

Table of contents

Introduction	2
Agenda.....	3
Goals of the roadmap.....	4
iSeries Developer Roadmap Version 3: Architecture.....	6
iSeries Developer Roadmap: Refactoring versus modularizing code	7
iSeries Developer Roadmap: Modular advantages.....	8
iSeries Developer Roadmap: Re-architecting	9
iSeries Developer Roadmap: Tools	10
iSeries Initiative for Innovation Program	11
iSeries Tools Innovation Program	12
iSeries Tools Innovation members.....	13
Improving productivity	14
Enhancing the user experience.....	15
Creating a modular architecture.....	16
Integrating applications	17
Integrating business processes.....	18
Traditional environment.....	19
Modernizing development tools	20
Improving programmer productivity.....	21
Remote System Explorer	22
Improving the user experience: GUI	24
Improving the user experience: Creating a better interface	25
Improving the user experience: Employee familiarity.....	27
IBM WebFacing Tool for iSeries	28
Architecture of the IBM WebFacing Tool for iSeries	29
WebSphere Host Access Transformation Services (HATS)	30
WebSphere HATS architecture.....	31
iSeries Access for Web.....	32
WebSphere Portal.....	33
The modularization process	34
Modular architecture: Quick response to growing needs	36
Modular architecture: Web-based applications	37
Application integration.....	38
Application integration advantages	39
Business process integration	40
Business process integration advantages.....	41
Integrating with business partners	42
WebSphere Business Integration.....	43
Summary.....	44
Additional information.....	45
Trademarks and special notices	46



@server®

iSeries. mySeries.

IBM iSeries Developer Roadmap, Version 3

Joint strategy from:
IBM Systems and Technology Group
and IBM Software Group

Fall, 2005

iSeries. mySeries.

Introduction

Welcome to this online course, entitled “IBM iSeries Developer Roadmap, Version 3.”

In this course, you will learn about the ideal method for a classic, green-screen IBM® eServer™ iSeries™ shop to move into the Web application world in a staged, nondisruptive manner.

The IBM Developer Roadmap has been designed to take into consideration the probable extent to which your shop is presently entrenched in a monolithic application model, as well as the large experience level of which you surely can boast, in relation to the traditional programming languages that support your green-screen applications.

Furthermore, Version 3 of this roadmap incorporates a large number of tools that are available from IBM solution providers to help in your modernization efforts.

You will learn a five-step evolution that can be implemented at a pace appropriate for your organization, budget, current skill set, and goals with the mission to become a Web application-driven enterprise.

Agenda

- **Goals overview**
- **Graphical user interfaces (GUI)**
- **Developer's tools**
- **Application integration**

Agenda

This course offers an overview of the iSeries Developer Roadmap, which provides the necessary steps for business application modernization. The various steps involved are discussed, including creating a modern graphical user interface (GUI), connecting various applications, and enhancing the end-user experience. In addition, various tools, including Remote System Explorer, the IBM WebFacing Tool for iSeries, IBM WebSphere® Host Access Transformation Services (HATS), and iSeries Access for Web will be discussed.

Goals of the roadmap

- Provide a step-by-step approach for modernizing applications
- Generate or build modern GUI interfaces
- Move to a modern architecture
- Merge new technology with traditional code
- Incorporate multiple entry and exit points, allowing gradual skill growth toward modernization
- Maintain a living document

Goals of the roadmap

The iSeries Developer Roadmap is now at Version 3. The original release provided an evolutionary path for developers to pursue when modernizing applications. It is evolutionary in that developers can gradually improve their ability to accommodate the ever changing needs of the business. Since the first version of the roadmap, IBM has listened to the experiences of solution providers and development shops as they have progressed down a path toward modernization resulting in clarification and refinement of the messages contained in the original roadmap strategy.

In Version 3, there is a defined step-by-step approach to build skills and to rework existing applications. These steps support an even easier evolution that does not pressure developers to prematurely attempt to apply technologies for which they might not be prepared. Depending on the requirements of the business, some parts of the modernization process might be accelerated, or skipped altogether.

There are several main goals for the roadmap:

