

# The IBM WebFacing Tool A Better User Interface

By Phil Coulthard and George Farr



In “An RPG-to-J2EE Roadmap” (August 2003, article at [www.iSeriesNetwork.com](http://www.iSeriesNetwork.com)), we introduced a simple roadmap for moving from RPG or Cobol to Java 2 Enterprise Edition (J2EE). So far, we’ve covered the Better Tools step in the roadmap (see “Remote System Explorer in WSDc 5.0,” and “Better Tools: iSeries Projects,” in previous TUG magazines. Now, we’ll turn to the next step: the Better User Interface.

This step involves adding a modern, Web-based user interface to an existing 5250 application. The IBM offerings at this step are the IBM WebFacing Tool and Host Access Transformation Server (HATS). Here, we introduce the WebFacing Tool; next month, we’ll discuss the new HATS offering. After that, we will present “Detour to WebSphere Development Studio Client 5.1,” where we take a small detour from the roadmap to talk about the new 5.1 release of WebSphere Development Studio Client (WSDc), which contains most of the tools described in the roadmap.

## A Simple Transition

The WebFacing Tool in WSDc provides a simple way for you to convert display file DDS into Web page source code in the form of JavaServer Pages (JSPs). These user interface source files are then deployed on WebSphere Application Server (WAS). At runtime, the data written from the application is intercepted at the WorkStation Data Manager and inserted into the JSPs, and the resulting Web page is sent to the Web browser. No change is needed to the application business logic, unless you need to code around a keyword not yet supported by the WebFacing Tool, or you want to optimize screen flow for the Web. A system

API (QqfEnvironment) is supplied to query whether you’re currently running in WebFacing or not, should you need this optimization. Other than WAS (Express or higher), you don’t need to purchase anything else in order to run a WebFacing application.

## Converting

Using the WebFacing Tool for the first time is simple. The wizard for creating a new WebFacing project prompts you for the DDS and UIM-help (User Interface Manager) members to convert, the overall style to apply to the generated Web pages, one or more CL commands to invoke the application, and whether to hard-code the user ID and password or prompt for them at runtime. Then, the wizard creates the project.

To convert a DDS file, you merely right-click it and select Convert. The conversion process takes a few minutes for each source member, as the project is populated with the results of the conversion.

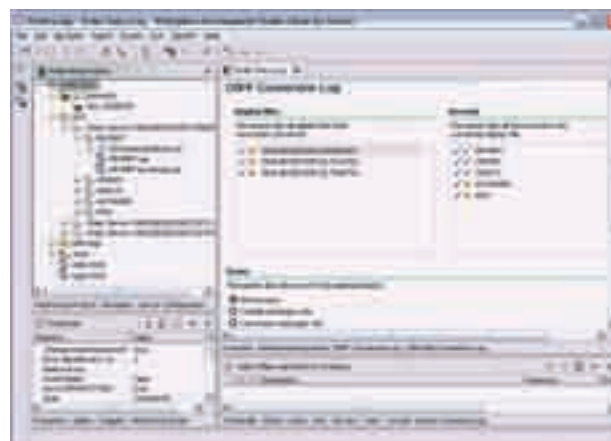
**Figure 1** shows the resulting WebFacing perspective after a conversion. Notice the conversion log file on the right, which includes all the DDS keywords

found and their status for WebFacing support. The first column shows compile status, and the second is conversion status. Generally, you can ignore warnings. Also notice in the left window under DDS the members that were converted and, under each, the record formats in that member and the output files resulting from the conversion of that record format: a Java Archive (JAR) file containing the XML describing the field data, a JSP for the Web page, and another JSP for the JavaScript (so it will be cached by the browser).

## Running

Congratulations; your new Web application is ready to run. Simply right-click it and select Run on Server, which fires up the built-in copy of WAS and runs the converted application with a built-in copy of a Web browser (**Figure 2**).

When you’re running the local WebSphere test environment, the user interface is processed locally in the tools, although the business logic still runs where it has always been run, on the iSeries. Note that when a JSP file is first “touched,” it is compiled into a servlet, so you’ll find that your application runs slowly the first time through. Further, the built-in test environment is designed for testing and debugging, so it is less speedy than your production environment will be. However, you can run your application immediately, and that’s going to make you feel great. You can even cut and



**Figure 1:** A WebFacing project after conversion

paste the URL and send it to a colleague, salesman, or boss, and they can run it on your machine from their machines.

