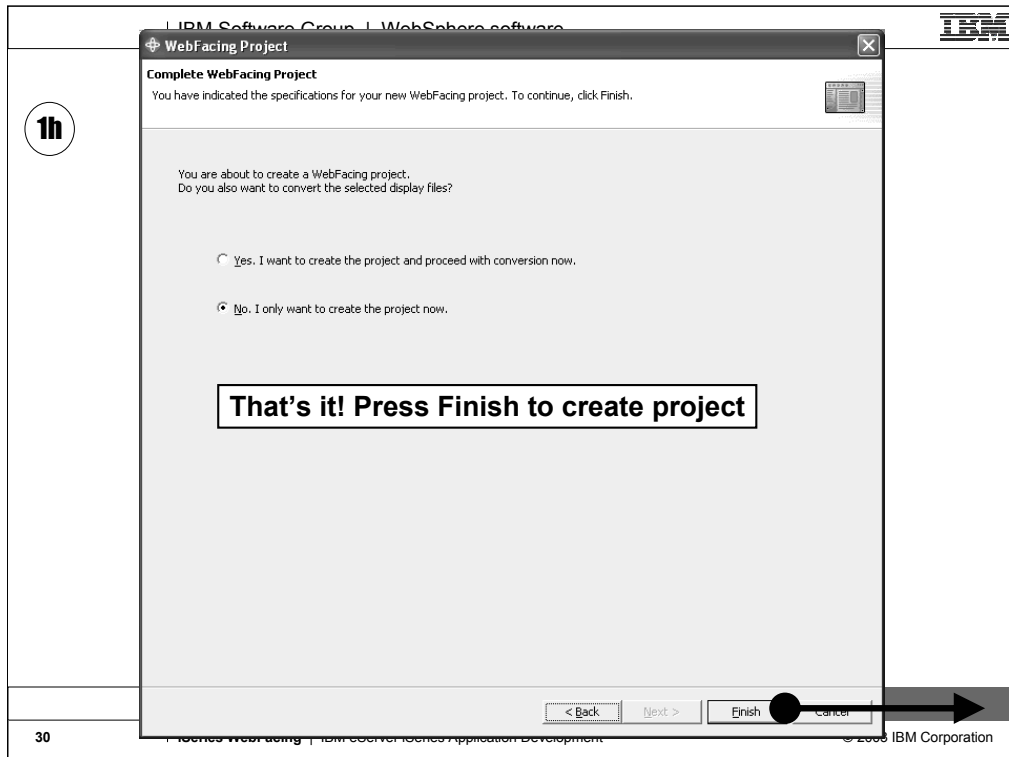
On this page of the wizard you select the UIM help members you want to convert. Again you drill down to locate the members.



On this page of the wizard you specify the command to call the application. The Command label text is the text that will appear on the generated Index.html page to launch the application. The CL command is an iSeries command to start the program. Typically you just do a CALL to the program. If the program has parameters you specify them here. Specifying parameters results in input fields being generated on the index.html page where the user would enter their value. If Prompt for sign-on is selected, an authentication dialog (login.jsp) is displayed before the application is launched. Alternatively you can specify a User ID and password to be passed to the WebFacing server before the program is started. Either way, the User ID and password is used to start an interactive job on the iSeries and invoke the specified command.

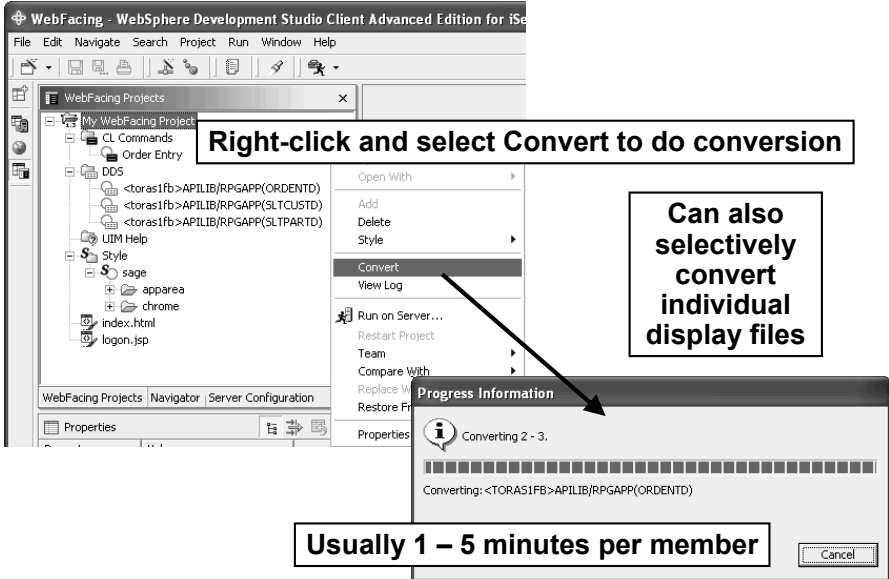


On this page of the wizard you choose the style for your application. After the project has been created another style can be applied without performing conversion. After conversion the WebFacing Tool WebSettings option or the Properties dialog allows you to customize one of the pre-defined styles and save it as a new style that can then be used by other projects.



On this page of the wizard you choose just to create the project and perform conversion at a later time, or create the project and perform conversion when Finish is pressed. You click Finish to complete the creation of the WebFACING project

2



Right-click and select Convert to do conversion

Can also selectively convert individual display files

Usually 1 – 5 minutes per member

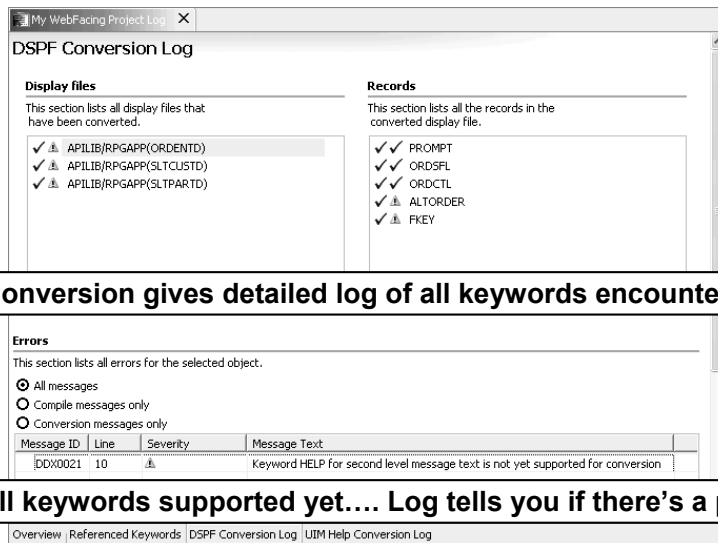
31

iSeries WebFacing | IBM eServer iSeries Application Development

© 2003 IBM Corporation

To perform conversion, right click on the DDS folder and choose convert. This will convert all members in this folder. If you made changes to a single member, you could just select that member to convert. Note that the source remains on the iSeries. The conversion process reads the DDS members and performs conversion in memory. Conversion creates 2 JSPs and 3 Java beans for each record format. It is important to convert all DDS members that are part of the application. If a member is not converted you will get a 'class not found' message at run-time.

2



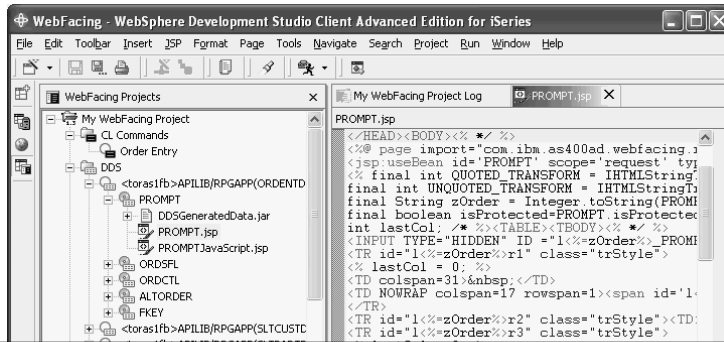
Conversion gives detailed log of all keywords encountered

Not all keywords supported yet.... Log tells you if there's a problem

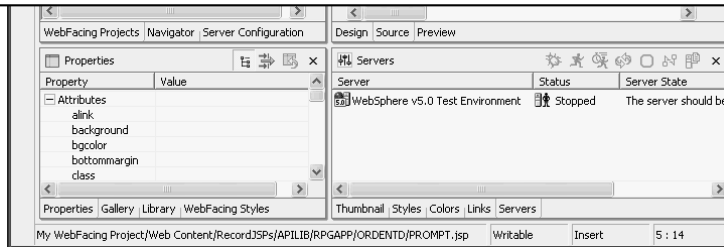
See product help for list of unsupported keywords

Once conversion is complete, the conversion log is displayed in its own view. Note the tabs at the bottom of the log view to see different parts of the conversion log. Review the conversion log to be sure all members were converted. A red X symbol indicates conversion failed for that member. No JSPs or beans were created. See the Message Text to determine the cause of the failure and fix as necessary.