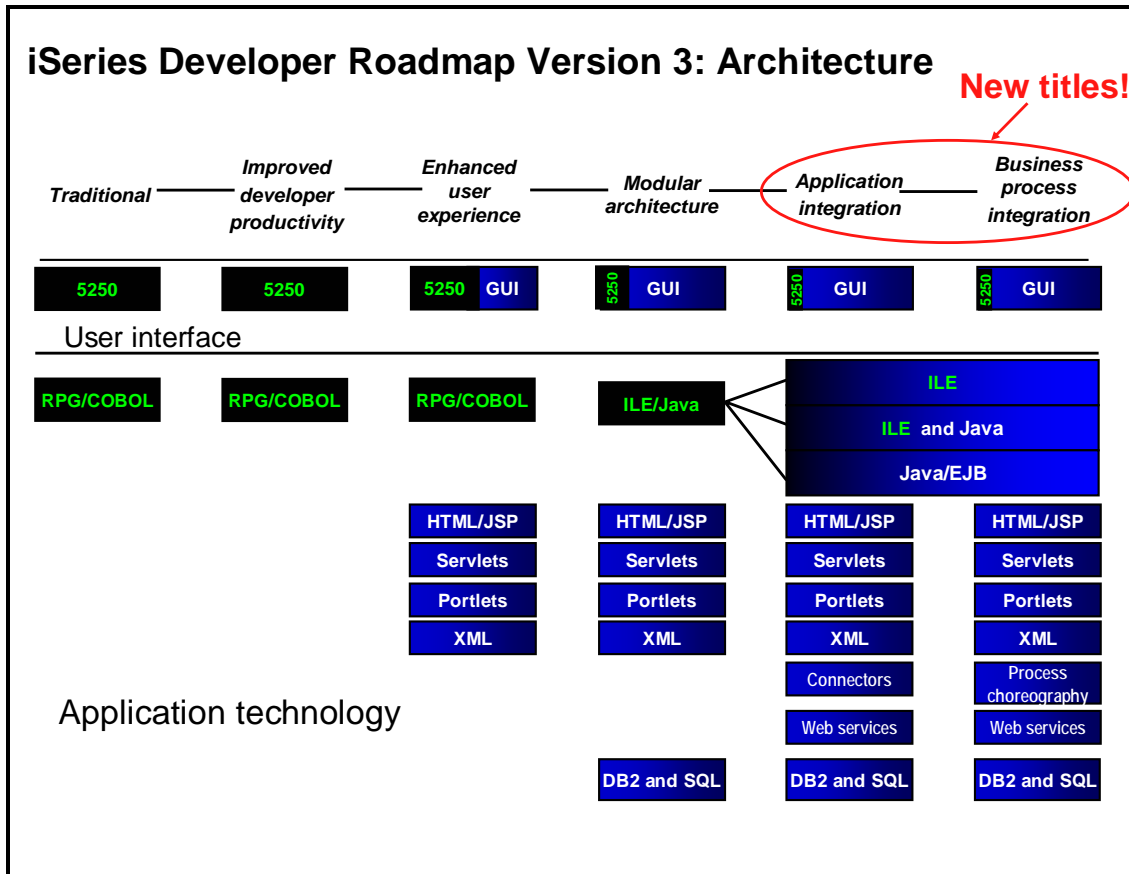
- The first mission is to transition the old-fashioned green-screen interfaces to GUIs that are more consistent with PC-based applications. Using a mouse and Windows® has become commonplace in many desktop programs. Making the line of business (LOB) applications follow that same graphical paradigm will increase the level of comfort for the user.
- Secondly, moving from a traditional, multi-function structured program to one that is more modern is critical. Using modern compilers allows developers to incorporate additional functions that are only available with the most current versions of the languages. Once the applications are modernized, you can start realizing the benefits of implementing an Integrated Language Environment (ILE) architecture. ILEs allow developers to implement the basic constructs of object-oriented technology. This includes things such as code modularity, code reusability, and easier distribution of maintenance activities among development staff.

- Next, when necessary to meet the needs of the business, a move to a service-oriented architecture might be the objective. This type of architecture allows applications to talk to each other, both within and across enterprises.

There are other important points to remember about the iSeries Developer Roadmap. The key objective is to modernize application code and bolster developer skills (on both an individual level and within an entire development shop). By restructuring code into smaller modules, developers are positioning their applications for future integration with emerging technologies. An example is the use of Web services, a function that is next to impossible if the application code is still in a large, multi-functioned program structure.

The roadmap accommodates applications at all stages of modernization. Based on existing skills and technologies, shops can determine which step is the most appropriate entry point. Also, based on long-term objectives, developers can decide where to leave off certain phases of modernization, allowing them to gradually build skills as required to implement business objectives.

Finally, Version 3 of this roadmap is still a living document. It will continue to be enhanced and clarified, based on feedback and experience from iSeries customers, partners, and software vendors.

