



IBM eServer™ iSeries™

iSeries Developer Roadmap

RPG/COBOL Modernization

www.ibm.com/series/roadmap

iSeries Application Development Team, IBM Toronto
www.ibm.com/software/awdtools/wds400

© Copyright IBM Corporation, 2004. All Rights Reserved.
This publication may refer to products that are not currently
available in your country. IBM makes no commitment to make
available any products referred to herein.

Welcome

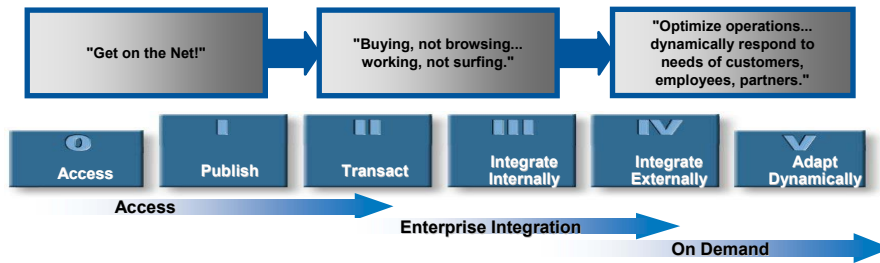
Hi, and welcome to this online course, entitled “iSeries Developer Roadmap to J2EE.” In this course, you will be learning about the most ideal method for a classic, green screen iSeries shop to move into the Web application world... in a staged, non-disruptive manner. This Java 2 Enterprise Edition® (J2EE®) roadmap has been specifically designed to take into consideration the extent to which your shop is probably presently entrenched in a 5250 application model, as well as the large amount of experience of which you surely can boast, in relation to the traditional programming languages that support your green screen applications.

You will learn of a five-step evolution that you can move along at a pace that is appropriate for your organization, but all the while knowing that the goal is to become a Web application-driven enterprise.

So, let’s get started.

Phases of e-business Adoption

From accessible to adaptable



Historical Phases of e-business Adoption

You can see from this diagram that there have historically been several phases of e-business adoption. In fact, you have probably been hearing about many of these phases for years now. Some Web enterprises have trod through every phase shown here, while others have sidestepped one or more phases as they have moved forward. As you study this, you should determine where along this line your organization currently resides. This is important because that situational awareness will mark your starting point to “keep heading on down the road.”

Of course, the first phase is that a business needs to “just get on the ’net” as quickly as possible with some kind of presence. This means that the organization must have access to the Web and must have the minimal skills to publish at least static pages to a corporate Web site. “Static” pages in this reference means that the Web pages display information that does not change. This may be as primitive as providing directions to the physical enterprise, offering some FAQs for customer service purposes, or even publishing annual reports.

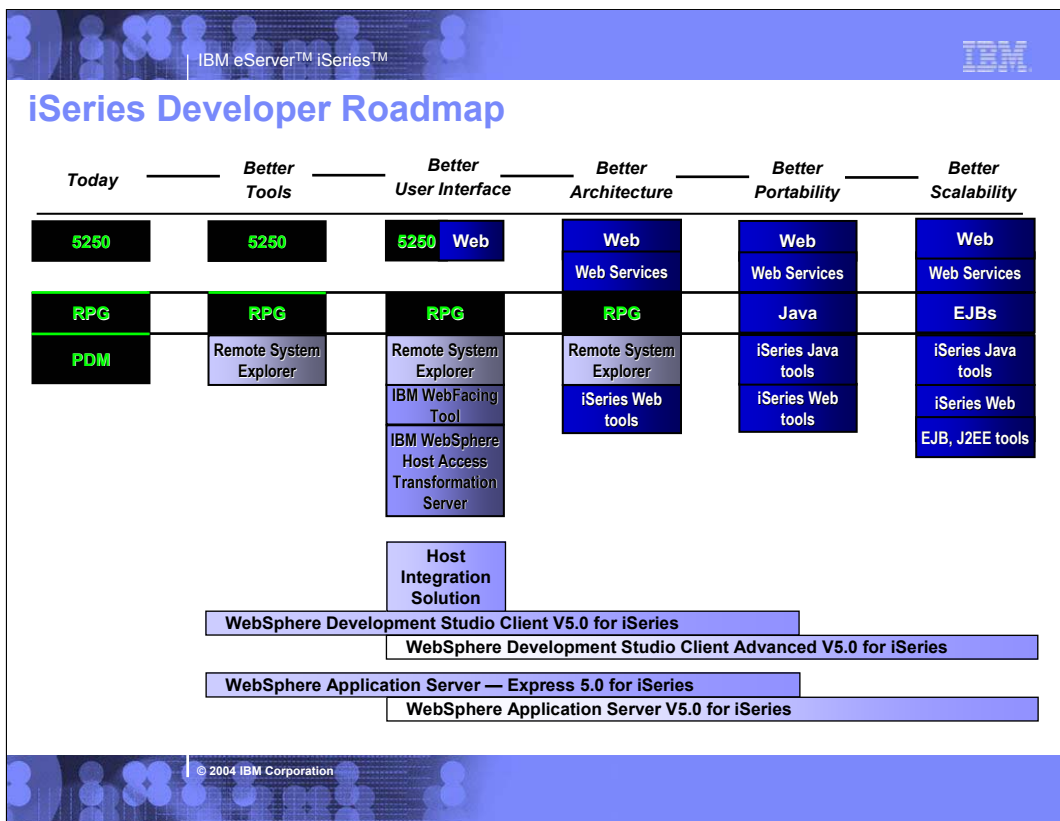
From this point, the company will move into realizing the need to support customers in their buying efforts, not browsing; and to support employees and vendors with their work tasks, not surfing. At this point, it will be important for the IT team to begin concentrating on the concept of on-line transactions—and on internal integration of the Web transaction applications with the line-of-business applications.

Previous Slide Notes cont'd

As those of you who are already at this point will agree, this is not an easy task.

The final phase of getting on the Web involves greater concentration on optimizing operations and delivering dynamic responses, both internally, and externally with vendors and customer interactions. During this phase, the enterprise will have already achieved internal integration, and will now be looking to complete an external integration with its vendors' systems and processes. At the same time, the company will need to hold paramount the imperative of constantly protecting its ability to dynamically adapt as the world around it changes.

These concerns and these sequences of maturing have all been at the forefront of concerns for IBM® as its people move to assist you in these processes in a consultative manner. To that purpose, in this course, you will be learning about the J2EE roadmap. This roadmap is different than, and is much more explicit and effective than, the historical phases of e-business adoption just outlined. Further, this roadmap is the result of a great deal of collaborative effort, both within the ranks of IBM, and from its Business Partners and customers, so you can expect that it is influenced and improved by many perspectives.



iSeries Developer Roadmap to J2EE

Moving from traditional iSeries programming languages (RPG, COBOL, C) to J2EE is a long jump, but does not require a single leap. This chart represents a roadmap—discreet, achievable steps that move developers and applications on the path to an excellently implemented Web future. This roadmap is described in more detail subsequently, but here is a quick overview:

RPG, as shown in the charts, represents any of the traditional programming languages.

Starting from the left, let's look at where most IT shops that use the IBM eServer® iSeries™ platform and Business Partners are, today. Typically, day-to-day development tasks still involve building and maintaining green-screen applications via traditional 5250 tools such as Programming Development Manager (PDM), Source Entry Utility (SEU), and Screen Design Aid (SDA).

Under "Better Tools," the first step to Web application enhancement involves embracing modern tools to do the same development work previously accomplished via PDM, SEU, and SDA. Remote System Explorer (RSE) in WebSphere® Development Studio Client offers powerful new tools for the development of RPG, COBOL, C, C++, CL, SQL, and DDS. The adoption of this single tool significantly increases productivity, and also important, attracts and

Previous Slide Notes cont'd

keeps new programmers. RSE introduces the same Interactive Development Environment (IDE) used further to the right on this chart. This means learning curves incurred now will be useful all the way down the Web-enablement path.

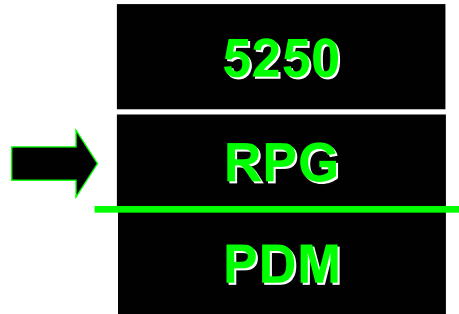
A need end users consider urgent is a better user interface (UI) than the generations-old green screen. For most applications, this is best addressed by moving to a browser-based user interface. The WebSphere Development Studio Client product (that contains RSE) also includes the IBM WebFacing Tool for iSeries, which converts DSPF DDS source into Web page source (that is, JavaServer™ Pages, or JSPs™). The newly deployed Web application then runs the 5250 application with no changes to the legacy code—it merely displays the green-screen output from a Web browser. The converted browser screen results can also be customized. IBM WebSphere Host Access Transformation Server (HATS) can instantly transform, at runtime, any 5250 or 3270 sessions into Web pages. The resulting Web pages can also be customized in HATS. Since the underlying applications do not change, they are maintained and enhanced as before, perhaps using RSE. “Better Architecture” is a significant step where—from scratch or from cut-and-paste—you create a Web application, an application enhancement, or even a new Web service. You will author non-interactive business logic (in RPG, COBOL, or another language) by using RSE. Then, using the iSeries Web tools, you may put a Web UI or Web Service front end onto the business logic. You can leverage advanced DB2® UDB for iSeries features such as triggers, stored procedures, and move to SQL for data access. Separating the business logic from the user interface is a very important experience, and is fundamental to a superior architecture that allows for logic reuse.