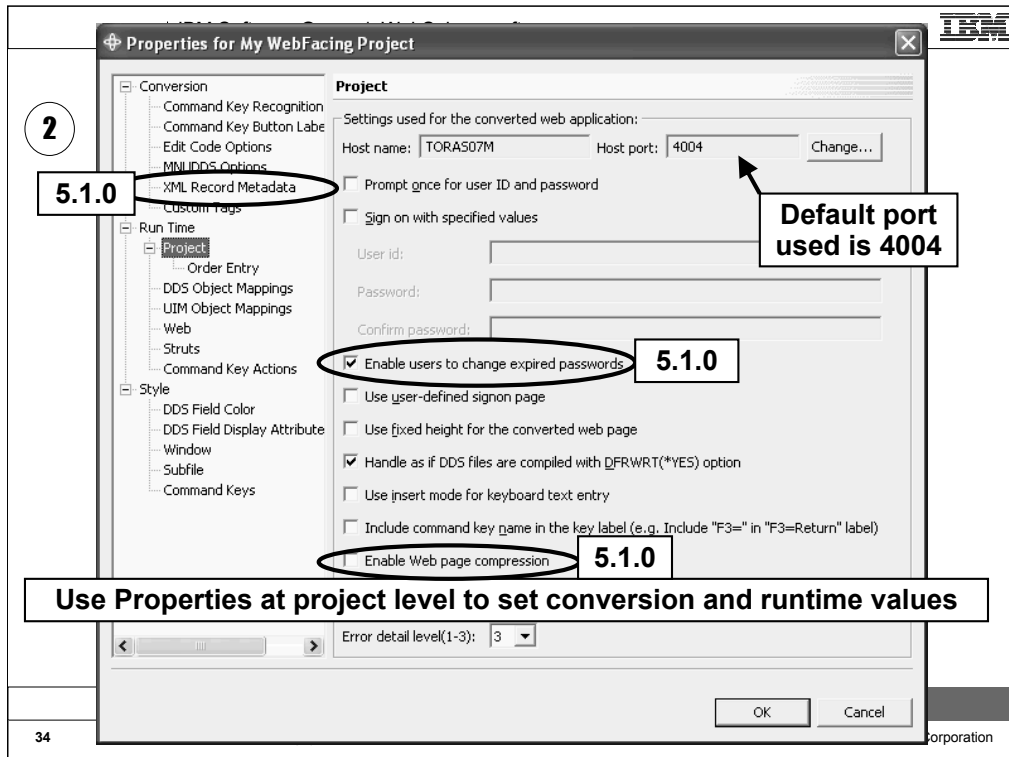
2



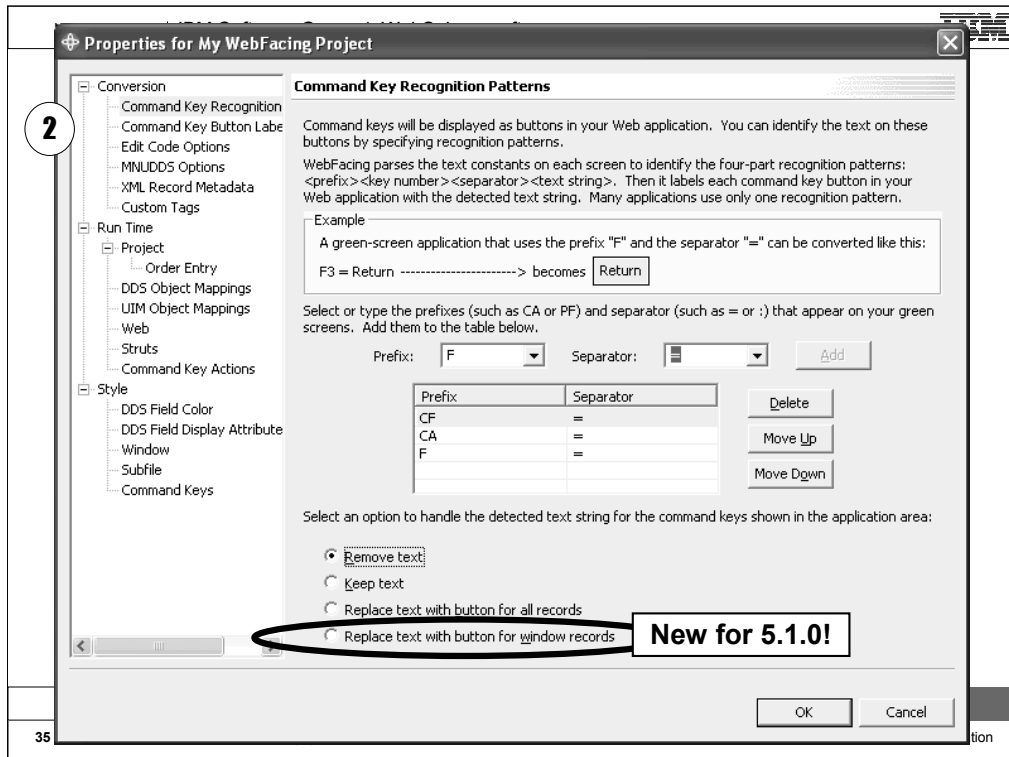
After conversion, you can see generated files per record format



By switching to the WebFacing Projects view you can see the directory structure of the project after conversion. Note that there are 2 JSP files created for each record format. To see the generated JSP, double-click it to open it in Page Designer.



To make changes to the project right-mouse click the project and choose Properties to get the Properties dialog. The Host Port number is the port number the WebFacing run-time uses to invoke the program on the host. 4004 is the default and can be changed on the host with the WRKSVRTBLE command. If you change it on the host be sure to change it here also. You can indicate that the user will be prompted to sign-on to the application or sign-on will occur using the User ID and password specified here. By checking Retain user ID and password, the user will not be prompted to sign-on for subsequent calls to the program in the same session. If this is left unchecked, the user will be prompted each time they invoke the program. The JSP level indicates the amount of tracing detail to put in the log file. 3 outputs the most detail. The log is used by developers for debugging.



A new property for release 5.1, which was highly requested, allows the function-key buttons to be placed within the window for windowed records only, while remaining on the left for non-windowed records.

3

WebFacing - My WebFacing Project Log - WebSphere Dev

File Edit Navigate Search Project Run Window Help

WebFacing Projects

My WebFacing Project

- CL Commands
- DDS
 - <toras07m>ORC
 - PROMPT
 - DDSGene
 - PROMPT.
 - PROMPT.
 - ORDSFL
 - ORDCTL
 - ALTORDER
 - FKEY
 - <toras07m>ORC
 - <toras07m>ORC
 - UIM Help
 - Style
 - index.html
 - login.jsp

WebFacing Projects Navigator

Properties

New

- Open
- Open With
- Add
- Delete
- Style
- Convert
- View Log
- Debug on Server...
- Run on Server...
- Profile on Server...
- Restart Project
- Team
- Compare With
- Replace With
- Restore from Local History...
- Properties

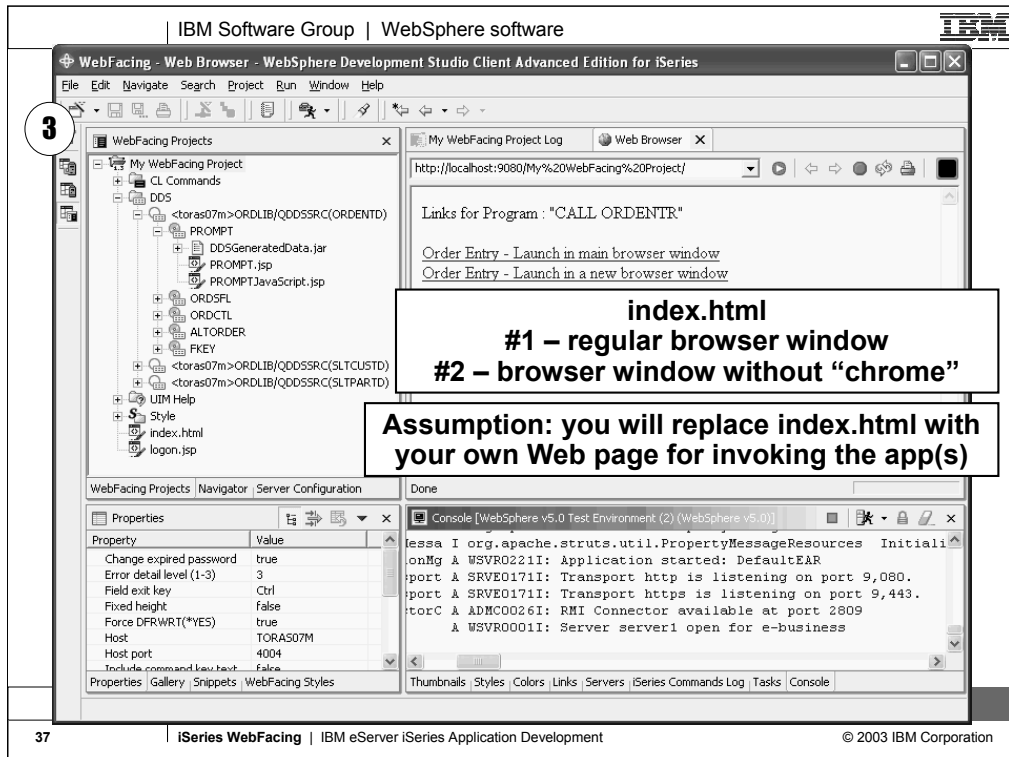
Time to run it!