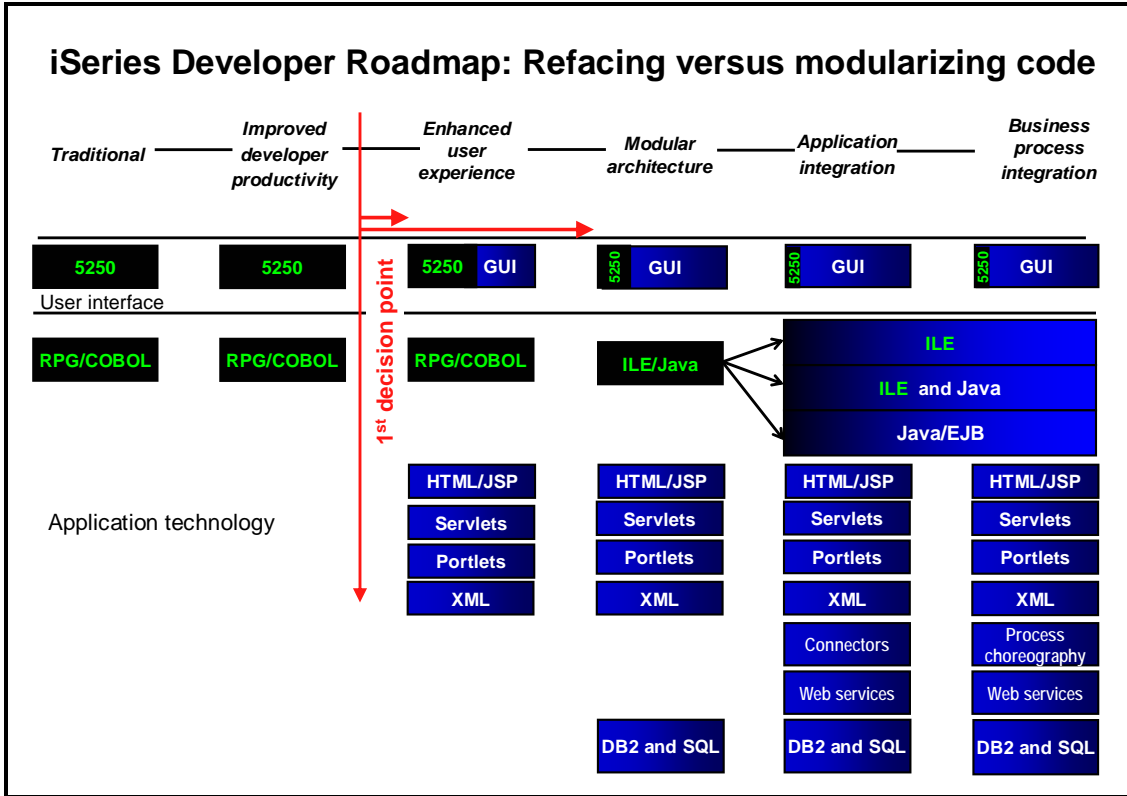


iSeries Developer Roadmap Version 3: Architecture

This is Version 3 of the Developer Roadmap, and although it would seem as though there are major changes to the steps, it is, in fact, only a clarification of some of the more subtle messages inherent in the first two versions.

One major enhancement in Version 3 is that the steps shown in the two right-most columns have new titles that more accurately indicate the final processes required to attain **application integration** and **business process integration**. Instead of the former titles (**better portability** and **better scalability**, respectively), the new titles reflect the intent of eBusiness:

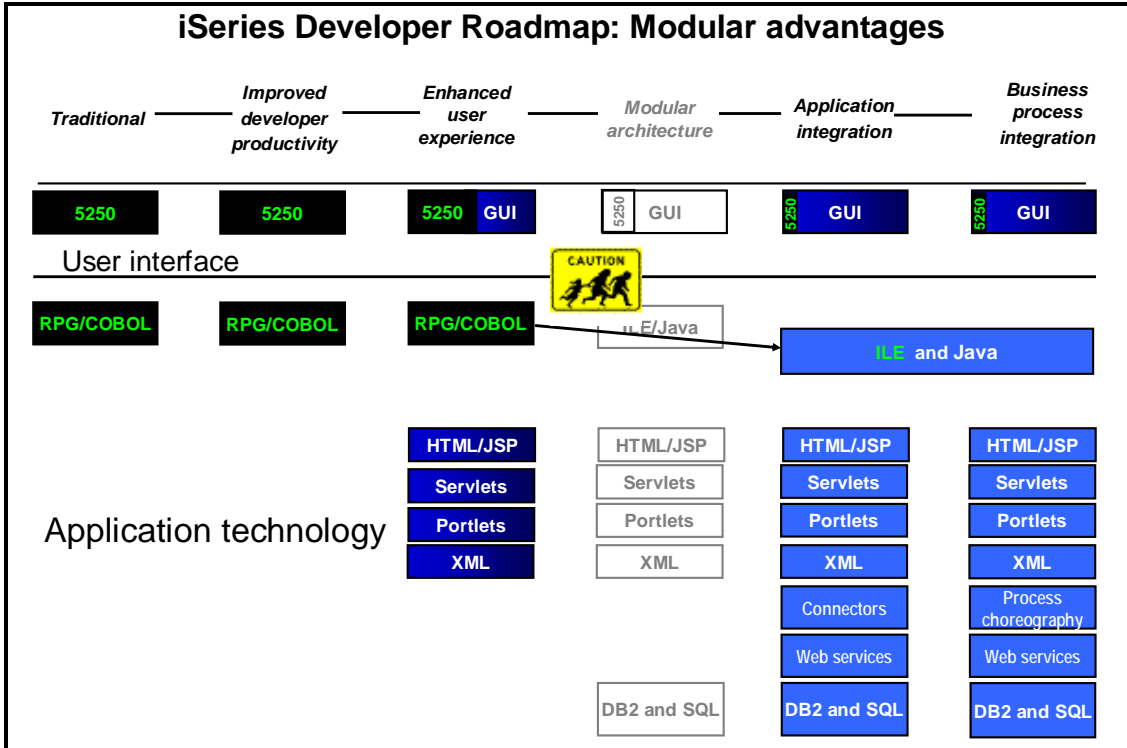
- Application integration reflects the ability to build modular applications through the use of technologies such as: Web services, XML, and connectors.
- Business process integration implements modular application technologies across enterprises. This involves the use of business process modeling tools. It also establishes and maintains links with trusted partners to execute multiple components of the business applications.