“Better Portability” involves a move from creating business logic in traditional languages to writing it in Java™. You use simple, standard Java—referred to as Java 2 Standard Edition (J2SE™)—that accesses data in the familiar SQL ways. SQL-based database access is accomplished via JDBC™. iSeries extensions to the Eclipse Java tools, found in WebSphere Development Studio Client, make it easy to deploy Java code to an iSeries server—and to run, debug, and profile it there.

Finally, on the far right, “Better Scalability.” For companies requiring highly scalable Web applications or the full OO and functional power of J2EE, the J2SE Java code is replaced with Enterprise JavaBeans™ (EJBs™) and Message Driven Beans. The tools for these are in the Advanced Version of WebSphere Development Studio Client.

Today

- Most iSeries developers still:
 - ▶ Create or enhance 5250 applications
 - ▶ Use traditional, host-based development environment (PDM, SEU, SDA, RLU, DFU, & system debugger)



Today

As just briefly mentioned, an enormous number of iSeries development shops are still maintaining, and even developing, literally trillions of lines of 5250 green screen application code. And, they are still using development tools that are more than 20 years old (PDM, SEU, SDA, RLU, DFU, and the system debugger) on a routine basis—to build or enhance green screen applications.

As you move forward through this course, you will not hear IBM being insensitive to your need to maintain tried-and-true code. Nor will we posture that you must leave your many years of coding experience behind. IBM recognizes the value of these mature assets, both tangible and intangible.

What you will learn, though, is that it is not only imperative (which it is) that you embrace newer ways of embellishing your code—it is also the most efficient (and frustration-eliminating) direction in which you can take your iSeries developers—while benefiting from their existing skills and utilizing your well-proven business logic... over and over again.

Your IT team will be more effective, your code will improve, your end users will benefit, and your executive management team will be ultimately, utterly, unbelievably pleased. What more could you ask for... we want to help you be a hero!

The image displays two screenshots of the IBM iSeries green screen environment. The top screenshot shows the PDM (Personal Data Manager) interface, which is used for file management. It displays a file list with columns for 'Opt', 'Member', 'Type', and 'Text'. A green arrow labeled 'PDM' points to this window. The bottom screenshot shows the SEU (Source Entry Utility) interface, used for editing source code. It displays a source listing with columns for 'Line', 'Number', and 'Description'. A green arrow labeled 'SEU' points to this window. Both windows are titled 'Session A - [24 x 80]' and show a connection to a remote server/host.

Today...

Here, you can briefly look at a green screen display for PDM and for SEU. We are sure you are already familiar with these development methodologies. But keep your imagination open and receptive to the use of a “Better tool”...

Step 1. Better Tools

- **Move iSeries application development to Eclipse Integrated Development Environment (IDE)**

- **Remote System Explorer (RSE)**

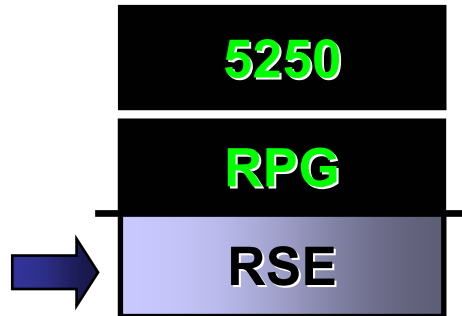
- ▶ Provides iSeries server access from Eclipse IDE (included in WebSphere Development Studio Client for iSeries)
- ▶ 21st century follow-on to PDM, SEU, SDA, RLU, & system debugger

- **Value Proposition**

- ▶ Improves productivity over host 3GL tools
- ▶ Provides easy access to advanced application development tools from IBM & vendors

- **Education Enablement**

- ▶ WebSphere Development Studio Client packaging & introduction
- ▶ Eclipse education (how to use)
 - Tool solution developers may be interested in how to extend Remote System Explorer
- ▶ Remote System Explorer education



Step 1. Better Tools

The first step in the J2EE roadmap does not involve any change to the applications in use within your organization. Rather, it allows you to simply replace your traditional development tools with more exciting and modern tools to support the same code base.

RSE is the follow-on to PDM, SEU, SDA, RLU, and the system debugger. It offers highly productive follow-on's to these “primitive by today's standards” tools. And these new RSE tools are all highly integrated with each other and with the Integrated Development Environment of Eclipse. RSE is also the 21st century Product Data Management (PDM) tool in the sense that it is the point of integration for all iSeries tools vendors, many of whom have already released plug-ins to complement the functionality provided by IBM with additional capabilities.

RSE comes with WebSphere Development Studio Client. By learning and using RSE, PDM programmers will enjoy greater productivity than with host-based 3GL tools. Programmers gain transferable skills that later apply to other Eclipse-based tools in WebSphere Development Studio Client. Learning RSE also opens opportunities to access the next generation of third-party tools that are built on top of Eclipse. Further, RSE works not only with OS/400® files, commands, and jobs, but also with IFS files and Qshell commands, and with Linux files and commands that reside in their own logical partition (LPAR). That is, from a Microsoft®

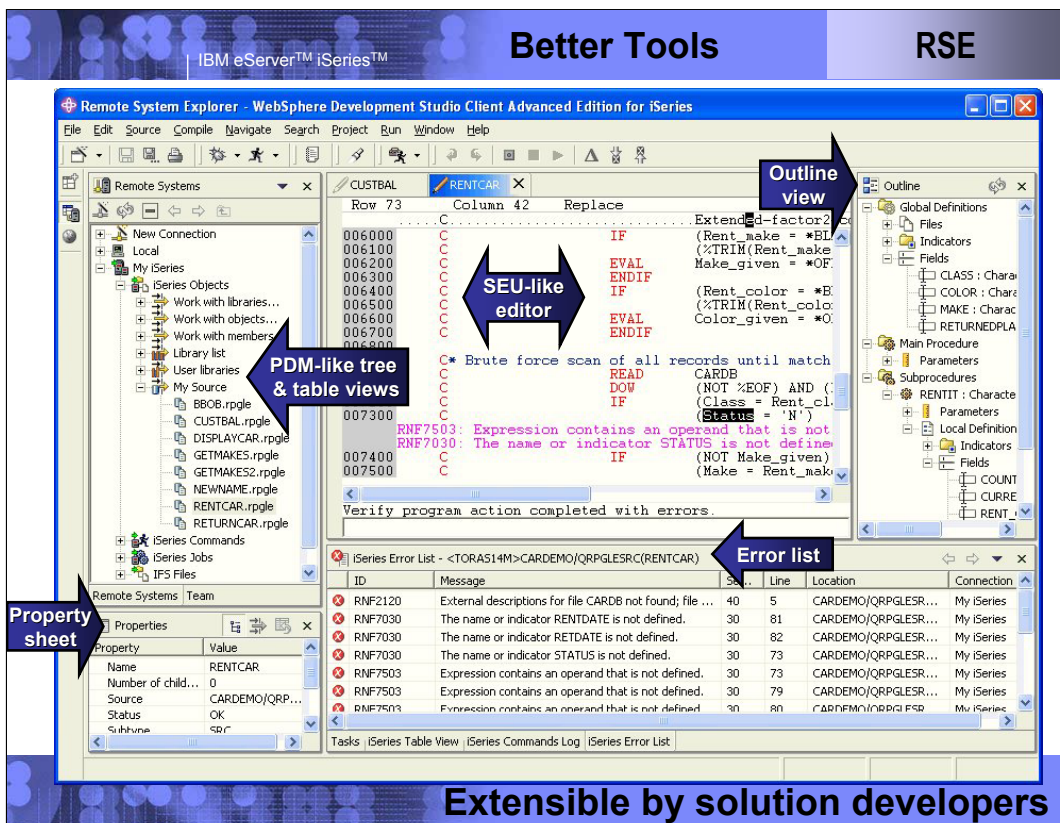
Previous Slide Notes cont'd

Windows® workstation, you can remotely access and edit files and run commands. RSE even works with the files and commands in remote UNIX®, Windows, or Linux servers, as well as with local Windows files and commands. Ultimately, as Java and Web services technologies are further adopted, this consistent support across file systems and command shells will be very important.

What are the “Delta” skills you will need to take this first step?

- WebSphere Development Studio Client
- Remote System Explorer
- Eclipse

[NOTE: We will list the skills needed for the tasks related to each step. “Topics” only are being listed. In the future, an IBM Web site will provide an extensive list of courses (Web-based, classroom, etc.) and manuals (Redbooks, redpieces, and white papers).