Starts index.html

Run On Server runs Web app in built-in WebSphere Test Env

36 | iSeries WebFacing | IBM eServer iSeries Application Development | © 2003 IBM Corporation

Once the application has been converted it can be tested in the built-in WebSphere Test Environment. Locate the index.html page and choose Run on server. If not yet started, the WebSphere Test Environment will start. The index page will appear in the internal browser view.

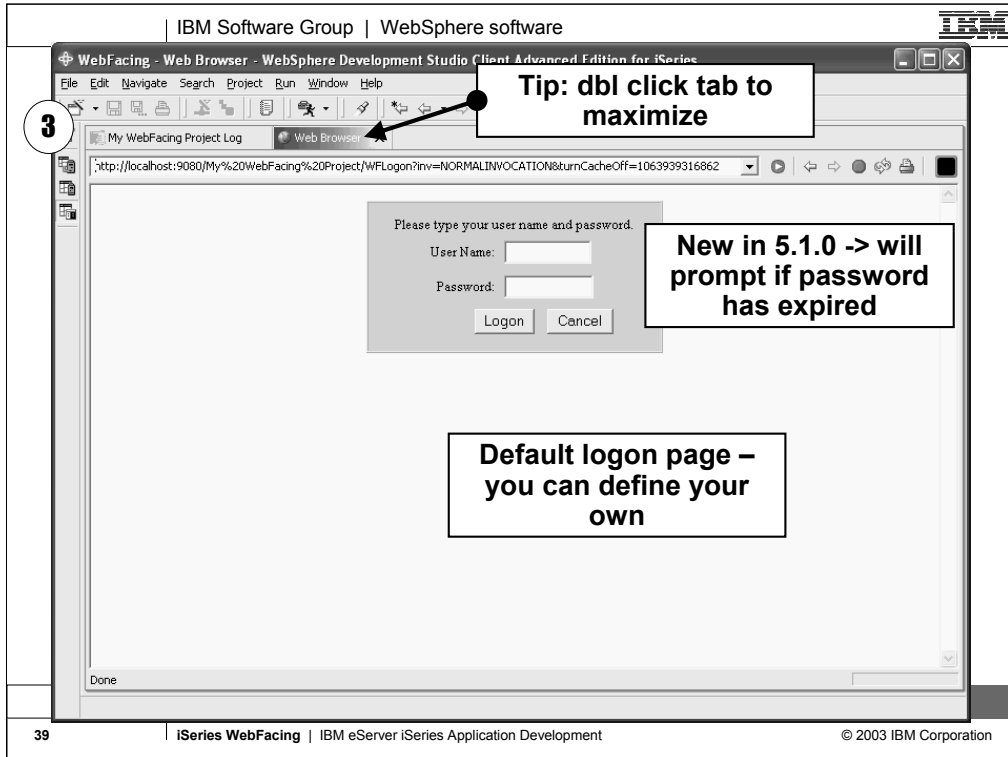


This is the index.html page that was generated based on your choices during conversion or by settings in the Properties dialog. If you click on a link, the WebFacing run-time servlet will be invoked (in WFRun.jar). Passed as a parameter is the invocation file that contains the system name and the name of the program to call. To see how WebFacing is invoked choose View/Source from the browser window.

Invocation Sequence

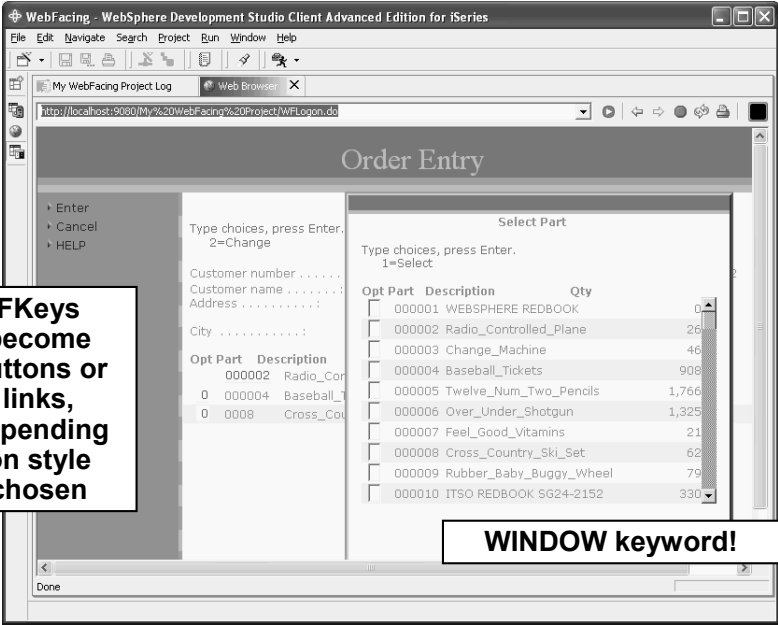
- Generated link calls WebFacing Servlet runtime
 - Extracts CALL information from web.xml
 - Extracts user ID information from web.xml, or prompts for it
 - Starts server job, in WebFaced mode
 - Calls/invokes application
 - Application runs, writing dspf formats and then reading
 - Workstation Data Manager passing data to WF Server
 - Which passes it on to WebFacing Servlet
 - Which puts it in a bean and passes it to the JSPs
 - All written JSPs up to the read are collected into one
 - User sees screen, presses Enter
 - Control goes back to WF Servlet, which passes to WDM
 - WDM passes data back to application as usual

Here you see the invocation sequence. The generated link calls the WebFacing Runtime Servlet which extracts the call information from web.xml. Then the user ID information is extracted from web.xml or user ID information is prompted. The server job is started in WebFaced mode. This calls the application. The application runs and writes dspf formats and then reads. The Workstation Data Manager passes data to the WebFacing Runtime Server which passes it to the WebFacing Runtime Servlet which puts it in a bean and passes it to the JSPs. The user sees the screen and presses Enter. Control returns to the WebFacing Runtime Servlet which passes to the WorkStation Data Manager. Workstation Data Manager passes data back to the application.



The Sign-on dialog opens. This is default sign-on page. You can design your own sign-on page.

3



FKeys become buttons or links, depending on style chosen

WINDOW keyword!

40 | iSeries WebFacing | IBM eServer iSeries Application Development | © 2003 IBM Corporation

Here's the result of a WebFaced application. running in a browser. In this case, the DDS used the WINDOW keyword. In that case windows are created that can be moved around the browser. Notice that subfiles are fully supported. If you click the up/down arrows in the subfile scroll bar, this action will navigate the subfile.

WebSphere Test Environment

▪ WebSphere Test Environment (WTE)

A full copy of **WebSphere Application Server** is embedded:

Instant/easy testing of Web projects within Development Studio Client

Standalone all-in-one testing

No dependency on WAS installation or availability

Easily debug prior to publish/deploy

Supports configuring multiple Web apps

Access to WebSphere Application Server Administration Client

▪ Server Tools

Supports start, stop, restart, publish actions for WTE, local or remote WebSphere Application Server or local TomCat

When using "Run on Server" you are by default running the full fledged copy of WebSphere Application Server that is built into Development Studio Client. The purpose of this is to enable extremely easy testing of your Web application prior to publishing or deploying to your test or production server. The Server perspective that "Run on Server" switches you to is in fact quite rich in its function, allowing starting, stopping and publishing of the server through the right click popup menu. You can even bring up the full WebSphere Application Server Administration console directly from this perspective. Very handy! In addition to managing the built-in copy of WebSphere Application Server, this Server perspective can be configured to manage any local copy of WebSphere Application Server or TomCat, or any remote copy of WebSphere Application Server including WebSphere Application Server on iSeries. You can associate any Web project with any such local or remote WebSphere Application Server, and it will automatically be published and configured there for you when you "Run on Server" or explicitly "publish"! There is simply nothing more productive for Web Development than the Server perspective. You cannot imagine living without it after using it. Be aware though, that WebSphere Application Server requires at least 512MB of memory, so you will need this much to use this awesome function effectively.