iSeries Developer Roadmap: Refactoring versus modularizing code

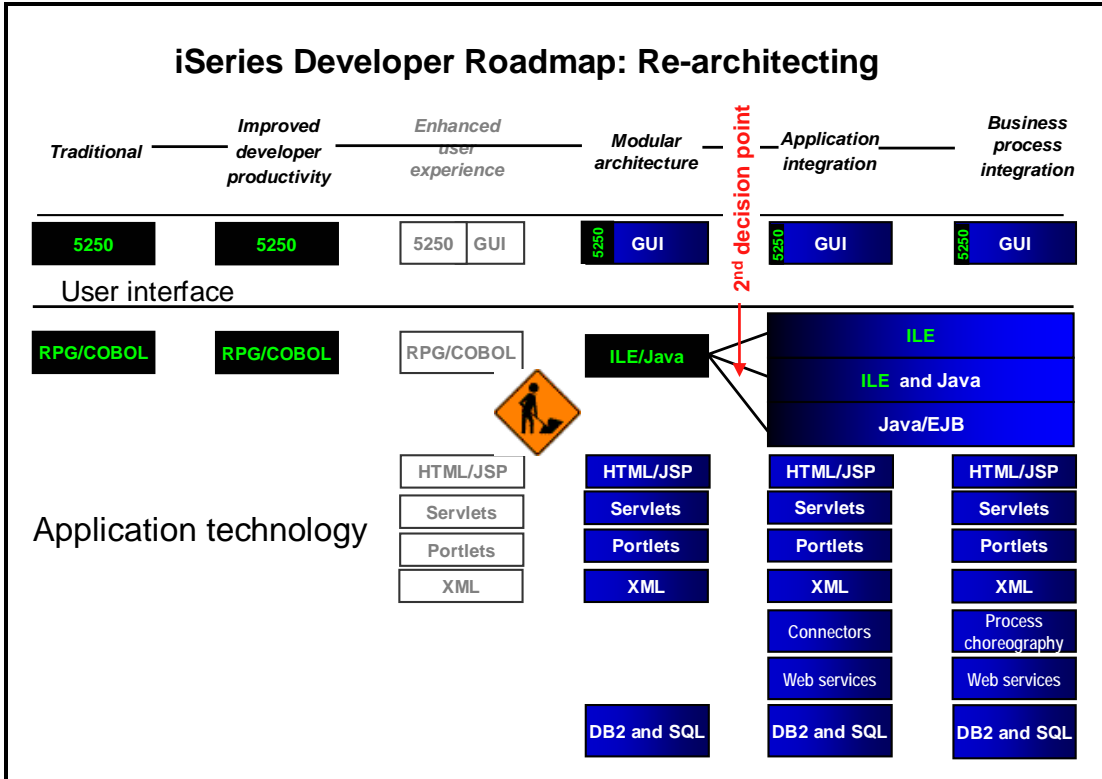
It has become evident that those who are modernizing existing application code will need to decide on long-term strategies for the effort once the development environment has been placed on the desktop. If the business requirement is to put a graphical interface onto an existing application, then an enhanced user interface option is certainly viable. However, if the application has to be capable of including new technologies, a more appropriate choice is to skip the refactoring step and instead modularize the existing code set, thereby producing a graphical user interface in the process. It is at this time that developers can implement some of the modern development constructs, such as maximizing reusability while at the same time mitigating risk.