### Refining

If you're happy with the default conversion that WebFacing does, then you're done! Easy, isn't it? It was designed to be. But who is ever happy with any default transformation? Typically, we want to

for Version 5.1, as is the XML Record Metadata page, where you can turn off archiving of the generated XML for better team support. You can set many important properties in this dialog, so explore each page to see what it offers.

**Web settings.** Although the Properties pages are easy to change, they only support global settings such as colors and fonts.

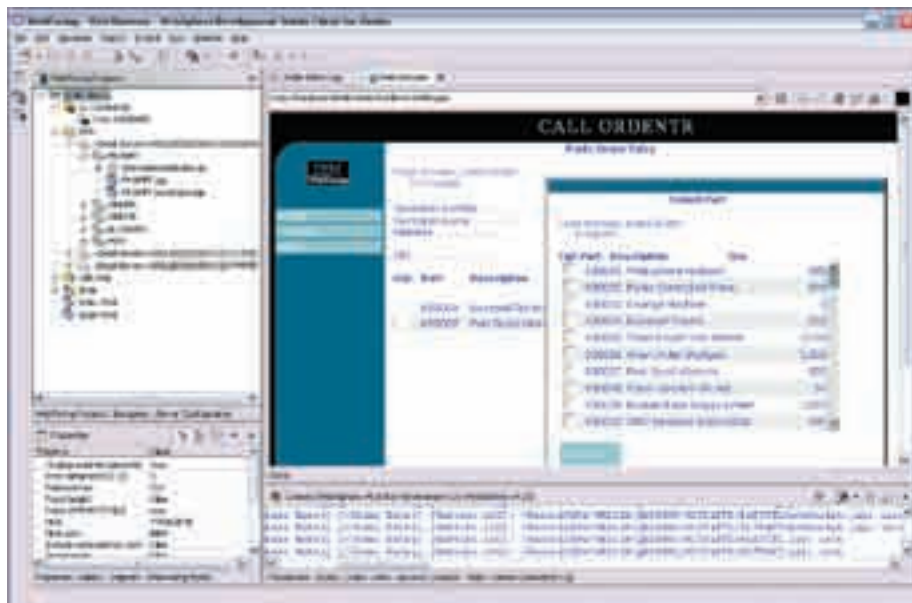


Figure 2: Running in the local test environment

customize the Web user interface and make it look more “Webby” and hence more attractive and competitive. This is where the full power of a development-time conversion becomes most apparent. There are three ways to affect the Web user interface: (1) by using Properties for the WebFacing project, (2) by using Web settings that are set in the DDS source, and (3) directly, by double-clicking on the generated JSP files and editing them. Let's look at these options one at a time.

**Properties.** The easiest way to affect what you see is to right-click on the project name and select Properties. You can change several properties; some of them require a re-conversion (under Conversion), and some require only republishing (under Run Time and Style). Figure 3 shows the Properties dialog with one of the pages open. This page is where you enable the function key buttons to appear within windowed records, as we saw in Figure 2. This is a new function

Often, you want to refine individual pages of your application. For example, you might want to hide some text, or show different text on the Web than on the green screen, or insert an image whose file name is determined at runtime from a named field. You might even want to change a selection subfile so that column 1 is hidden, and column 2 is a hyperlink that maps to typing “1” in column 1 and pressing Enter.

All this is possible with Web settings, and you don't need any HTML or JavaScript knowledge. Because Web settings are DDS comments, you also won't need to recompile your DDS, although you will have to reconvert your WebFacing project.

To work with Web settings, right-click the DDS member in your project and select Open With CODE Designer. This launches the follow-on to SDA, a really cool what-you-see-is-what-you-get designer for screens and reports. For Web settings, you simply open the design window, select the fields you want to affect, and go to the Web Settings tab. Figure 4 shows an example of a new 5.1 Web setting (at the bottom of the screen) that tells the conversion tool that the selected output field contains function key descriptions sent at runtime.