6 Deploying to Remote WebSphere

- **Use File->Export to export EAR file**

 - Export to a mapped IFS drive

 - Use WebSphere Administrator Console to import EAR file

 - Bring up Browser, enter URL to start application

- **Or, configure a server in Server Tools for iSeries WebSphere Application Server** and use Publish to push to it, and then Run On Server to run it

 - This requires a PTF still coming at time of press

 - This also replaces the server's configuration file

 - So use a dedicated WAS instance for testing

- **Tip:** look at Change Management tools like SoftLanding's to automate delta deployments

Once the application has been tested and is running properly it is deployed to the production WebSphere Application Server.

Choose File Export and specify the EAR file name you specified when you created the project. Once the EAR file has been deployed, use the WebSphere Application Server Administration console to install the application in WebSphere Application Server.

Table of contents

e-business Primer

What is WebFacing?

WebSphere Development Studio Client

WebFacing Tools

□ □  **Customizing WebFacing**

What's new in WebFacing 5.0 and 5.1.0

Summary

Previously we reviewed a typical WebFacing application scenario including application testing and deployment. Now lets look at customizing a WebFaced application.

3 Ways to Customize WebFaced Applications

1 By Properties

Much can be affected in the conversion and runtime, including overall style, by use of properties and style wizard

2 By Web Settings

These are DDS comments for affecting the conversion results.
Easy to set with the CODE Designer tool

3 By Editing the Output

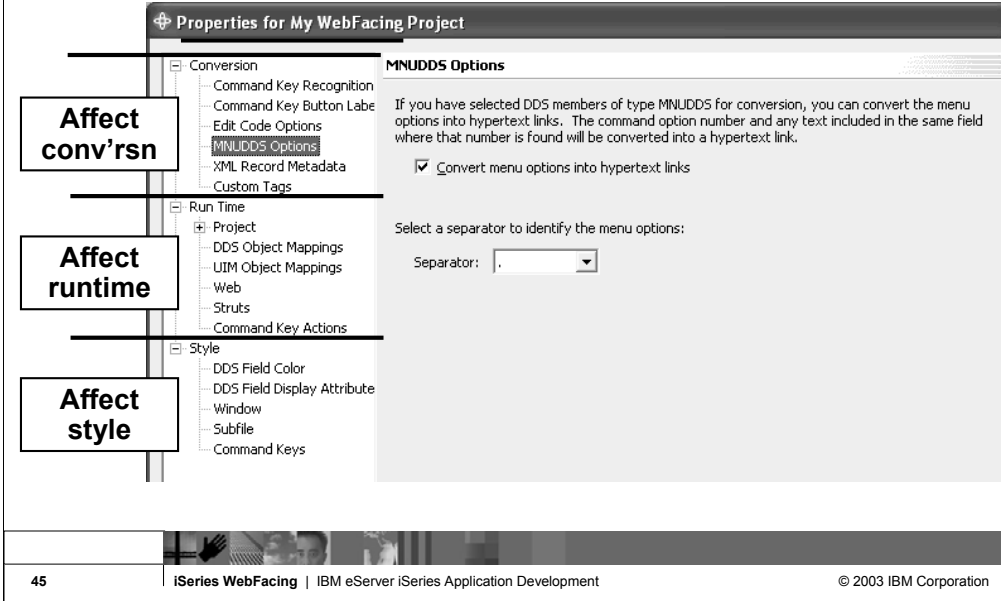
Because of the built-in Java and JSP editors, you can always edit what WebFacing generates

But be careful....

There are 3 ways to customize WebFaced applications.

1. Before conversion, use properties dialog to change conversion, runtime and style properties.
2. Before conversion use the WebSettings feature of CODE designer. WebSettings are stored as comments in the DDS. With WebSettings you may have certain fields that may not be relevant on the Web. You can specify that a field contains an image name, and an tag should be created for that field. You can specify that a field is a hyper-link. Clicking on the link at run-time will launch the URL specified in WebSettings.
3. After conversion, WebFacing copies the Cascading Style Sheet (CSS) to the project. With some knowledge of CSS you can modify it to customize your pages. You can use Page Designer to edit the created JSPs. Note, conversion will overwrite customized JSPs

1 Customizing By Properties



Affect conv'rsn

Affect runtime

Affect style

Properties for My WebFacing Project

- Conversion
 - Command Key Recognition
 - Command Key Button Label
 - Edit Code Options
 - MNUDDS Options**
 - XML Record Metadata
 - Custom Tags
- Run Time
 - Project
 - DDS Object Mappings
 - UIM Object Mappings
 - Web
 - Struts
 - Command Key Actions
- Style
 - DDS Field Color
 - DDS Field Display Attribute
 - Window
 - Subfile
 - Command Keys

MNUDDS Options

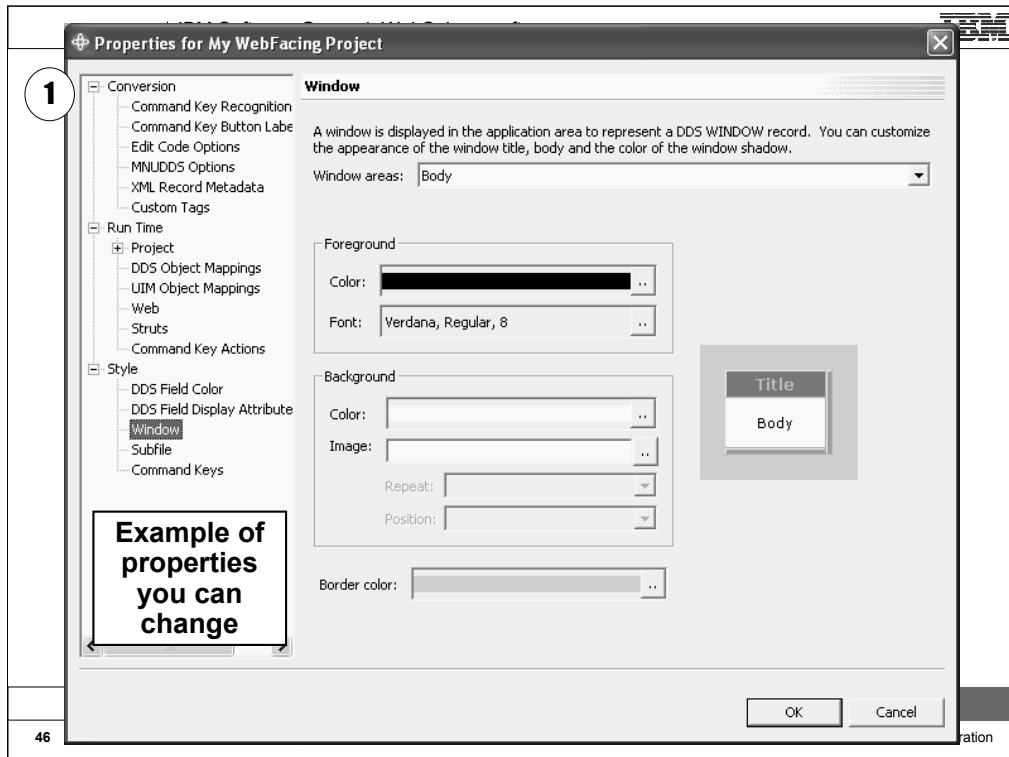
If you have selected DDS members of type MNUDDS for conversion, you can convert the menu options into hypertext links. The command option number and any text included in the same field where that number is found will be converted into a hypertext link.