Some development shops might choose to create graphical interfaces as an intermediate solution, thus allowing the necessary time for larger re-architecting projects. This is a perfectly viable strategy also. The IT staff needs to understand the rationale for picking one path over another before a decision is made.



iSeries Developer Roadmap: Modular advantages

As stated previously, picking an enhanced user experience does not suggest that the application cannot be integrated with other technologies. It simply means that, without changing the fundamental structure of the underlying code, the classic style of coding RPG or COBOL for iSeries systems does not lend itself as easily to integration with additional technologies. There are things that can be done, such as performing the transfer of data via XML. However, it is a more difficult interface to develop, and most likely will be less robust in the final implementation.



iSeries Developer Roadmap: Re-architecting

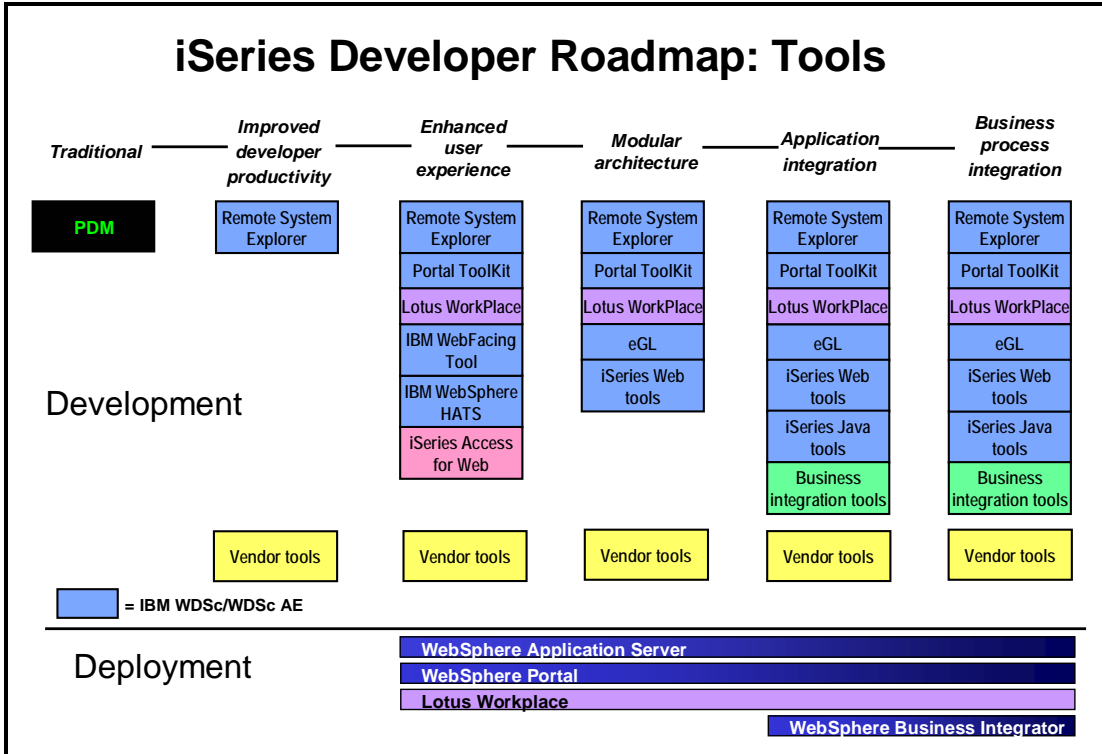
The best business decision might be to re-architect the application. This involves three efforts:

- Modularizing the business logic
- Moving the data-specific functions to the database engine
- Handcrafting new user interfaces

If re-architecting the application fits your needs, it will NOT be necessary to modernize the green-screen interfaces by using one of the screen replacement technologies.

Although IBM does not have tools to specifically assist in all of these re-architecting efforts, there are tools in the marketplace that will read through the application and provide reports to help you understand the structure of your monolithic code. Some tools also assist with the separation of business logic and display interface logic, whereas others guide you through the conversion from RPG II or RPG III code to the more progressive RPG IV style of code that supports modularized techniques.

Be sure to visit the eServer Tools Network site listed in the "Additional information" section of this course.