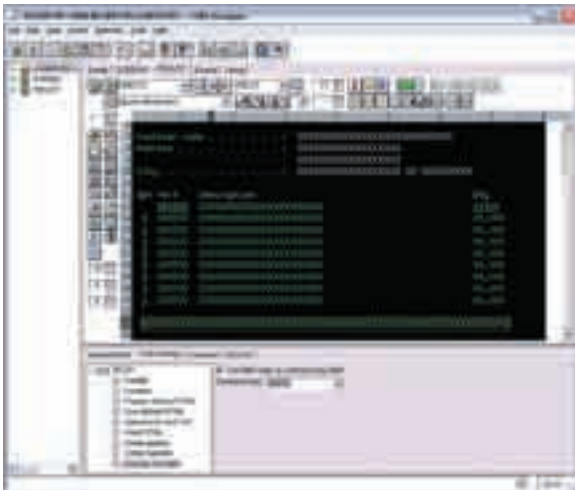
**Direct Editing.** Finally, you have the option of directly editing your generated JSP files. This is as simple as double-clicking on them to open them in the built-in JSP editor, known as the Page Designer. If you find this an attractive option, beware that when you next convert, the generated file will override your work. However, the previous version of the file is still there, and you can use right-click actions to compare and merge the two versions, or even replace the current version with the previous version. This is a nice capability, but it's tedious and error-prone, and IBM recommends you use this approach only if Web settings don't meet your needs.

### Publishing

After you've had a great deal of fun iteratively refining and re-testing your WebFacing application, you'll want to put it into production or at least test it in your production copy of WAS. To do this, you export the project as an enter-



Figure 3: Properties for WebFacing project



**Figure 4: Specifying Web settings**

prise archive (EAR) file to the IFS on your iSeries and then import that EAR file into WAS on the iSeries (or anywhere that WebSphere will run).

Someone on your team must become the designated WebSphere administrator, installing it on your server and configuring it. This is a strategic skill to have

these days, as WAS is becoming more pervasive, and more applications are shipped that need it. The good news is that today, it is much easier to work with the application server than ever before, especially with the Express version of WAS, which has a simplified Web-based console.


Once the application server is installed and configured, the importing of EAR files is straightforward and can be done by a developer or administrator.

**Help**

Web-enabling applications using the WebFacing Tool is rapidly becoming a cottage industry. Help is available from IBM and many Business Partners. IBM ships extensive documentation with the product, and online and offsite courses are available that teach both WAS

Express and the WebFacing Tool. To find courses, go to <http://www.ibm.com/series/websphere> and click Education.

A hands-on lab is also available online. For a quick tour of it, go to <http://www.ibm.com/software/awdtools/wds400> and click Library. IBM also offers WebFacing services, which you can find under the Services link. You can also find the IBM Redbook *The IBM WebFacing Tool: Converting 5250 Applications to Browser-based GUIs* at [www.ibm.com/redbooks](http://www.ibm.com/redbooks), and a book by IBMers Claus Weiss and Emily Bruner, *Understanding the IBM WebFacing Tool: A Guided Tour*, is available at <http://www.mc-store.com/mcpresonline/5216.html>.

Beyond IBM, many Business Partners now also offer their own WebFacing courses and services for helping do the conversions. These can be very useful to overcome missing keyword support, to provide enhanced functionality such as embedded system screens or pop-up calendar windows, or even just to create more appealing styles. You can find these Business Partners by looking at some of their WebFacing reference accounts at <http://www.ibm.com/software/awdtools/wds400> under the link for Success Stories. You can also try typing in “WebFacing” at Google! 

**Phil Coulthard** works at the IBM Toronto lab, where he is the lead architect for application development tools and languages on iSeries.

**George Farr** works at the IBM Toronto lab, where he is the technical development manager for the RPG and VisualAge for RPG languages, as well as the new RPG and Cobol tools in WDS*c*. Phil and George are frequent speakers at many conferences and user groups worldwide, and their books “Java for RPG Programmers, 2nd Edition” and “Java for S/390 and AS/400 COBOL Programmers”, are available from Penton and MC Press.

This article was first published in iSeries News magazine.

# The 5<sup>th</sup> Wave

By Rich Tennant



“THE IMAGE IS GETTING CLEARER NOW... I CAN ALMOST SEE IT... YES! THERE IT IS—THE GLITCH IS IN A FAULTY CABLE AT YOUR OFFICE IN DENVER.”

© The 5th Wave, www.the5thwave.com