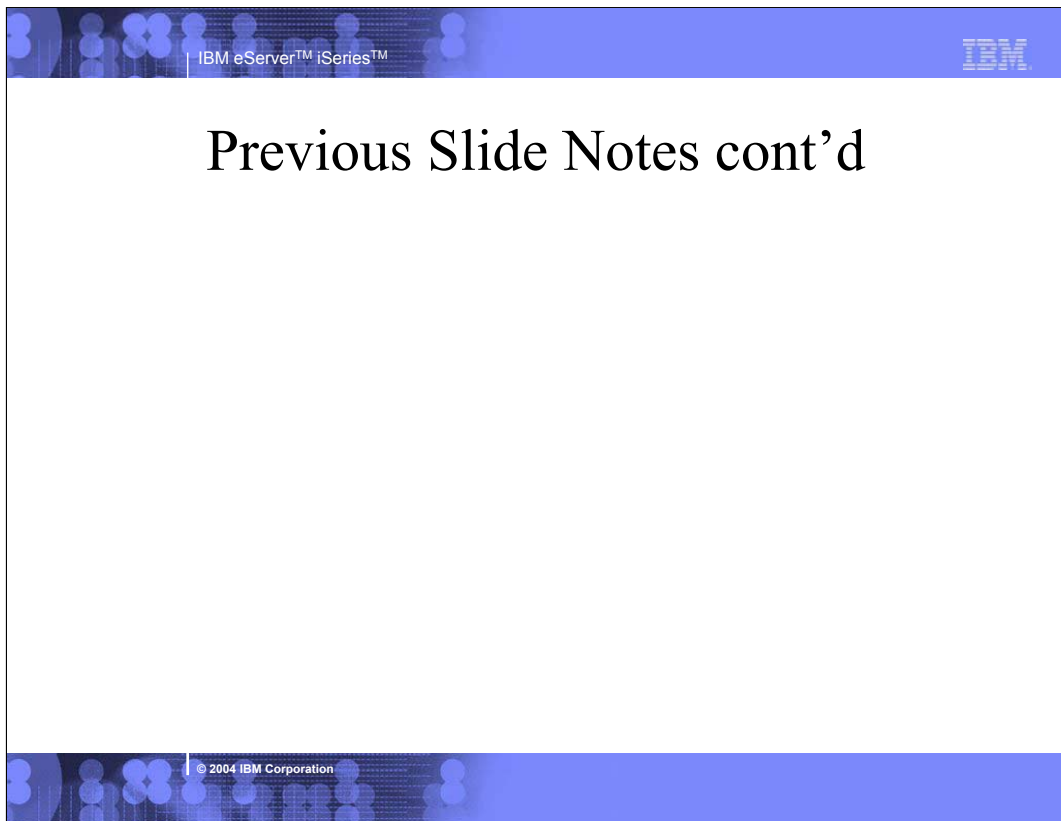
Extensible by solution developers

Better Tools – RSE

Examine this screen capture of Remote System Explorer. The collection of views and editors in RSE are called the Remote System Explorer “perspective.” Developers can open multiple perspectives and flip between them via icons in the bar on the left.

The primary view in the RSE perspective is the Remote Systems perspective, where you create connections to remote iSeries, UNIX, Windows, or Linux servers. This screen shows a connection to an iSeries server whose OS/400 objects are expanded so you can work with them, as well as with members... similar to PDM. Look more carefully at the “My iSeries” tree—you can also work with commands, jobs, and IFS files. As you select objects in the Remote Systems view, the property sheet (lower left) shows information about the selected object (which here is RENTCAR). Some of the object’s property sheet information is directly editable—right there, for your convenience.

Many right-click actions are provided for all object types and members—including source members. The option to open the member in the RSE editor is shown on this screen capture. Beyond what IBM supplies, user-defined actions are possible. The editor is rich in function, far exceeding SEU while retaining features such as entering “D” in the prefix area to delete a line. The editor supports syntax-checking and cursor-sensitive F1 language help. It also has a built-in program verifier for RPG, COBOL, and DDS which does a full error



check, reporting the results in the iSeries Error List (shown in the bottom right). Double-clicking on an error in the list automatically positions the cursor in the editor at the offending line, and optionally inserts the error lines for context, also as shown here (error numbers RNF7503 and RNF7030). The same error feedback is used when the member is remotely compiled.

In the top right is the outline view, which offers an at-a-glance hierarchical picture of the source member currently active in the editor. This is handy for program understanding and navigational purposes.

The editor supports content-assist for RPG, which allows you to press “Ctrl+Spacebar” to acquire a list of available options for where the cursor is currently positioned. For example, in the opcode area, you get a list of valid opcodes, plus “help”—simply select one of the opcodes to insert it at the cursor position.

Tabs are at the bottom of the screen for some of the many other views, such as the iSeries Table View. It is an alternative to the tree view—offering a PDM-like, sortable table view. As with PDM, this is where the command line exists for entering and running OS/400 commands. The “iSeries Commands Log” shows all explicitly or implicitly run commands for this session.

Like all Eclipse perspectives, users can rearrange the views to fit their preferences.

Step 2. Better User Interface

•Create Browser-based UI for 5250 Applications

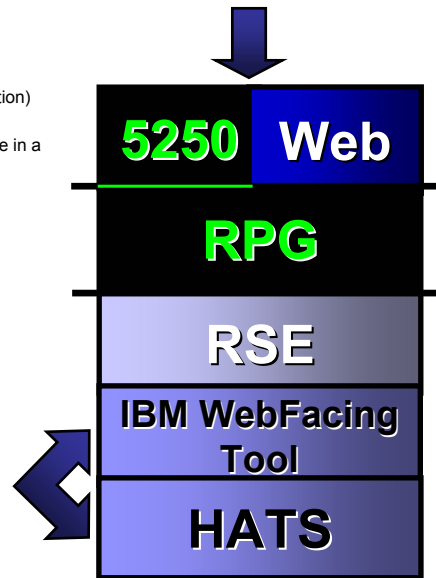
- ▶ Nearly instant conversion to Web UIs
 - At development time for WebFacing (application-by-application)
 - At runtime for HATS (session-based)
 - For IBM WebFacing Tool & HATS, refinements can be made in a repeatable manner
- ▶ No impact to underlying application logic

•Value Proposition

- ▶ Web User Interfaces offer:
 - Ease-of-use characteristics & reduced learning curve
 - Easy access to applications for remote employees, vendors, & partners
 - Expanded market for solution developer solutions
 - Elimination of interactive cycles when using WebFacing & new iSeries servers—reduces total cost of solution

•Education Enablement

- ▶ IBM WebFacing Tool &/or Host Access Transformation Server (HATS)
 - Plus IBM WebFacing Tool service provider offerings
- ▶ WebSphere Application Server — Express for iSeries
- ▶ Introduction to Web application artifacts, architecture, & terms



Step 2. Better User Interface

Other tools create a better user interface for 5250 applications. Two IBM options are IBM WebFacing Tool for iSeries and Host Access Transformation Server (HATS). Both produce a Web user interface from a 5250 UI—with no impact to underlying application logic. They produce UIs that run on WebSphere Application Server - Express (or higher releases)—on any operating system that runs WebSphere Application Server.

The IBM WebFacing Tool for iSeries converts DSPF DDS source, at development time, into a Web application that uses JSPs. The conversion is refined by the CODE Designer tool to add Web settings (via special comments) into the DDS source, which affects the result of the conversion. For example, fields can be hidden or replaced with HTML tags, where the tag contents are derived from the field contents. For example, you can send an IFS image file name to a hidden 5250 field which is converted to an HTML image tag to show the image in the Web page. The CODE Designer tool is the follow-on to SDA, offering a 5250 WYSIWYG view of the application UI.

The IBM WebFacing Tool for iSeries is part of, and leverages the other tools in the WebSphere Development Studio Client (including RSE) for the RPG and DDS logic. Applications created with this tool, do not produce a 5250 datastream. Rather, the OS/400 runtime intercepts the data written by the application to the record formats, sending it to the WebFacing Tool servlet which inserts it directly

Previous Slide Notes cont'd

into the generated JSPs. These applications do not use interactive cycles when run on the new iSeries 8x0 servers announced in January 2003. And, an important “cost” note, there is no runtime fee for these applications.

HATS, the other IBM tool for creating better UIs, is part of Host Integration Solution for iSeries. It converts 5250 or 3270 datastreams, at runtime, to a native Web user interface that also runs in WebSphere. Because it is a runtime conversion, it is truly plug-and-go, instantly transforming screens to run in the Web. HATS developers can easily refine, in a repeatable manner, the conversion results to improve the Web UI. HATS is separately purchased and includes the Eclipse-based development environment.

Often, HATS is used to convert system-based and application-driven screens, into Web UIs. Then, the WebFacing Tool is used on particular applications that, for competitive or usage reasons, need the refinement that comes from a UI source conversion. When source is not available, or the 5250 application changes constantly and needs to be redeployed, HATS is used. Then, its tools are leveraged to fine-tune the result in a repeatable manner.

One last point regarding HATS and the IBM WebFacing Tool for iSeries—both provide portlets for new browser-based UIs to be included in an enterprise portal.

What are the “Delta” skills you will need to take this second step?

- IBM WebFacing Tool for iSeries
- HATS
- Introduction to Web applications
- WebSphere Application Server administration

Other skills that will be advantageous:

- HTML
- JavaScript™

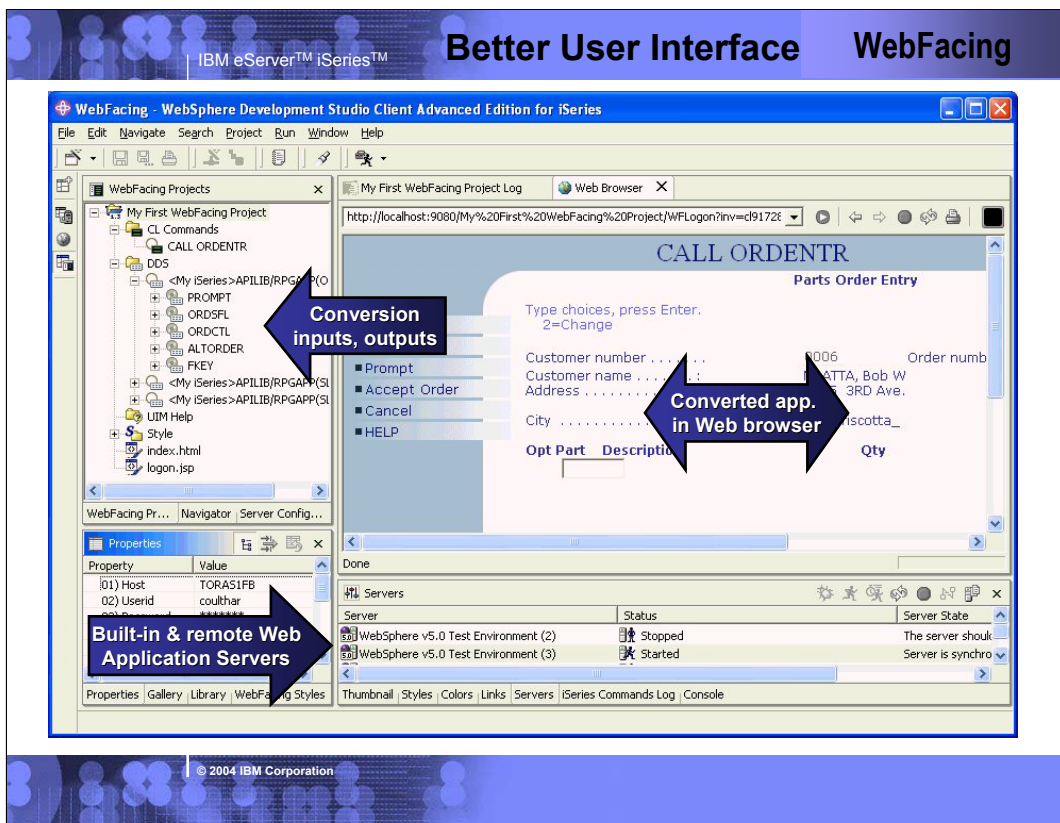
WebFacing

- Given DSPF DDS input, **generates a Web application** that is still driven by the unchanged iSeries business logic.
 - DDS and UIM-Help become JavaServer Pages (JSPs), JavaBeans and a controller Servlet. All generated for you.
 - JSP == Display File for the Web
- Does require the source
- No runtime fee. No interactive cycles on new 8x0 models
- WebSphere Development Studio Client Advanced-only capability
 - Option to use a **struts**-based framework for generated code
 - Option to generate **portal**-tolerant application through iPortal
 - Option to use **custom tags** in generated JSPs for easier customization

There are many options available now for transforming green-screens to Web pages, but WebFacing is unique among them. WebFacing does this conversion at development, taking as input the display file DDS source, and generating JavaServer Pages for the output (for example, Web pages).

The JSPs generated by WebFacing are then deployed to a Web application server, such as WebSphere Application Server, and the application runs as is, but in a Web browser. To enable this to work, with no changes to the underlying application, there is a runtime intercept in OS/400, which detects when an application is running in WebFacing mode. When this happens, the screen data from the application is passed directly to the WebFacing runtime, instead of being used to generate a 5250 datastream. So WebFacing absolutely is not a screen scraper. Indeed, it can't be because there is no 5250 datastream to scrape when running in WebFacing mode.

WebFacing is the result of business partners pleading for a conversion option that allows infinite finessing of the generated output, and is cost effective. WebFacing enables this because its output is editable at development time, and the runtime is a free part of OS/400 in V4R5 or higher.



Better User Interface – IBM WebFacing Tool for iSeries

This is a screen capture of the IBM WebFacing Tool development environment provided by WebSphere Development Studio Client. The entire collection of views and editors is called the WebFacing “perspective.” Developers can open multiple perspectives and flip between them using the icons in the bar on the left. Is this beginning to look familiar? It should because all the development tools are built to have the same look and feel.

Developers use a wizard to create a project. They specify the DDS and the User Interface Member (UIM) Help Panel members to be converted. The resulting project allows editing access to the original DDS and to the generated files. It is easy to test the converted output in the IBM WebFacing Tool for iSeries by right-clicking the project and selecting "Run on Server." This executes the built-in, preconfigured copy of WebSphere Application Server (V4.0, V5.0, or Express V5.0).

The view at the bottom right shows the built-in copies of WebSphere and allows developers to easily start, stop, and publish to them. The same view can be used to manage remote instances of WebSphere Application Server. When running an application within the tool, the application’s UI is shown in the built-in browser, as is seen in the large, right window shown on this screen capture.

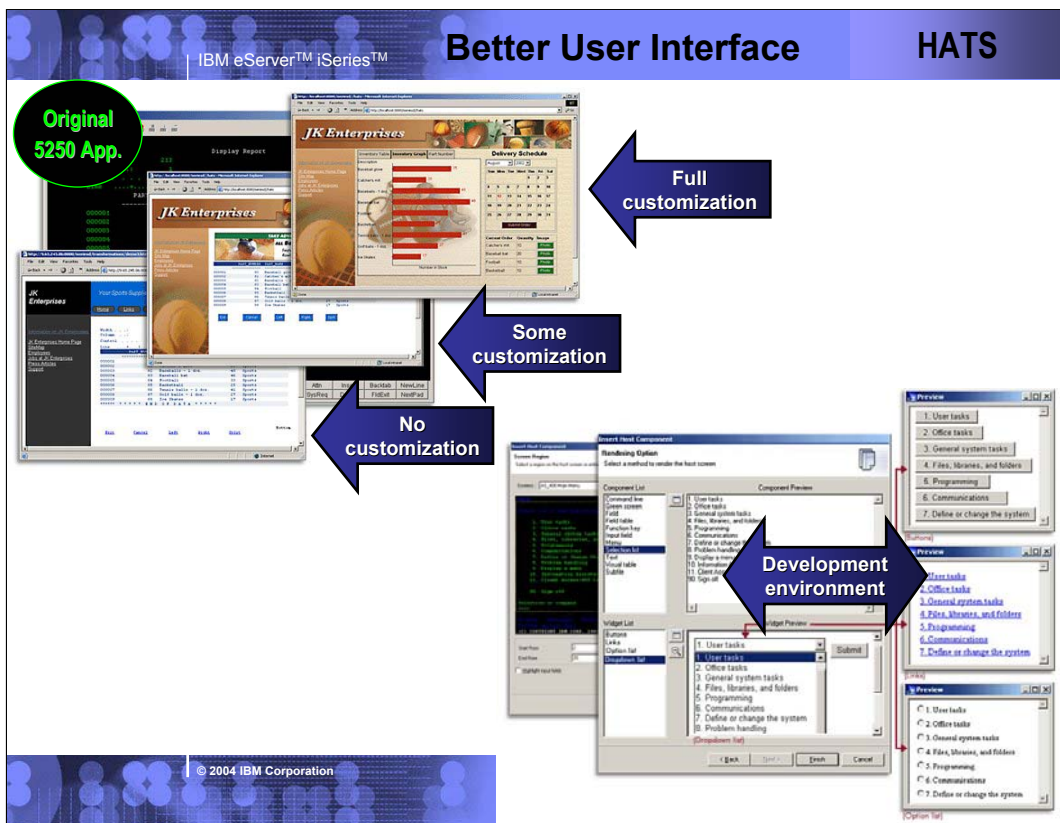
Host Access Transformation Server

- Generates a Web Application that converts 5250 datastream into a Web user interface with JavaServer Pages
- While default transformation is quite good, there is also extensive customization-capability
 - Involves tools to identify the screen to be customized, and the type of transformation to apply to selected screen area
- **Does not require the source**
- Does require a runtime license per-user
 - Unless you use default transformation that comes with Client Access for the Web

There are many options available now for transforming green-screens to Web pages, but WebFacing is unique among them. WebFacing does this conversion at development, taking as input the display file DDS source, and generating JavaServer Pages for the output (for example, Web pages).

The JSPs generated by WebFacing are then deployed to a Web application server, such as WebSphere Application Server, and the application runs as is, but in a Web browser. To enable this to work, with no changes to the underlying application, there is a runtime intercept in OS/400, which detects when an application is running in WebFacing mode. When this happens, the screen data from the application is passed directly to the WebFacing runtime, instead of being used to generate a 5250 datastream. So WebFacing absolutely is not a screen scraper. Indeed, it can't be because there is no 5250 datastream to scrape when running in WebFacing mode.

WebFacing is the result of business partners pleading for a conversion option that allows infinite finessing of the generated output, and is cost effective. WebFacing enables this because its output is editable at development time, and the runtime is a free part of OS/400 in V4R5 or higher.



Better User Interface – HATS

The set of four progressive screen captures (on left) illustrate the result of using HATS:

- The original green screen
- The same 5250 screen as converted to a Web page by HATS—no customization
- The same UI with some customization
- The same user interface with significant customization

You can see by looking at the diversity among these screens, a great deal of built-in customization capability is provided in HATS, including the ability to turn subfiles into graphs and to insert calendar date-pickers.

The three screen captures on the right show the Eclipse-based development environment for HATS—used for the customization of the generated UIs. Alterations require no HTML skills and are accomplished via easy-to-use wizards and dialogs. As with the IBM WebFacing Tool for iSeries, HATS leverages the built-in test environment for WebSphere to simplify the process of seeing the results.

IBM eServer™ iSeries™

Step 3. Better Architecture

- **Create new iSeries e-business solutions**
 - ▶ Using Web & Web services tools in WebSphere Development Studio Client
 - RPG / COBOL / SQL for business logic
 - iSeries Web Tools to generate other
- **Value Proposition**
 - ▶ Properly architected new Web applications offer:
 - Easier maintenance / reuse of business logic
 - Respond to increasing demands
 - Quicker support for multiple interfaces: Web, phone, ...
 - Faster support for B2B via Web services
 - ▶ iSeries Web Tools offer:
 - Focus on business logic, not J2EE plumbing
 - Reuse SDA skills in Web page development
- **Education Enablement**
 - ▶ Introduction to e-business
 - Web application artifacts, architecture, terms
 - Introduction to Struts framework for Web applications
 - Introduction to Web services
 - ▶ Tools
 - iSeries Web, Struts, Web services (including iSeries ProgramCall wizard)
 - ▶ WebSphere Application Server education

© 2004 IBM Corporation

Step 3. Better Architecture

This is the first “roadmap” step that involves developing new business logic. It is a gentle process because you still tap into your abundance of traditional programming knowledge. You may leverage advanced database features or move business rules into DB2 UDB for a better architecture by using constraints and triggers. You will write this new business logic in your familiar RPG or COBOL language. You may leverage SQL, as well. You may extract code that is called by 5250 but which now also is used by a new Web application. You can turn RPG, COBOL, and other language programs into stored procedures to be leveraged by Web applications. Or, you can code RPG that never uses a 5250 interface; instead it is driven by a new Web user interface. Or, it might be exposed as a Web service.