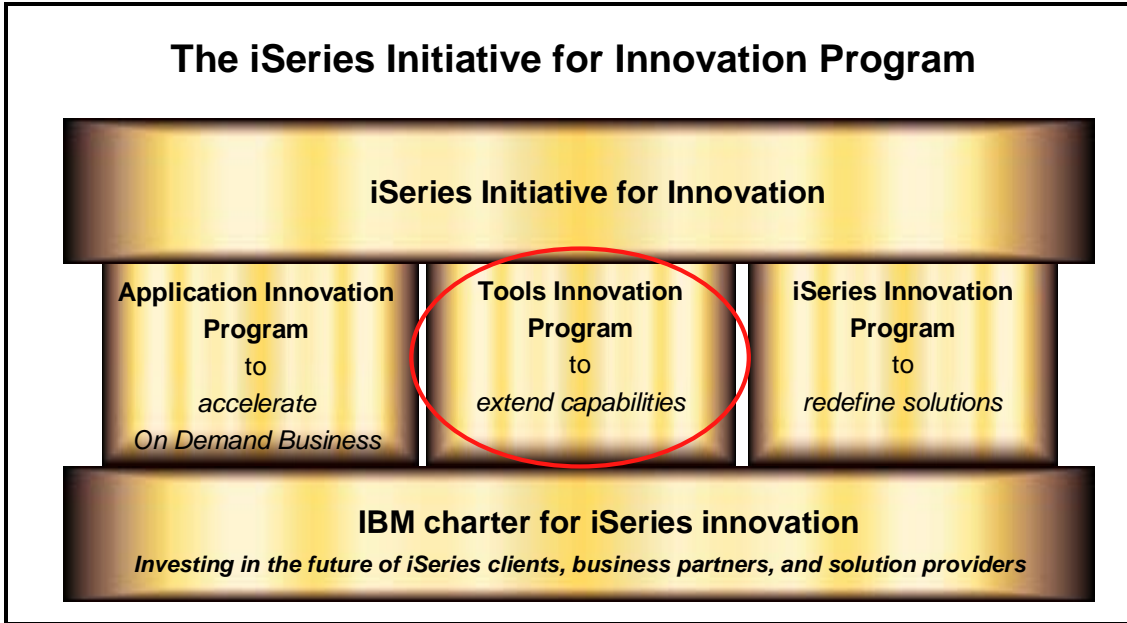


iSeries Developer Roadmap: Tools

This chart illustrates the various tools available to assist in implementing each of the technologies. The blue boxes represent components of the WebSphere Development Studio Client (WDSClient) or WebSphere Development Studio Client Advanced Edition (WDSClient AE). Most of the other colored blocks represent additional IBM tools

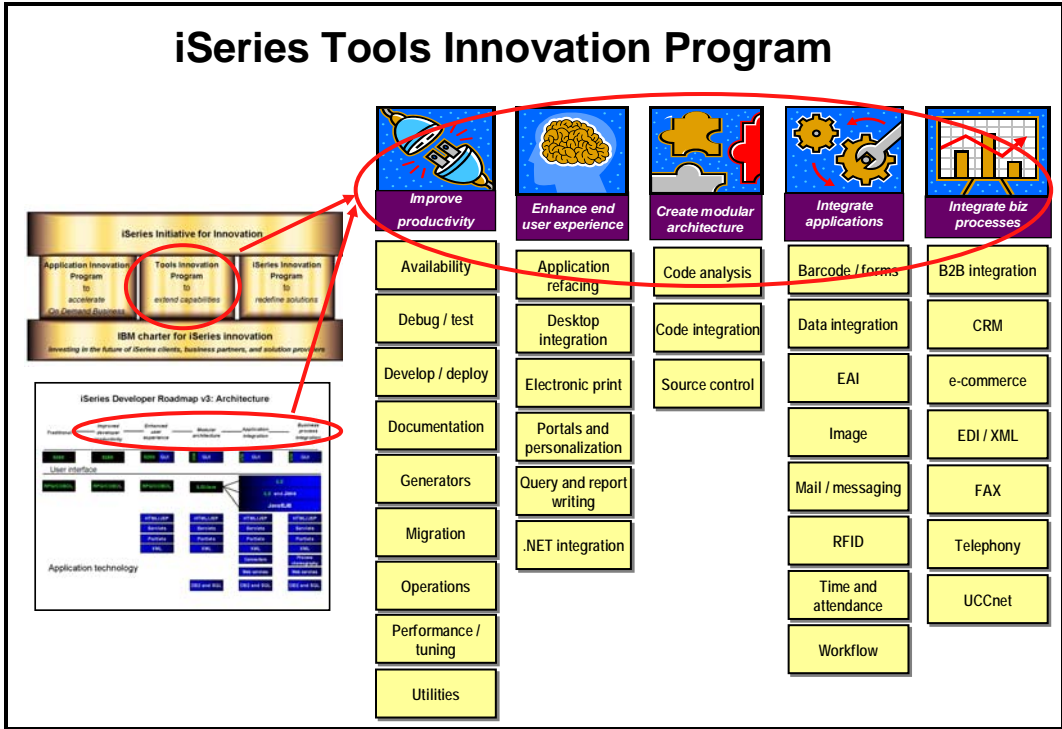
However, notice the yellow blocks. These are new in Version 3 of the roadmap and represent vendor tools that provide functionality that is needed in each of the technology categories. IBM does not currently offer a product for every aspect of implementing a modernization strategy. Therefore, filling in the gaps with vendor tools allows for a more complete set of modernization processes. It also accurately represents an ongoing, cohesive relationship between IBM and its vendors.

The IBM sponsored program that nurtures the development of tools and other software solutions with these third-party providers is called the **iSeries Tools Innovation Program** and is one of the three pillars of the **iSeries Initiative for Innovation**. We will talk about these programs next.



iSeries Initiative for Innovation Program

The **iSeries Initiative for Innovation Program** focuses on modernizing the capabilities of iSeries applications by leveraging a wider variety of excellent third-party tools and provider skills. The objective is to extend and add value to all applications, regardless of the underlying programming language. Some members featured here will be able fill functional gaps in IBM offerings. For example, if IBM does not currently have a tool to perform RPG analysis, one of these vendors can provide this function.



iSeries Tools Innovation Program

Within the **iSeries Tools Innovation Program** (the middle pillar), vendors are organized into one of five categories depending on their primary function. This makes it easier for developers to identify tools and vendors that meet their specific needs.