Convert menu options into hypertext links

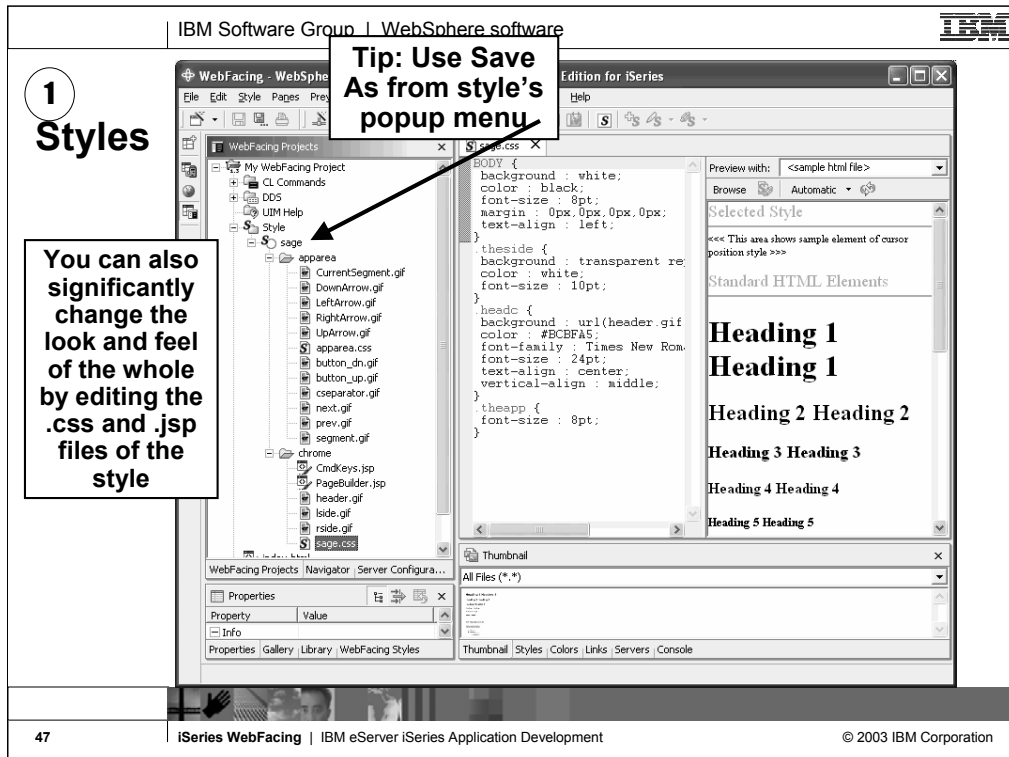
Select a separator to identify the menu options:

Separator: .

To edit the properties for a WebFacing project, in the WebFacing Projects view, right-click the icon for your Project, CL commands, DDS, UIM Help, or Style folder and select **Properties**. Conversion properties control how selected DSPF and UIM files are being converted for WebFacing use. Values for conversion properties are stored in the file conversion.rules under the config directory of the WebFacing project. Run-time properties determine the behavior of the converted Web application when it is being used by an end user. You can customize the look of the application area and the command keys using the Style properties. If you want to change the look of the layout and frame surrounding these areas, you must use a CSS editor to update the style files stored in the chrome directory.



Here you see an example of changing the style property Window. You use the Window screen to indicate how you would like DDS WINDOW records to look when they are converted for Web use. You can customize the look of the title, body, and shadow of the window.

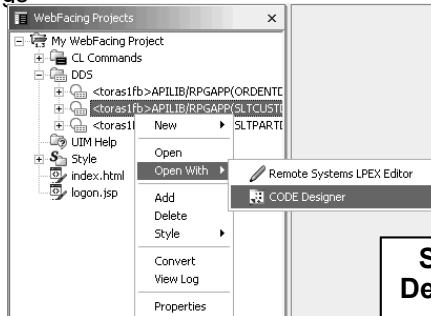


If you want to edit the style that is used for the application area and command key elements, you can use the Style properties pages or you can edit the apparea.css file directly using tools supplied in the IDE. The Style properties pages make it easier to visualize the modifications that are being made and shows you how these changes apply to the DDS elements such as window and subfile records. The changes made through the Style properties pages are then applied to the corresponding style class names in the apparea.css file.

If you want to edit the layout and the frame surrounding the application area and command keys, edit the user-defined CSS files in the \chrome directory. To use the IDE's CSS editor, right-click the file that you want to edit and choose **Open with --> CSS Designer**. If you want to edit the layout of the frame, edit the file PageBuilder.jsp. To edit the frame style, right-click **Style --> Edit Style**. PageBuilder.jsp can then be edited using the Page Designer tool supplied by the IDE. The CSS file in the \chrome directory can be edited using the CSS Designer in the IDE. The Web perspective can be useful when editing CSS files. To open the Web perspective, select **Window --> Open Perspective --> Other --> Web**.

Customizing

2 By Web Settings



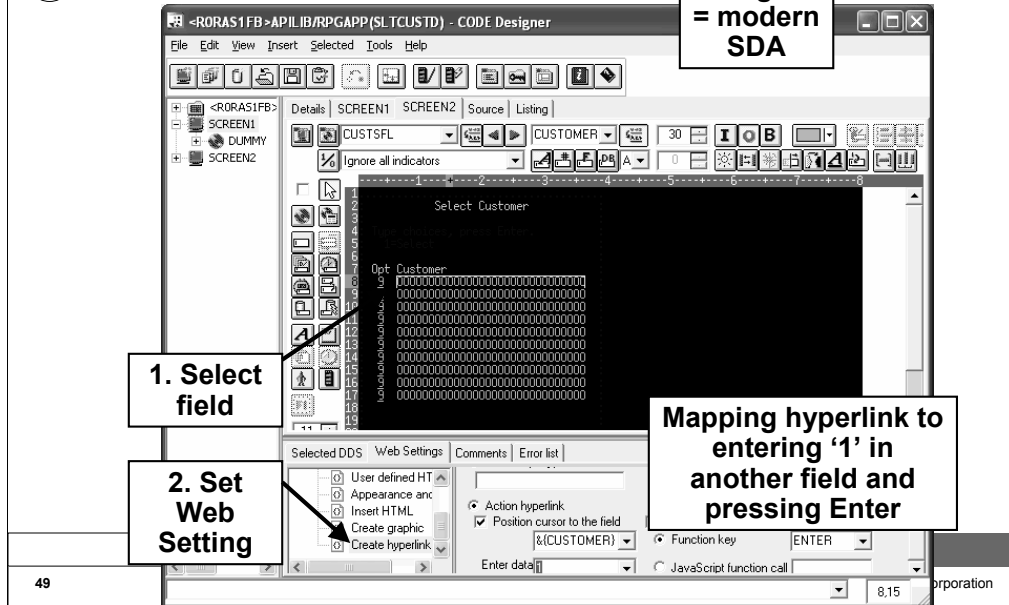
**Start CODE
Designer from
within
WebFacing
project**

You can use the Web Settings tab in CODE Designer to customize how your programs will look and function when accessed through a Web browser. Programs that use DDS source to describe 5250 display screens can be accessed with a Web browser after the DDS source has been converted with the WebFacing Tool. Use Web Settings when you want to manipulate the Web presentation of individual screens and individual fields within screens. If you want to change the Web presentation of multiple screens or of an entire WebFacing project, use Style properties.

Web Settings enable you to affect how your pages will appear before they are created using the WebFacing wizard. If you would like to modify your pages after conversion, you can use a text editor or a Web design tool such as WebSphere Studio. An advantage to customizing your pages with Web Settings is that the instructions for the customizations are embedded as comments in your DDS source; since Web Settings become part of your source, changes that you make are not lost if you later reconvert a WebFacing project.

2 Customizing By Web Settings

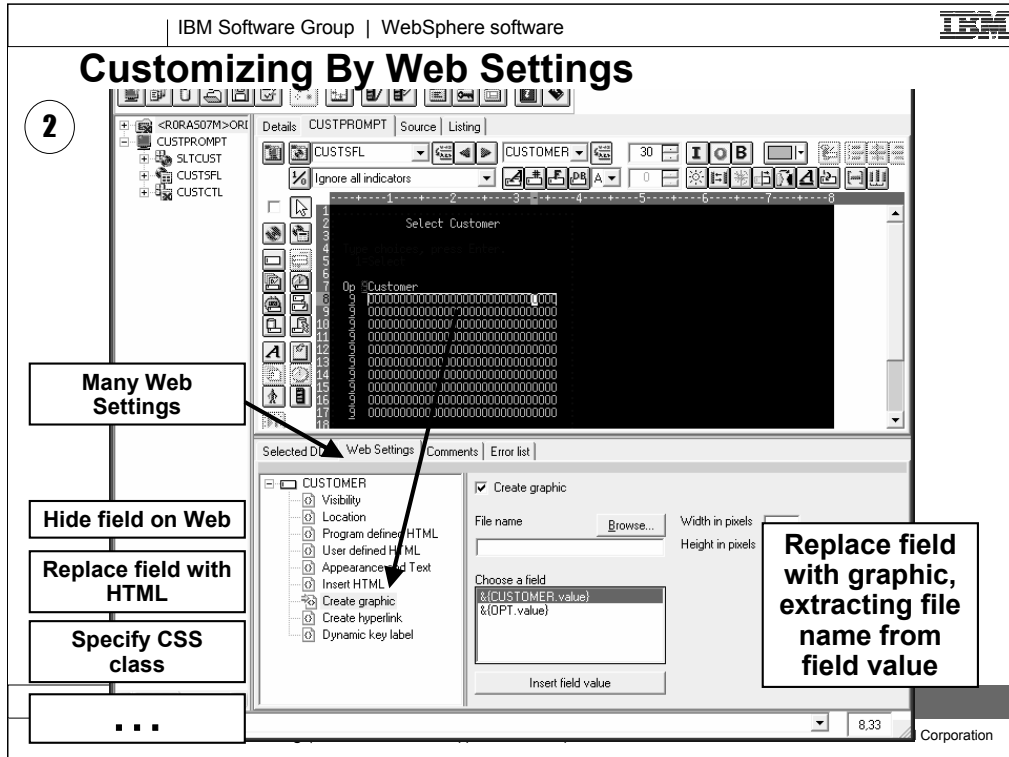
CODE
Designer
= modern
SDA



After a DDS source member has been loaded into CODE Designer, Web Settings can be accessed by clicking the icons in CODE Designer's DDS Tree. The DDS Tree is located on the left-hand side of the CODE Designer window. Web Settings are also accessible by selecting DDS objects from within CODE Designer's **Details** and **SCREEN** tabs. If Web Settings are available for the selected object, the Web Settings tab will be displayed in the bottom pane of CODE Designer.