This new business logic is written with RSE. Then, the iSeries Web tools generate the Web user interface and the controller logic, or the Web service. Both of these iSeries Web Tool functions are provided with the WebSphere Development Studio Client. In fact, the iSeries Web tools include a wizard that, given the information needed to call an existing program or procedure, will generate the needed Java logic to make that call, after which, the Web Services wizard is used to turn that Java logic into a Web service.

[NOTE: A Web service allows functionality (in the form of a remote function call) to be exposed to others who can execute it from another server. without knowing

Previous Slide Notes cont'd

or caring about the language in which the service is written or the operating system on which it runs. Web services can be used within a company to get complete reuse of key functionality, or it can be made available between companies for B2B communications.]

The iSeries Web tools also contain a Web Interaction wizard to generate input and output Web pages for the input and output parameters of a program or procedure. It also generates the Web application “glue” needed for binding. The Web pages are functional and ready to be visually improved by Web designers, if preferred.

To help iSeries programmers move from SDA to Page Designer for building Web pages, the iSeries Web tools also come with a number of Web controls, known as visual custom tags that leverage SDA skills. For example, the label and entry field controls support edit code and edit words, while entry fields support validity checking and even database referencing. The necessary HTML and JavaScript is generated for you.

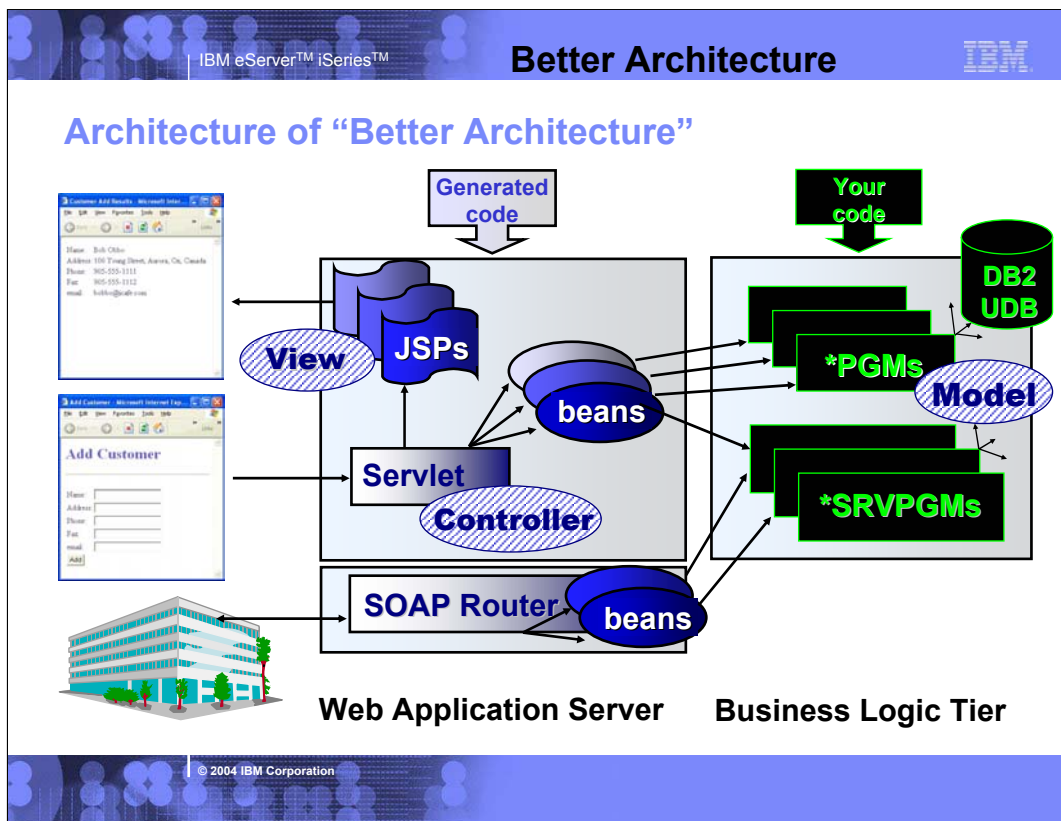
WebSphere Development Studio Client contains database development tools to support DB2 UDB for iSeries—used for creating and managing files in the Web application.

What are the “Delta” skills you will need to take this third step?

- RPG IV or ILE COBOL
- ILE
- Web Services & Web Services Tools in WebSphere Development Studio Client

Skills that will help:

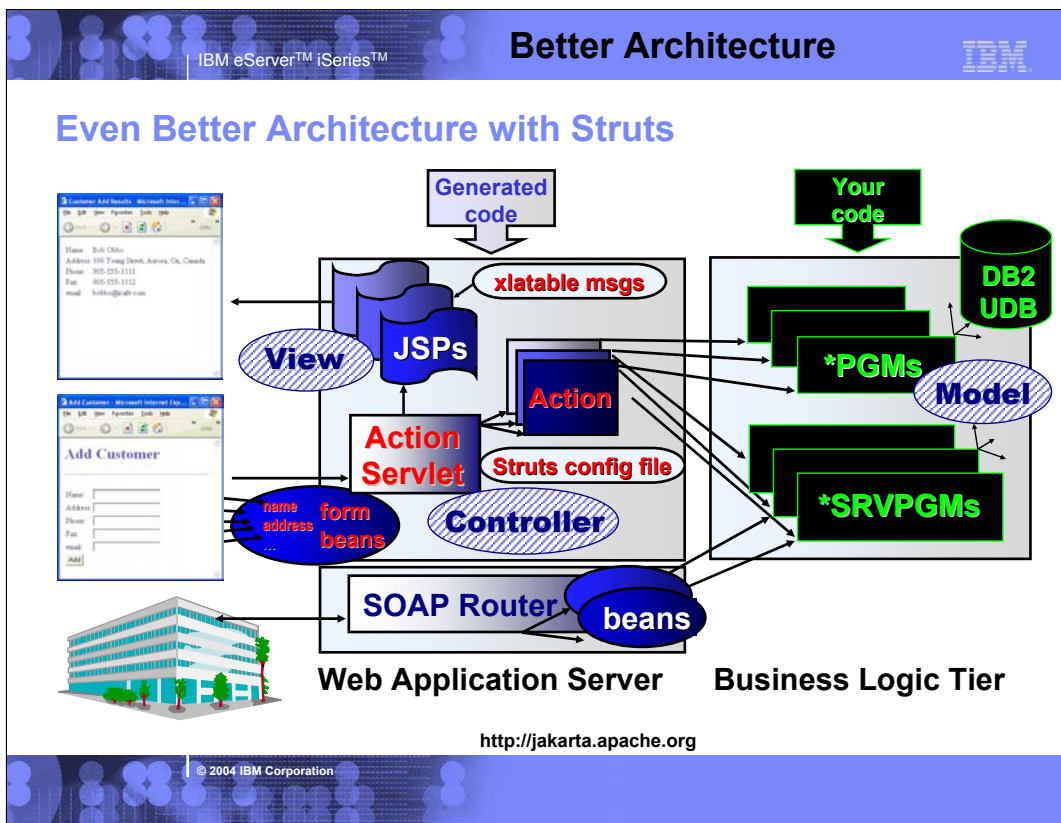
- Separating business logic from UI logic
- Writing modular logic using advanced DB2 UDB and SQL techniques
- Struts



Architecture of a “Better Architecture”

Here, you can see a rather detailed architectural picture of a Web application. The second tier, known as the Web application tier, consists of a single servlet that acts as a controller deciding which business logic Java bean (not an EJB!) is to be called for a given input request. The job of the business logic Java bean is to invoke the actual business logic on the third tier, which from your perspective at this point, is an RPG or COBOL program or procedure. The user inputs are passed to the business logic and the results are returned. The controller servlet then decides what Web page (that is, what JSP) to show next and passes control to it, and at the same time, passing it the business logic Java bean that now contains the results. The results are inserted into the JSP and the resulting page is shown to the user.

For Web services, the caller invokes a generated client, which calls the SOAP router to find the appropriate Java bean to be called. Again, this Java bean is a wrapper of the business logic in the procedure or program. *[SOAP (Simple Object Access Protocol) is an XML-syntax-based messaging protocol to access Web services. It provides enterprises with more capabilities to connect systems internally and externally via the Internet. Its role is similar to that of DCOM and CORBA® distributed object systems—but is lighter weight and less programmer-intensive. SOAP is becoming more widely used to invoke services and as a messaging system over the Web.]*



Even Better Architecture with Struts

In this enhanced version of the same Web application architecture you just looked at, a very popular open-source Web application framework, called Struts, is being leveraged.

Apache's Struts Web Application Framework is becoming a standard as more companies are using it for their Web applications. It is a method for resolving Web application issues such as: Should the servlet controller perform error handling? Is there a simple way to change the panel-to-panel flow? What about internationalization?

With Struts, the servlet is supplied, and whatever needs to be coded becomes an Action class in Java, which in turn, calls the actual business logic. The base Action class is supplied by Struts. The decision about which Action to invoke for a given Web page form, and which JSP to show next, is guided by a Struts config file (shown here in red text). This allows the flow to be altered without modifying the code. Struts also make it easier to build internationalized Web pages by using externally described text. (See "xlatable msgs" shown here in red text.)