(Notice that the five categories within the iSeries Tool Innovation Program coincide with the five phases of the iSeries Roadmap.)

We will talk about each of these tools categories in more detail.



iSeries Tools Innovation members

Wow! There are lots of vendors shown on this screen, right? “You betcha,” as we like to say here in Minnesota. And, we are very proud of these and many other vendor relationships that are a result of the iSeries Tools Innovation Program.

Information about these vendors can be found on the iSeries Tools Innovation Web site listed in the “Additional information” section of this course.

Improving productivity

 Improve productivity	    
Availability	    
Debug / test	     
Develop / deploy	    
Documentation	   
Generators	    
Migration	    
Operations	
Performance / tuning	
Utilities	

Improving productivity

Many vendors who are members of the iSeries Tools Innovation Program have slotted their products into the **Improve productivity** column. These tools are used in areas such as: availability solutions, testing and debugging, developing and deploying, documenting, as well as other aspects of improving productivity.

Enhancing the user experience



Enhance end user experience

- Application refacing
- Desktop integration
- Electronic print
- Portals and personalization
- Query and report writing
- .NET integration



Enhancing the user experience

Additionally, some vendors who are members of the iSeries Initiative for Innovation Program have slotted their products into the **Enhance the user experience** column. These tools are used in areas such as application refacing, desktop integration, portals, personalization, and other aspects of improving the experience for the end user of the application.

Creating a modular architecture

Create modular architecture

Code analysis

Code integration

Source control

ASNA

DATABOROUGH
legacy enhancement solutions

Arcad
Software

ADVANCED
BusinessLink

L.D. CONSULTING INC.
Operational Information Systems

Creating a modular architecture

Other vendors have slotted their products into the **Create a modular architecture** column. These tools are used in areas such as code analysis, code integration, source control, and other aspects of creating a modular architecture.



Integrating applications

Some vendors have slotted their products into the **Integrate applications** column. These tools are used in areas such as data integration, mail and messaging, workflow, and other aspects of integrating multiple applications.

Integrating business processes

Integrate biz processes

B2B integration

CRM

e-commerce

EDI / XML

FAX

Telephony

UCCnet

Jacada

BinaryTree, Inc.

EXTOL

XL ExcelData

nuBRIDGES

LINOMA SOFTWARE

iMS Messaging Systems, Inc.

IBM

ASC

RJS Software Systems

eOne

inoVIS

MAGIC www.magicsoftware.com

Real Vision Imaging

PATRICK TOWNSEND & ASSOCIATES

L.D. CONSULTING Operational Information Systems

QUADRANT SOFTWARE

LAOSA ADVANCED SOFTWARE MADE SIMPLE

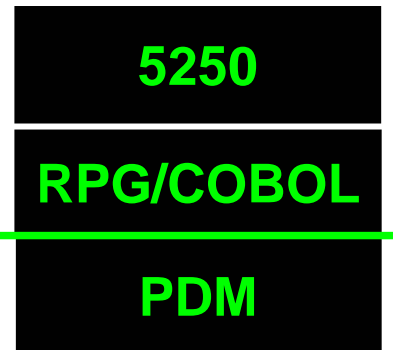
Integrating business processes

Finally, many vendors have slotted their products into the Integrate business processes pillar. These tools are used in areas such as B2B integration, e-commerce, telephony, and other aspects of integrating various business processes.

Traditional environment

■ In a traditional environment, developers:

- ▶ Create or enhance 5250 applications
- ▶ Maintain applications in large multi-functioned programs
- ▶ Use a traditional, host-based development environment
- ▶ Use traditional code languages