Customizing By Web Settings

2



Many Web Settings

Hide field on Web

Replace field with HTML

Specify CSS class

...

Replace field with graphic, extracting file name from field value

CODE Designer allows you to design DDS screens graphically. In CODE Designer, each DDS object is represented by an icon or push button. For example, there are push buttons so that you can easily create named fields or text constants for your DDS screen. The Web Settings available for each DDS object vary depending on the object that you are working with. The online help lists the Web Settings available for each DDS object. In the online help refer to the section of this document *Web Setting descriptions* for more details on each setting. When you use Web Settings for an object, special comments are added to your DDS source which later get processed by the WebFacing conversion. Web Setting comments begin with the characters *%%WB.

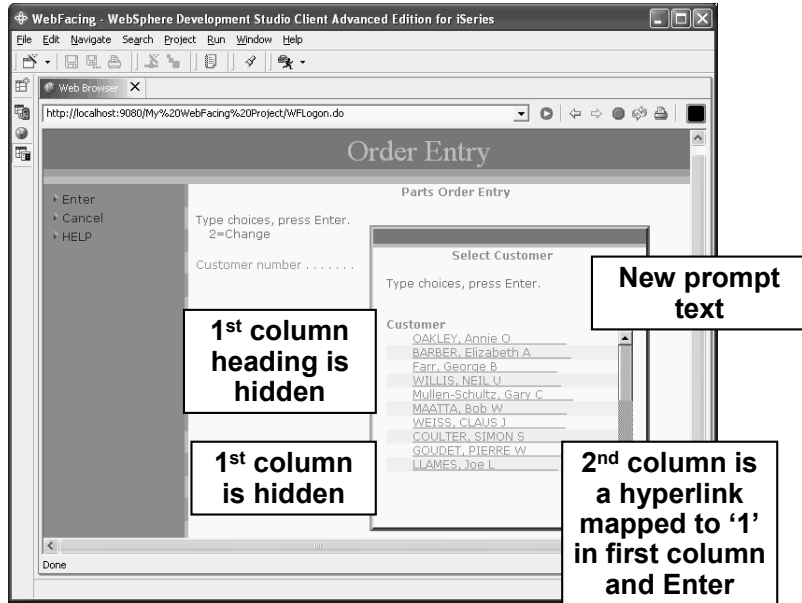
Customizing By Web Settings

2

The screenshot shows the 'Web Settings' tab for 'FIELD1'. The 'Command keys' dropdown is set to 'AUTO'. A callout box highlights the 'Use field value as command key label' checkbox and the 'Command keys' dropdown. A text box on the right states: 'New in 5.1.0: support for named output fields that contain f-key descriptions sent at runtime'. The 'FIELD1' tree view on the left includes 'Visibility', 'Location', 'Program defined HTML', 'User defined HTML', 'Appearance and Text', 'Insert HTML', 'Create graphic', 'Create hyperlink', and 'Dynamic key label'. The main window displays a 'Select Customer' dialog with a list of customer names.

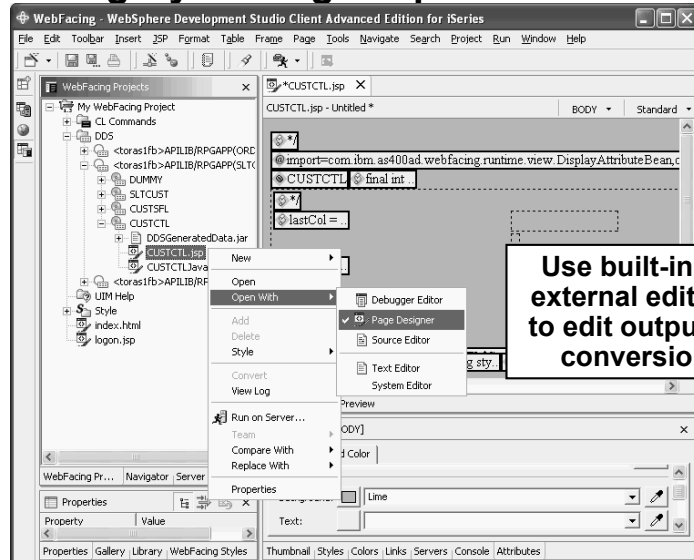
New feature in 5.1, often requested! If you use a named output field to hold f-key descriptions which are determined and sent at runtime, then WebFacing now supports you.

2 After Customizing By Web Settings



Here you can see the results of customizing your WebFaced application style using Web Settings. 1st column heading is hidden, new prompt text appears and the 2nd column heading is a hyperlink.

3 Customizing By Editing Output



If you want to edit the layout and the frame surrounding the application area and command keys, edit the user-defined CSS files in the \chrome directory. To use the IDE's CSS editor, right-click the file that you want to edit and choose **Open with --> CSS Designer**. If you want to edit the layout of the frame, edit the file PageBuilder.jsp. To edit the frame style, right-click **Style --> Edit Style**. PageBuilder.jsp can then be edited using the Page Designer tool supplied by the IDE. The CSS file in the \chrome directory can be edited using the CSS Designer in the IDE. The Web perspective can be useful when editing CSS files. To open the Web perspective, select **Window --> Open Perspective --> Other --> Web**.

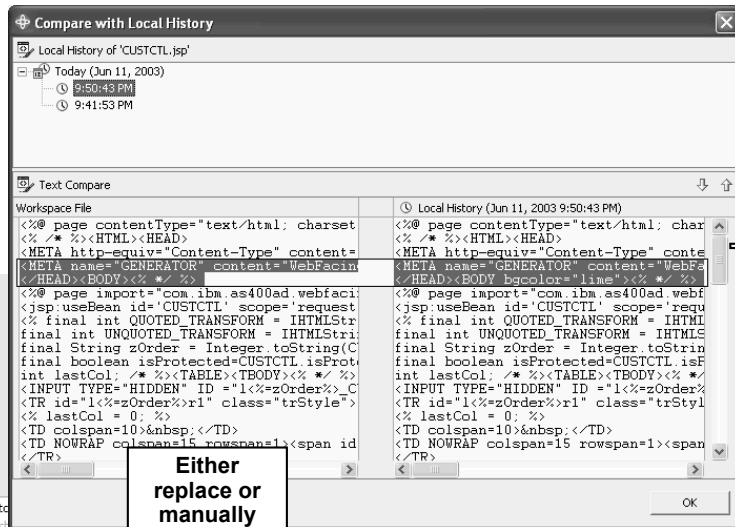
3 Customizing By Editing Output

What happens on subsequent conversion?

The file is replaced with new file!

However, the edited version is available in history

- Open
- Open With
- Add
- Delete
- Style
- Convert
- View Log
- Run on Server...
- Team
- Compare With
- Replace With
- Properties



Compare with Local History

Local History of 'CUSTCTL.jsp'

Today (Jun 11, 2003)

- 9:50:43 AM
- 9:41:53 PM

Text Compare

Workspace File	Local History (Jun 11, 2003 9:50:43 PM)
<%@ page contentType="text/html; charset	<%@ page contentType="text/html; char
<% /* %><HTML><HEAD>	<% /* %><HTML><HEAD>
<META http-equiv="Content-Type" content=	<META http-equiv="Content-Type" conte
<META name="GENERATOR" content="WebFacin	<META name="GENERATOR" content="WebF
</HEAD><BODY bgcolor="lime"><% /* %>	</HEAD><BODY bgcolor="lime"><% /* %>
<%@ page import="com.ibm.as400ad.webfaci	<%@ page import="com.ibm.as400ad.webf
<jsp:useBean id='CUSTCTL' scope='request	<jsp:useBean id='CUSTCTL' scope='requ
<% final int QUOTED_TRANSFORM = IHTMLStr	<% final int QUOTED_TRANSFORM = IHTML
final int UNQUOTED_TRANSFORM = IHTMLStri	final int UNQUOTED_TRANSFORM = IHTMLS
final String zOrder = Integer.toStrin(C	final String zOrder = Integer.toStrin
final boolean isProtected=CUSTCTL.isProt	final boolean isProtected=CUSTCTL.isF
int lastCol; /* %><TABLE><TBODY><% /* %>	int lastCol; /* %><TABLE><TBODY><% /*
<INPUT TYPE="HIDDEN" ID="1<%zOrder%>_C	<INPUT TYPE="HIDDEN" ID="1<%zOrder%
<TR id="1<%zOrder%>r1" class="trStyle">	<TR id="1<%zOrder%>r1" class="trStyl
<% lastCol = 0; %>	<% lastCol = 0; %>
<TD colspan=10> </TD>	<TD colspan=10> </TD>
<TD NOWRAP colspan=15 rowspan=1><span id	<TD NOWRAP colspan=15 rowspan=1><span
</TR>	</TR>

Either replace or manually merge

Changes to the styles made through the property pages or by direct editing apply only to the current project. To save them for use in other projects, right-click **Style** --> **Save as** and give your style a name. Then this named style becomes available for selection the next time you choose a Web style during project creation or the next time you select a style to replace the current project style using right-click **Style** --> **Select Style**.