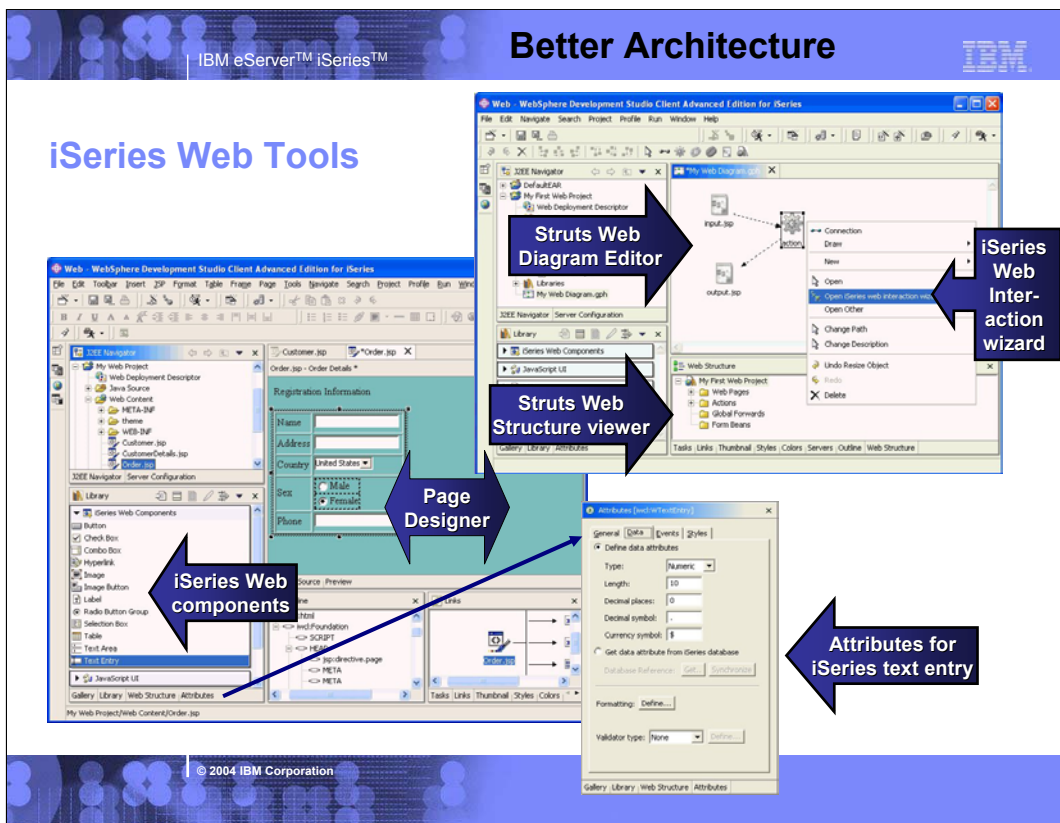
Struts further simplifies the processing of the user input by using "form beans" to capture all the user input from the input page (shown here in red text). For each input field in the input page, the data is copied to a property in the bean with the

Previous Slide Notes cont'd

same name as the field. This bean is then passed to the Action, which can further pass it along to the business logic.

The iSeries Web tools include all the necessary tooling for Struts and will generate everything in the Web application tier, after you pictorially "layout" the application flow. Again, this allows the iSeries developer to focus on the RPG or COBOL business logic, using existing programming skills.

For information on the Struts framework, see the Struts Web site listed on the "Additional Information" page at the end of this course.



iSeries Web Tools

These screen captures illustrate a bit of the functionality provided by the iSeries Web tools.

The picture in the upper right shows the Web Diagram editor. This editor lets you layout the flow of a Web application by dropping into the diagramming “white space” the various icons that represent Web pages and Actions. Then, you can use the iSeries Web Interaction wizard, per Action, to identify the programmatic or procedural business logic to be called. Then, the Web pages and Action are generated for you.

The picture in the lower left shows the Page Designer being used to create a new page or to enhance an already generated one. The Library palette in the lower left shows the iSeries-friendly Web components that can be dragged and dropped to the Web page designer. Also shown here is the Attribute view for a selected entry field iSeries Web component. As you can see, it supports attributes for data type, length, and decimal places, as well as edit code/word formatting, and compare and range validity checking. This is all similar to display files and SDA, but in this case, JavaScript is generated in lieu of DDS.

IBM eServer™ iSeries™

Step 4. Better Portability

- **New e-business Solutions**
 - ▶ Using Java, Web, & Web services tools in WebSphere Development Studio Client
 - Even business logic is written in Java (not EJBs, though)
- **Value Proposition**
 - ▶ Allows developers to create e-business solutions
 - Object oriented & portable
 - ▶ Expand market for ISV solutions
- **Education Enablement**
 - ▶ Java
 - Introduction, Advanced, IBM Toolbox for Java, OO
 - ▶ Detailed e-business (including Java code)
 - How to write servlets & JSPs, Javascript, HTML
 - Struts, including Struts actions
 - Web services, XML parsing
 - ▶ Tools
 - Eclipse Java tools & iSeries extensions
 - iSeries Web, Struts, Web services, & iSeries ProgramCall wizard
 - ▶ WebSphere Application Server
 - Beyond WebSphere Application Server - Express

© 2004 IBM Corporation

Step 4. Better Portability

In this step, the business logic is written in J2SE Java (remember, that's Java 2, Standard Edition)—it is not coded in RPG or COBOL. The remainder of the application is the same as it was when you were still operating at the “Step 3” level of evolution toward J2EE.

This change to J2SE Java is valuable because it allows the business logic to be ported and deployed to any server that runs a Java Virtual Machine™ (JVM™). In other words, for the first time, your code does not have to reside or execute on an iSeries server. This means that solution providers can extend the market for their applications.

This change in the language that is used to write the business code also allows for the incorporation of objects and components, as well as the many Java industry tools and standards that are available—such as design patterns and UML (Unified Modeling Language. [UML is an object-oriented, analysis and design language from the Object Management Group™ (OMG™)].

For this step, the Java tools and their iSeries extensions will be used in WebSphere Development Studio Client.

Previous Slide Notes cont'd

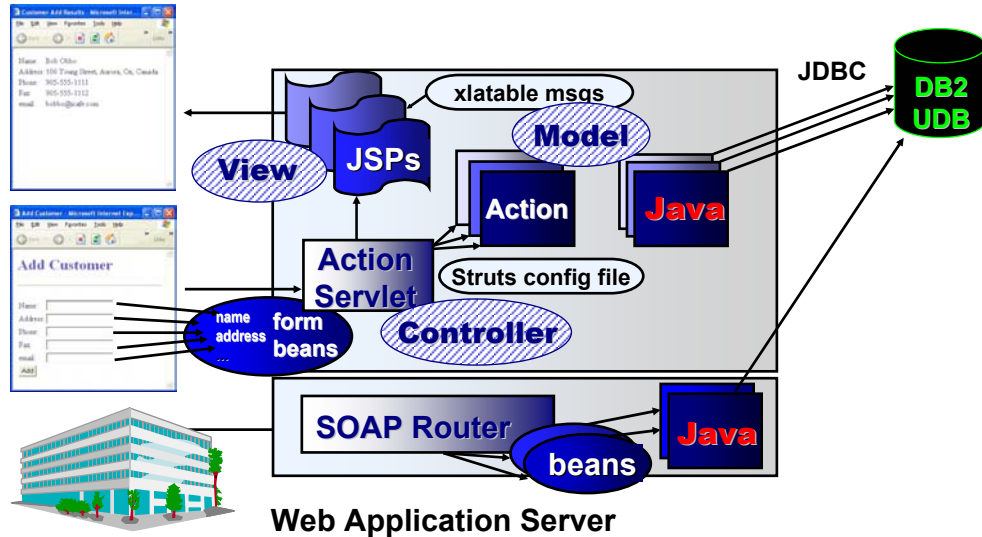
What “delta” skills are needed to take this fourth step?

- Java introduction
- Advanced Java: JDBC, etc.
- IBM Toolbox for Java
- Java tools and iSeries Java tools in WebSphere Development Studio Client

Skills that will help:

- OO and UML
- Design Patterns
- Rational Rose
- SQL

Better Portability



Better Portability

In this architectural diagram, you will notice that Java is used for the business logic. Just to reiterate, there is no RPG or COBOL code in this architecture... that would prevent the application from being portable... which now, is a primary goal.

The Java business logic runs in the second tier (shown in red text), but it is also possible to run it on the third tier (also shown in red text) by using technology such as Remote Method Invocation (RMI). In this case, each Action essentially calls one or more Java class to perform the needed business logic.

Please note that it is not recommended that the business logic be actually placed into the Action class. You can see in this chart that, in the second tier, the Java code stands alone from the action code.

Also, be aware that the Java code will probably use JDBC to access the data.

IBM eServer™ iSeries™

Step 5. Better Scalability

- **Create Highly Scalable e-business Solutions**
 - ▶ Using J2EE, EJB, Java, Web, & Web services tools
- **Value Proposition**
 - ▶ Create portable, J2EE-compliant solutions for high-volume, high-transaction-rate, e-business environments
- **Education Enablement**
 - ▶ J2EE
 - Enterprise JavaBeans (EJBs), Message Driven Beans (MDBs)
 - J2EE services (e.g., JMS)
 - J2EE security, LDAP, JNDI
 - ▶ Tools
 - Eclipse Java tools, J2EE tools, & iSeries extensions in WebSphere Development Studio Client
 - Tools: iSeries Web, Struts, Web services (including iSeries ProgramCall wizard)
 - ▶ WebSphere Application Server V5.0 & ND

Web
Web
Services
EJBs, MDBs
iSeries Java
tools
iSeries Web
tools
EJBs, J2EE
tools

© 2004 IBM Corporation

Step 5. Better Scalability

In this final step, the J2SE Java logic is replaced with full-blown J2EE EJBs and Message Driven Beans (MDBs). Doing this allows full exploitation of the power of J2EE, both for functionality and for object-oriented concepts. It also permits the enterprise to tap into the highly expert, J2EE developer community.