Table of contents

e-business Primer

What is WebFacing?

WebSphere Development Studio Client

WebFacing Tools

Customizing WebFacing

□ □ □ **What's new in WebFacing 5.0 and 5.1.0**

Summary

Previously we reviewed WebFacing customization features. Now lets review what new enhancements were added for Version 5.0 of the IBM WebFacing Tool.

What's New in WebFacing 5.0

- Standard
- Advanced

In this section we look at the WebFacing Tool packaged in the standard edition of Development Studio Client followed by the IBM WebFacing Tool packaged in the Advanced edition of Development Studio Client.

WebFacing Version 5.0 Enhancements

- WebFaced applications are now non-interactive on new iSeries servers!
 - J2EE 1.2 and 1.3 support
 - Automatic configuration of UTF-8 in WebSphere Application Server 5.0
 - Web page compression when using WebSphere Application Server 5.0
 - Use web.xml versus proprietary config file
 - Support for function key buttons to be placed anywhere
 - Support to replace function key descriptions with buttons
 - Now generate one xml file versus two Java beans, per record
 - WAR file is much smaller, faster to create and deploy. Less memory

Some Version 5.0 WebFacing enhancements include:

- Yes, on the new 8x0 boxes, there is no interactive charge for WebFaced applications
- J2EE 1.2 and 1.3 support
- Automatic configuration of UTF-8 in WebSphere Application Server 5.0
- Web page compression when using WebSphere Application Server 5.0 (performance)
- Use web.xml versus proprietary config file
- Support for function key buttons to be placed anywhere
- Support to replace function key descriptions with buttons
- Now generate one xml file versus two Java beans, per record

WebFacing 5.0 Enhancements (continued)

- User edit code support
- Programmatic invocation
- Web session management for coexistence with other Web apps
- Web browser management for back button and closing
- iSeries load balancing and WebSphere Application Server cluster support
- Reply to DSPPGMMSG, application error
- Many performance, scalability, usability and functional enhancements . . .

Enhancements to the IBM WebFacing Tool in Version 5.0 include:

- User edit code support
- Programmatic invocation
- Web session management for coexistence with other Web applications
- Web browser management for back button and closing
- iSeries load balancing and WebSphere Application Server cluster support
- Reply to DSPPGMMSG, application error
- Many performance, scalability, usability and function enhancements

WebFacing Advanced Version 5.0 Enhancements

- **Support for Struts**

WebFaced applications can be Struts-enabled

- **Support for services**

Extensions through Struts. Roll your own with Struts Builder

- **Support for working with, viewing and printing spool files**

Supplied as an example of a Struts action

- **Support for JSP custom tags**

Ultimately will allow ISVs to affect conversion by supplying their own JSP custom tags

- **iFrame support for WebSphere Portal**

Limitation: one WebFaced portlet application per session

IBM is also making it easier to leverage and extend Web applications created with the IBM WebFacing Tool. IBM has provided an option during the code generation phase to output Struts-compliant Web application infrastructure that can be easily enhanced. The Struts-compliant runtime can now leverage the Struts tools to provide more customization capability. Also included in the IBM WebFacing Tool are:

- Conversion-time extension points to allow developers to replace the default HTML
- Option to use IBM and user-defined custom tags to enhance customization
- Ability to override command-key actions to invoke user-defined functions
- Support for displaying and printing iSeries spool files from inside your Web-enabled applications generated using the IBM WebFacing Tool

The iFrame portlet that makes it easy to add existing Web applications (including the IBM WebFacing Tool applications) to the WebSphere Portal environment

Advanced

Address <http://localhost:9080/vflabcal/RetrieveListOfReports.do>

List of Spooled Files

Views: 1,2,3

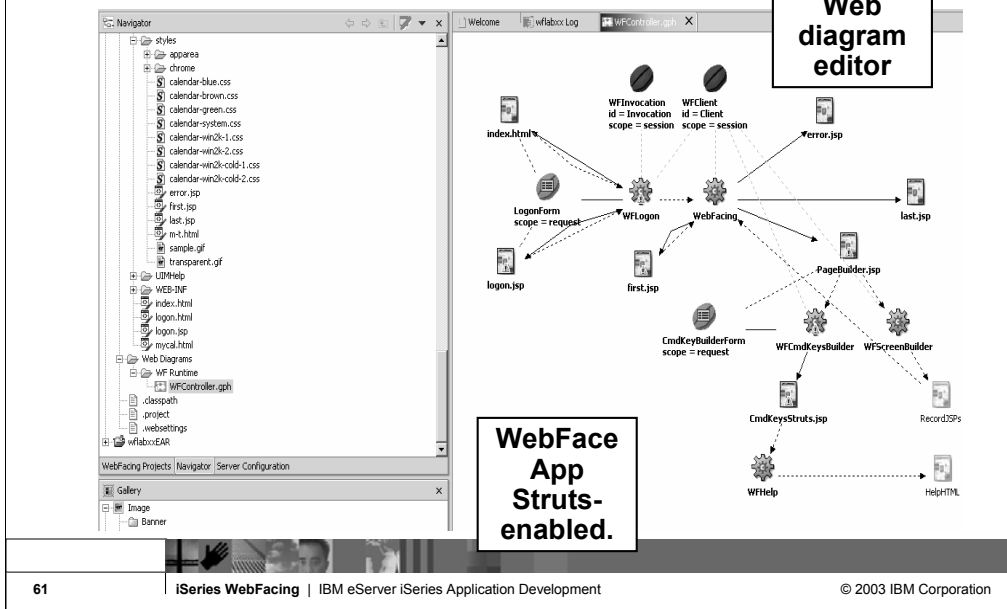
File	User	Device or Queue	User Data	Sta	Total Pages	Cur Page	Copy	Form Type	Pty	File Nbr	Job	Number
QFUDPRTF	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	1	0	1	*STD	5	240	DPRTJOB	256400
PAYROLL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	27	0	1	*STD	5	217	DPRTJOB	256400
QFUDPRTF	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	1	0	1	*STD	5	247	DPRTJOB	256400
GETDATA	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	227	DPRTJOB	256400
QPDZOTALOG	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	1	0	1	*STD	5	1	DPADV000X	913269
GETDSSOL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	226	DPRTJOB	256400
ZSLTCUSTR	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	15	0	1	*STD	5	1	WEISS	913287
GETDSSOL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	225	DPRTJOB	256400
A	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	10	0	1	*STD	5	1	A	913234
PAYROLL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	27	0	1	*STD	5	214	DPRTJOB	256400
PAYROLL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	27	0	1	*STD	5	213	DPRTJOB	256400
QPDSPJOB	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	7	0	1	*STD	5	2	DPADV000X	909682
A	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	10	0	1	*STD	5	240	DPRTJOB	256400
GETDATA	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	229	DPRTJOB	256400
PAYROLL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	23	0	1	*STD	5	221	DPRTJOB	256400
GETDAS	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	230	DPRTJOB	256400
WDCSETUPP	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	238	DPRTJOB	256400
SLTCUSTR	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	15	0	1	*STD	5	1	WEISS	913270
WDCSETUPP	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	3	DPADV000G	995678
GETDSSOL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	224	DPRTJOB	256400
WDCSETUPP	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	2	DPADV000G	995678
GETDASOL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	223	DPRTJOB	256400
WDCSETUPP	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	9	0	1	*STD	5	1	DPADV000G	995678
SLTCUSTR	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	13	0	1	*STD	5	1	WEISS	913286
SLTCUSTR	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTPUT		*READY	13	0	1	*STD	5	1	WEISS	913285

Report service running

You can now work with spooled files in the WebFacing Tool. You define the reporting service through the use of command key override, whereby you override one of the function keys, such as F12, and instead, use the key to define your report service. These command keys override the existing function keys and can connect to a Web application URI or an external URL.

To define a report service command key to work with your spooled files in the WebFacing perspective, right-click your project and select **Project Properties > Runtime > Command key actions**.

Advanced



The screenshot displays the IBM WebSphere IDE interface. On the left, the Navigator pane shows a project structure with folders like 'styles', 'WEB-INF', and 'Web Diagrams'. The main workspace is titled 'Web diagram editor' and contains a 'WebFace App Struts-enabled' diagram. This diagram illustrates the relationships between various WebFace components and Struts-based services. Key components include:

- WFInvocation** (id = Invocation, scope = session)
- WFClient** (id = Client, scope = session)
- LogonForm** (scope = request)
- WFLogon**
- WebFacing**
- PageBuilder.jsp**
- CmdKeyBuilderForm** (scope = request)
- WFCmdKeysBuilder**
- WFScreenBuilder**
- CmdKeysStruts.jsp**
- RecordJSPs**
- WFHelp**
- HelpHTML**

Connections are shown between these components, indicating the flow of data and control. A callout box in the bottom left of the diagram area reads 'WebFace App Struts-enabled.'

When you enable Struts support, you can develop Struts-based WebFacing services with the Struts builder, a visual development environment for building Struts services. Using the builder, you can drag and drop JSPs, action mappings, form beans, and Java beans onto a Web diagram. In the diagram, you make connections between items on the palette, and then activate those items by double-clicking and editing them through a wizard or source code.

What's new in WebFacing for Version 5.1.0 (Oct '03)

- Support for WebSphere Application Server 5.0.2
- Preference to compress/not-compress xml files.
- Better error-handling of userId/password prompt, including support to prompt for expired password.
- New properties affecting conversion / runtime:
 - Show function-key buttons within windowed records
 - Compress web pages before sending to browser
 - Mapping a key to field-exit behavior
- Support for dynamic function key fields
 - Web setting to identify output fields that hold function key descriptions
- Support for new IBM Site Designer
 - Create web sites easily, and contribute WebFaced apps to them
- Less resources used for subfiles, aiding scalability (large # users)
- Support for Struts 1.1 in advanced edition.

In the new 5.1 release, IBM continues the tradition of many enhancements to this strategic tool.

Web Site Designer

The screenshot displays the IBM Web Site Designer interface. On the left, the **Project Navigator** shows a tree view of the project structure, including folders like **DefaultEAR**, **My Web Project**, **Web Site Configuratic**, **Web Deployment: Des**, **JavaSource**, **WebContent**, **META-INF**, **theme**, **WEB-INF**, and files like **messages.js**, **My JSP.jsp**, **newpage2.jsp**, and **newpage4.jsp**. Below the navigator are tabs for **Project N...**, **Gallery**, and **Struts Ex...**.

The main workspace shows a **Layout Page Hierarchy** diagram. At the top is a box labeled **login**. Below it is a box labeled **work with customers**. From **work with customers**, three arrows point to boxes labeled **add customer**, **change customer**, and **display customer**.

At the bottom of the workspace, there are several toolbars and panels: **Attributes** (with the text "No attributes are available."), **Colors** (showing a color picker set to #0000ff), and a **Palette** panel with the message "A palette is not available for this editor".

At the very bottom of the window, the text **My Web Project/.website-config** is visible.

Web Site Designer

The screenshot shows the IBM Web Site Designer interface. The main workspace displays a page design for 'newpage4.jsp'. The design includes a header area with a dark oval, a navigation bar with buttons for 'login', 'work with customers', and 'change customer', and a content area with the text 'Place your page content here' and a list of links: 'login', 'work with customers', 'add customer', 'change customer', and 'display customer'. The interface includes several panels: 'Project Navigator' on the left showing a file tree with 'WEB-INF' and 'horizontal-tab' files; 'Attributes' at the bottom left; 'Design | Source | Preview' tabs at the bottom center; 'Colors' at the bottom right; and a 'Palette' on the far right containing various site parts like 'Horizontal Navigation Bar', 'Vertical Navigation Bar', and 'Site Map'. Three callout boxes with arrows point to specific elements: one points to the content area, another to the navigation bar, and a third to the 'change customer' button.

Content area is only unique part per page

Create each page with common template

Pre-supplied navigation bar and site map parts

Table of contents

e-business Primer

What is WebFacing?

WebSphere Development Studio Client

WebFacing Tools

Customizing WebFacing

What's new in WebFacing 5.0 and 5.1.0

□ □  **Summary**

Summary

e-business Primer

What is WebFacing?

WebSphere Development Studio Client

WebFacing Tools

Customizing WebFacing

What's new in WebFacing 5.0 and 5.1.0

Summary

This presentation reviewed what e-business was all about. Then we looked at WebFacing and what it is. Next we introduced Development Studio Client. The steps to WebFace an application were described in detail. WebFacing customization was described followed by a review of the new WebFacing Tool Version 5.0 features.

New Book

ISBN: 1-931182-09-4

www.mcpressonline.com/ibmpress



This book, the first in the IBM Press Guided Tour series, describes how to get started with IBM's new WebFacing tool and how to make progress with WebFacing efforts.

The systematic approach presented here takes readers through WebFacing a sample green screen application using IBM WebSphere Development Studio Client. It also demonstrates how to use the different features in the tool to enhance the application after the initial conversion. Even debugging strategies are discussed.

To maximize the speed and completeness of your learning experience, the author takes a practical, hands-on approach to explaining WebFacing features and how to use them. This approach, gained from involvement in defining the capabilities of the WebFacing tool allows you to benefit from both the step-by-step nature of the book as well as the most comprehensive treatment offered on WebFacing training.

The book's many exercises will teach you a variety of features in the WebFacing tool and the WebSphere Development Studio Client product. The relationship between these products will also be explained as well as the reason for each of the various exercises.



New Redbook

SG24-6801-00
www.ibm.com/redbooks



This redbook explains how the application conversion works and how you can customize your Web interface. It explains how to work with JavaServer Pages and cascading style sheets. It also offers helpful performance considerations and troubleshooting tips. Plus it looks at how to deploy WebFaced applications to Apache Tomcat.

This redbook is written for iSeries application developers who want an easy way to extend the life of existing host applications. It also applies to those who want to expand the reach of existing host applications to a wider set of clients.

New Redbook

SG24-6961-00

www.ibm.com/redbooks



Java and e-business are key to the future of the iSeries server. Web-enabling your 5250 applications allows you to quickly participate in the e-business arena by using existing applications and programming skills.

This Redbook discusses WebSphere Development Studio Client for iSeries V5.0 of which the WebFacing tool is a part. Development Studio Client includes several powerful tools that target the iSeries application developers:

- WebSphere Studio Site Developer Advanced (in WebSphere Development Studio Client Standard Edition for iSeries V5.0) or WebSphere Studio Application Developer (in WebSphere Development Studio Client Advanced Edition for iSeries V5.0) - the new IDE for developing applications. The unique characteristic of this IDE is the ability to add new features in the form of the plugins. Anyone can develop a new plugin and install it into the tool without creating a "plumbing" infrastructure. WSSD and WSAD include the development environment for creating plugins.
- Development Studio Client includes several iSeries specific features, like the IBM WebFacing Tool and the iSeries Web development tools, that are installed as plugins.
- Cooperative Development Environment (CODE) - a workstation-based tool that supports the development of the applications in many different host languages, including RPG and Java.
- VisualAge RPG - If you are already an experienced RPG IV programmer, you can create graphical user interfaces to RPG programs very quickly in VisualAge RPG.
- Integrated iSeries Debugger.

Disclaimer

- **Acknowledgment:**

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

Claus Weiss, Phil Coulthard, George Farr, Don Yantzi, Satish Gungabeesoon

- **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 PRIOR to making copies of this material for any reason.



Trademarks & Disclaimers

© IBM Corporation 1994-2003. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	IBM (logo)
AS/400e	iSeries
e (logo) business	OS/400
IBM	

Lotus, Freelance Graphics, and Word Pro are registered trademarks of Lotus Development Corporation and/or IBM Corporation. Domino is a trademark of Lotus Development Corporation and/or IBM Corporation.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

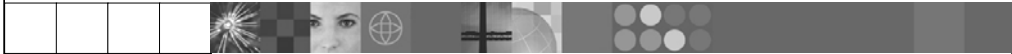
Photographs shown are of engineering prototypes. Changes may be incorporated in production models.



IBM Software Group

iSeries WebFacing

WebSphere. software



2003 | WDS V5R2 and WDSC 5.0, 5.1

© 2003 IBM Corporation