[NOTE: While this course is not intended to fully explain the intricacies of Enterprise JavaBeans, let's take a moment to discuss them briefly. EJBs are server-side Java logic that implements a business object—exposing a simple set of methods to the rest of the application, while internally handling all the complexity of multiple data sources and transactions. EJB Containers have to supply many Java services (dictated by the J2EE specification) to the EJBs running in them, but they are more than a set of APIs. They manage the EJB lifecycle and encapsulate one set of EJBs from another. EJBs come in two flavors, session beans and entity beans. Entity beans are the business objects that encapsulate complex access to one or more data sources. They represent business entities such as 'employee,' 'customer,' or 'widget'—while sessions beans offer a tailored view of one or more EJBs for the purpose of driving a particular application's Web page, for example, and contain no persistable data themselves. The classic example of an EJB is a shopping cart session bean that references inventory entity beans. It is the session beans that are distributed from the third tier to the second tier where they are leveraged by Struts action classes, for instance.

Previous Slide Notes cont'd

The third type of EJB, the message driven bean, is an event handler bean that is invoked when an asynchronous message arrives of a message type for which they are registered. Messages come and go via the Java Messaging Services support in J2EE.]

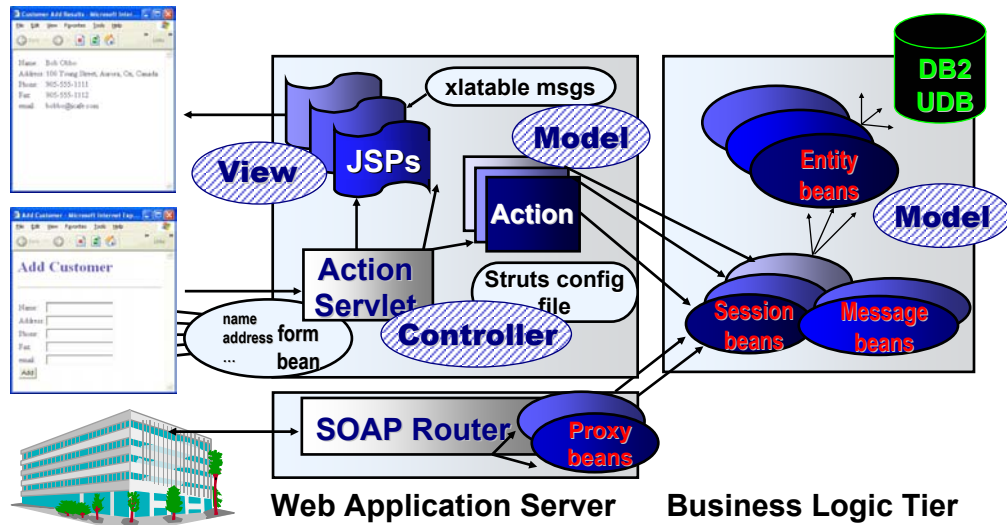
The fact that the EJBs run on the third tier also means the application will scale very well, and with the proper use of session beans, can even be spread over numerous servers.

For this step, the tools used for creating the EJBs and for managing their persistence to the database are provided in the WebSphere Development Studio Client Advanced edition.

What are the “Delta” skills you will need to take this fifth step?

- J2EE: EJBs, MDBs
- WebSphere Development Studio Client and WebSphere Studio Application Developer EJB and J2EE tools
- Transactions

Better Scalability



Better Scalability

Once again, you are looking at an architectural diagram... an optimal architecture that has been enhanced to use EJBs and MDBs in the third tier (shown in red text).

The important idea to take away from this diagram is that the Entity Beans are persisted to the database, and then session beans are used to "front end" the entity for a particular application. The message-driven beans are used if the Java Messaging Service is also being utilized for asynchronous message support.

The session bean proxies are called from the Action classes in the second tier. Similarly, a session bean proxy can be exposed as a Web service.

▪ Other great things to explore...

- WebSphere Portal Server
- WebSphere Commerce
- WebSphere MQ for iSeries
- WebSphere Application Server ND
- WebSphere Personalization for iSeries
- Lotus Domino
- Tivoli Access Manager

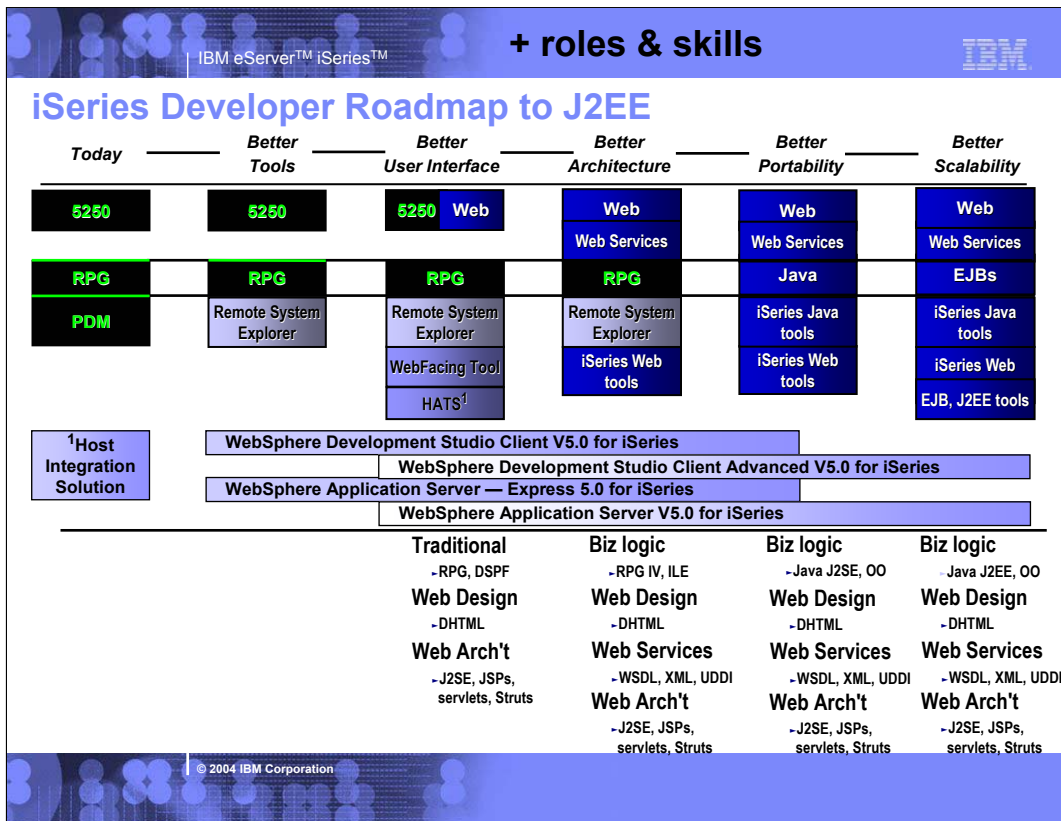
Other great things to explore...

There is literally a plethora of additional Java and WebSphere-related technologies and products that can be exploited along the way. These include:

- WebSphere Portal — Helps improve employee productivity and strengthen relationships with customers and trading partners by allowing users to interact, in a personalized way, with dynamic information, applications, processes, and people
- WebSphere Commerce — Provides powerful sell-side solutions to handle the challenges encountered in customer and trading partner environments, helping you implement B2C, B2B, or private exchange business models using open, industry-accepted standards
- WebSphere MQ for iSeries — Connects business software together to form one efficient enterprise by providing an open, scalable, industrial-strength messaging backbone
- WebSphere Application Server Network Development Edition (ND) — Delivers world-class caching, high availability, and industry-leading Web services support on top of the base WebSphere Application Server foundation
- WebSphere Personalization for iSeries — Provides WebSphere Advanced users with the capabilities to build a Web site, intranet, or extranet to deliver customized Web pages to each site visitor

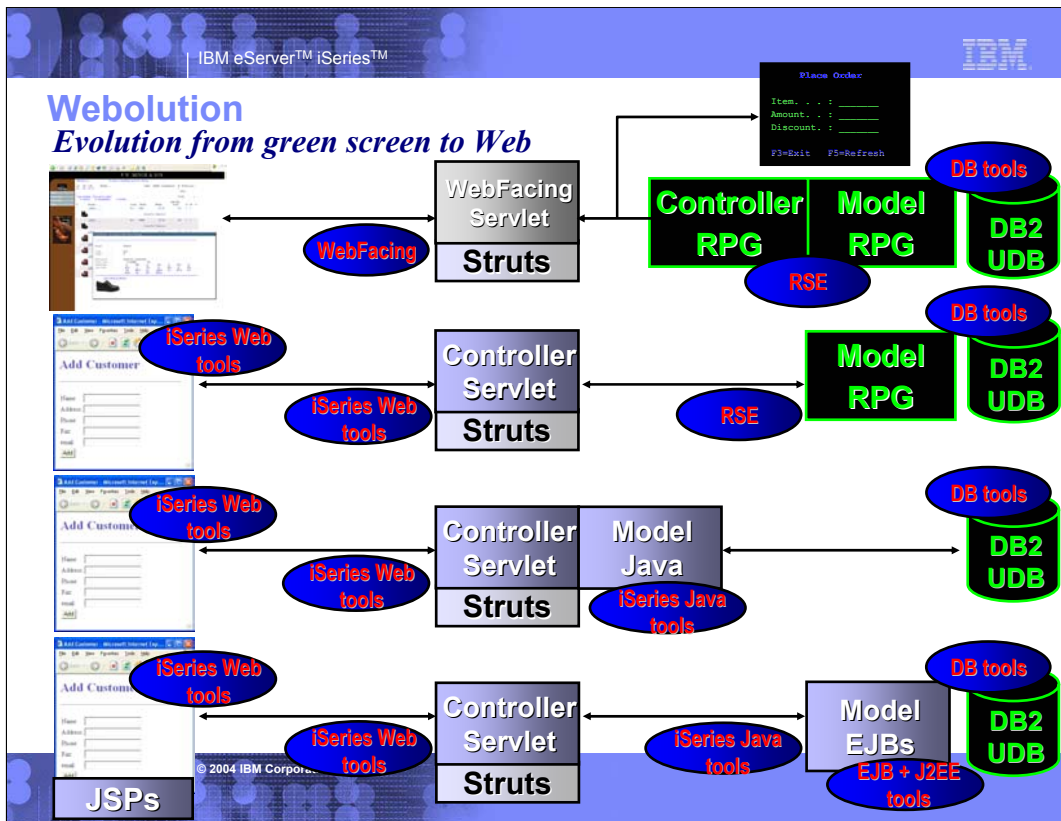
Previous Slide Notes cont'd

- Lotus Domino Toolkit for WebSphere Studio — Enables easy creation of integrated Domino and WebSphere applications by connecting the Domino database to WebSphere Studio
- Tivoli® Access Manager for e-business — Policy-based access control solution for e-business and enterprise applications, enabling new and rapidly scaling e-business initiatives to reach new markets and customers



iSeries Developer Roadmap to J2EE (with skills needed)

As a recap, this version of the roadmap, that you saw very early in this course, shows an overlay of the technologies you will need to understand to embrace that step or model.



“Webolution”

This diagram illustrates, pictorially, the evolution of a green screen application from its totally non-enhanced state to that of being a J2EE-portable, Web-enabled application.

From the top of this screen, row one shows a 5250 application that has only benefited from the process of converting its green screens using “better user interface” to build Web user interfaces via the IBM WebFacing Tool for iSeries or HATS.

Row two depicts the fact that this application is now in the “better architecture” step where pure RPG business logic is driven by the Struts controller logic.

Row three demonstrates an additional improvement (the “better portability” step) where the RPG business logic has been completely replaced with J2SE Java logic on tier 2.

Row four represents the “better architecture” step, with EJBs replacing the J2SE Java logic that runs on tier 3.

The dark blue ovals (with red text) show the tools used to enhance or replace parts of the application as it proceeds along the J2EE roadmap.

IBM eServer™ iSeries™

WebSphere Development Studio V5

The ubiquitous tools package for iSeries AD
ibm.com/software/awdtools/wds400

Unlimited Licenses

WebSphere Development Studio Client V5

iSeries	iSeries	iSeries	iSeries	IBM Web-Facing Tool	iSeries Projects
Java	Debug	Struts Web	Web Service		RSE
Trace	Profiling	DB	XML	WebSphere Application Server	

RPG

COBOL

C/C++

PDM et al.

V5R1 or V5R2
5722-WDS

No charge upgrade: 5905 (V5R1) or 5913 (V5R2)

+CODE
+VisualAge RPG

© 2004 IBM Corporation

WebSphere Development Studio V5

Here, you can see a chart of all the components that are a part of WebSphere Development Studio Version 5. It comes with the host compilers and tools, and offers unlimited licenses of WebSphere Development Studio Client Version 5 (standard).

WebSphere Development Studio Client (shown in medium blue) is a superset of WebSphere Studio Site Developer Version 5, with iSeries-unique extensions having been added (and shown here in light blue).

- There are iSeries extensions to the Java tools for accessing, exporting to, and running on an iSeries server.
- iSeries extensions are available to the debugger in order to allow the debugging of RPG, COBOL, C, C++, and CL code.
- iSeries extensions to the Web tools allow for the creation of Struts actions with RPG and COBOL business logic and for constructing Web pages by leveraging existing SDA skills.
- iSeries extensions to the Web Services tools allow for the building of a Web service from RPG or COBOL business logic.
- The IBM WebFacing Tool for iSeries lets the developer create a Web user interface from a 5250 user interface (by interpreting the DDS).

Previous Slide Notes cont'd

- Remote System Explorer (RSE) is intended for managing, editing, compiling, and debugging RPG, COBOL, DDS, CL, C, and C++ code. This means that once you get into step four or five, which are both exclusively Java application environments, you will have weaned yourself from your legacy application coding methods... at least from the perspective of new application development.
- iSeries Projects support is available for iSeries off-line and team-based development.

IBM eServer™ iSeries™

WebSphere Development Studio Advanced V5

The advanced tools package for iSeries AD
ibm.com/software/awdtools/wds400

Unlimited Licenses

WebSphere Development Studio Client Advanced V5

iSeries	iSeries	iSeries	iSeries	IBM Web-Facing Tool	iSeries Projects
Java	Debug	Struts Web	Web Service		RSE
Trace	Profiling	DB	XML	WebSphere Application Server	EJB* J2EE*
					Test case*

RPG **COBOL**

C/C++ **PDM et al.**

V5R1 or V5R2
5722-WDS
Feature: 5906 (V5R1) or 5914 (V5R2)

+CODE
+VisualAge RPG

© 2004 IBM Corporation

WebSphere Development Studio Advanced V5

The WebSphere Development Studio Advanced Version 5 offering is priced higher than the standard edition just discussed in the previous slide. However, it offers unlimited licenses of WebSphere Development Studio Client Advanced. WebSphere Development Studio Client Advanced V5 is a pure superset of the Standard, and of WebSphere Studio Application Developer V5 (shown in blue), with many iSeries-unique extensions also having been added (shown in grey).

The tools that are unique or are enhanced for WebSphere Development Studio Advanced V5 are shown with an asterisk on this chart:

- The iSeries Web tools support a Java Connector Architecture (JCA) for calling RPG and COBOL (you can still call these in the Standard version, but without the industry-standard JCA support).
- The IBM WebFacing Tool for iSeries supports Struts, as well as spool file management and printing that is available only in the Advanced version.
- The EJB, J2EE, and Testcases tools are unique to the WebSphere Development Studio Advanced V5 (having been inherited from Application Developer).

Conclusion

- There is an ordered roadmap from 5250 to Web.
- You can move to new language and development paradigms—while protecting IT assets.
- There are many ways to gain needed expertise to accomplish this move to the Web.
- It's time to get started!

www.ibm.com/series/roadmap

Conclusion

As you have gone through this course, we suspect it is possible you have run into many terms and concepts that are foreign to you—or are at best only fleetingly understood by you at present. If we tried to cover everything in detail that has summarily mentioned, this course would take weeks to complete. Instead, this course is intended to leave you with the following messages:

There is an ordered course in which to move your green screen-based application set to a much more progressive Web-based application set that supports your company's needs in today's always changing, ever demanding business environment.

This roadmap has been designed to allow you to move, ultimately, to new programming languages and new development paradigms in a manner that both achieves your company's goals in a timely manner and protect the valuable IT assets in which you already have invested (line-of-business applications and many man years of experience).

There is a great deal to learn... and an inordinate number of methods to gain that learning (on-line training courses, classroom training, Web sites, Redbooks and redpieces, white papers, supportlines, and much more).

The time to get started is NOW. Even with the first step along this roadmap, you will realize large improvements in programming efforts.

Additional Information


- **developerWorks: IBM's resource for developers**
ibm.com/developerworks/
- **WebSphere Application Server Network Deployment**
ibm.com/servers/eserver/iSeries/software/websphere/wsappserver/indexnd50.html
- **WebSphere Host Access Transformation Server (HATS)**
ibm.com/software/webservers/hats/
- **ASF Tomcat Web Application Server on iSeries**
ibm.com/servers/eserver/iSeries/software/http/services/tomcat.html
- **IBM WebFacing Tool**
 - For general details...
ibm.com/software/awdtools/wdt400/about/webfacing.html
 - For a Redbook...
www.redbooks.ibm.com/abstracts/sg246801.html
 - For a "guided tour"...
mc-store.com/unibmwetool.html
- **WebSphere Development Studio**
ibm.com/software/awdtools/wds400
- **The Eclipse Consortium**
- **WebSphere and Eclipse**
ibm.com/software/info1/websphere/index.jsp?S_TACT=103BHW06&tab=eclipse/index
- **WebSphere Application Server - Express**
ibm.com/software/websphere/info/express/index.jsp
- **The Struts framework explained...**
www.jakarta.apache.org/struts/
- **WebSphere Portal**
ibm.com/websphere/portal/
- **WebSphere Commerce**
ibm.com/websphere/commerce/
- **Lotus Domino Toolkit for WebSphere Studio**
ibm.com/software/info1/websphere/index.jsp?tab=products/studiotoolkits
- **Tivoli Access Manager for e-business**
ibm.com/software/tivoli/products/access-mgr-e-bus/
- **WebSphere Personalization for iSeries**
ibm.com/servers/eserver/iSeries/software/websphere/wspersonal/
- **WebSphere MQ**
ibm.com/software/integration/wmq/
- **DB2 UDB for iSeries**
ibm.com/software/db2/

Trademarks and 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
AS/400e	IBM (logo)
eServer	iSeries
	OS/400

Lotus and SmartSuite are trademarks of Lotus Development Corporation and/or IBM Corporation in the United States, other countries, or both.

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

Microsoft and Windows NT are registered trademarks of Microsoft Corporation 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.

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

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

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

Other company, product or 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.