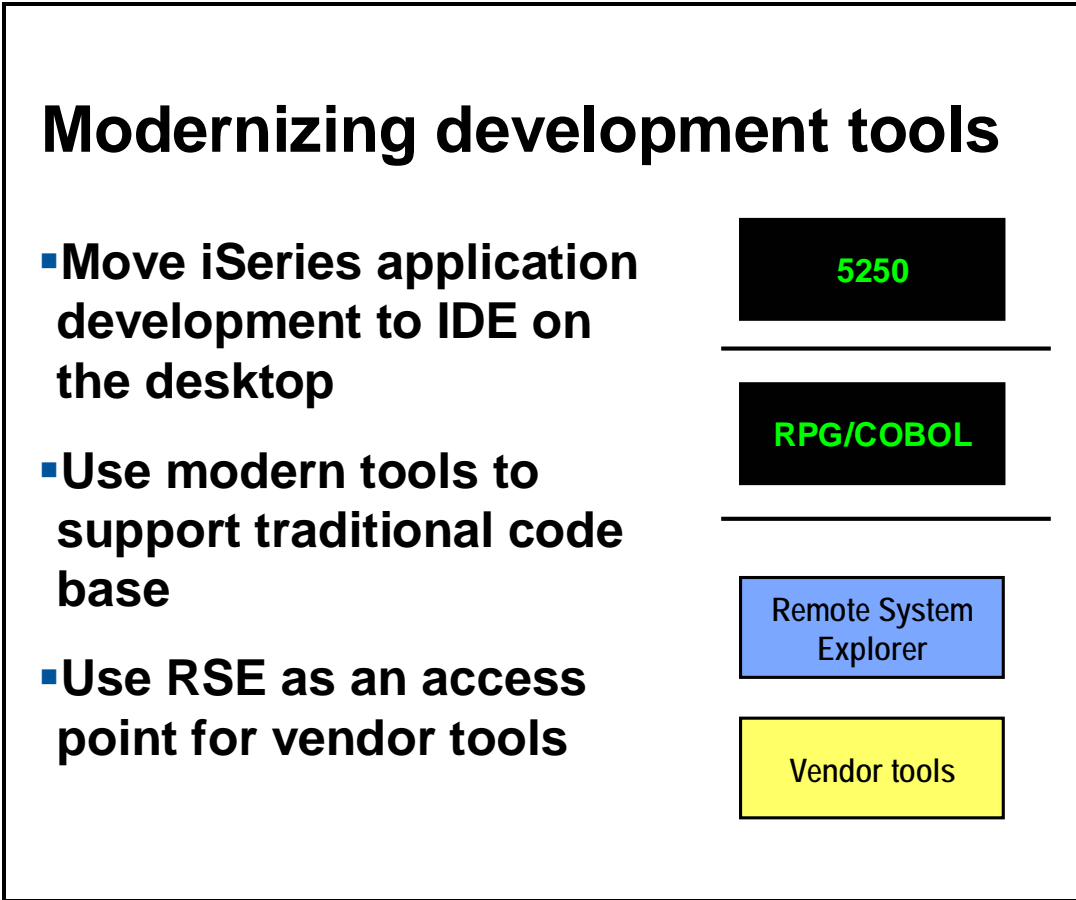


Traditional environment

Traditional development efforts depend on green-screen user interfaces and are driven by server-based tools. Programmers in this environment use languages such as RPG and COBOL (although some have branched into C and C++). Most of these shops depend on CL as their operational control environment.

In most iSeries developer communities, the IBM Application Development ToolSet (ADTS) (available since the introduction of the earlier IBM AS/400® family of servers), has been an important programming aid. The components of ADTS are:

- **Program Development Manager (PDM):** A list-based interface available for exploring libraries, files, and objects.
- **Source Entry Utility (SEU):** The green-screen editor used for entering source code on the system.
- **Screen Design Aid (SDA):** A WYSIWYG editor that allows you to build 5250 green-screen images; Display file DDS is created internally.
- **Report Layout Utility (RLU):** A WYSIWYG editor that supports the need to build externally described print files. Printer file DDS is created internally to do this.
- **Data File Utility (DFU):** A tool for accessing and updating data files.
- **Interactive System Debugger (ISDB):** This allows you to debug programs while viewing the source.
- **File Compare and Merge Utility (FCMU):** A split-screen representation of two versions of the source code that allows the comparison and optional merging of updates between versions of the code.



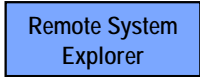
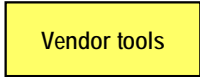


Modernizing development tools

The first step in the iSeries Developer Roadmap does not involve any change to the applications currently in use. Instead, it involves replacing the traditional development tools with more modern, productive methods to support the same code base. Ultimately, the application code will still be written in a traditional language such as RPG or COBOL and will still have a green-screen user interface using DDS. However, by working in an integrated development environment (IDE) based on IBM Eclipse, application development is moved to the desktop.

Remote System Explorer (RSE), which comes with WebSphere Development Studio Client, is the 21st century follow-on to the tools mentioned earlier: PDM, SEU, SDA, RLU, and the system debugger. RSE offers a highly productive environment that is integrated with the Eclipse IDE. RSE is also the point of integration for some iSeries tool vendors, many of whom have already released plug-ins to complement the functions provided by IBM.

Traditionally, vendor tools fall into a primary category, such as: change management, utilities and documentation, generators, as well as debug and testing tools. Vendors might also fall into a secondary category, such as: availability, migration, operations, performance, and tuning.

Improving programmer productivity	
<ul style="list-style-type: none">▪ Intuitive graphical user interfaces and familiar tree views	
<ul style="list-style-type: none">▪ Easy access to advanced application development tools	
<ul style="list-style-type: none">▪ Consolidated IDE for multiple types of projects	
<ul style="list-style-type: none">▪ Workstation centric development for a reduced development workload on the iSeries system	

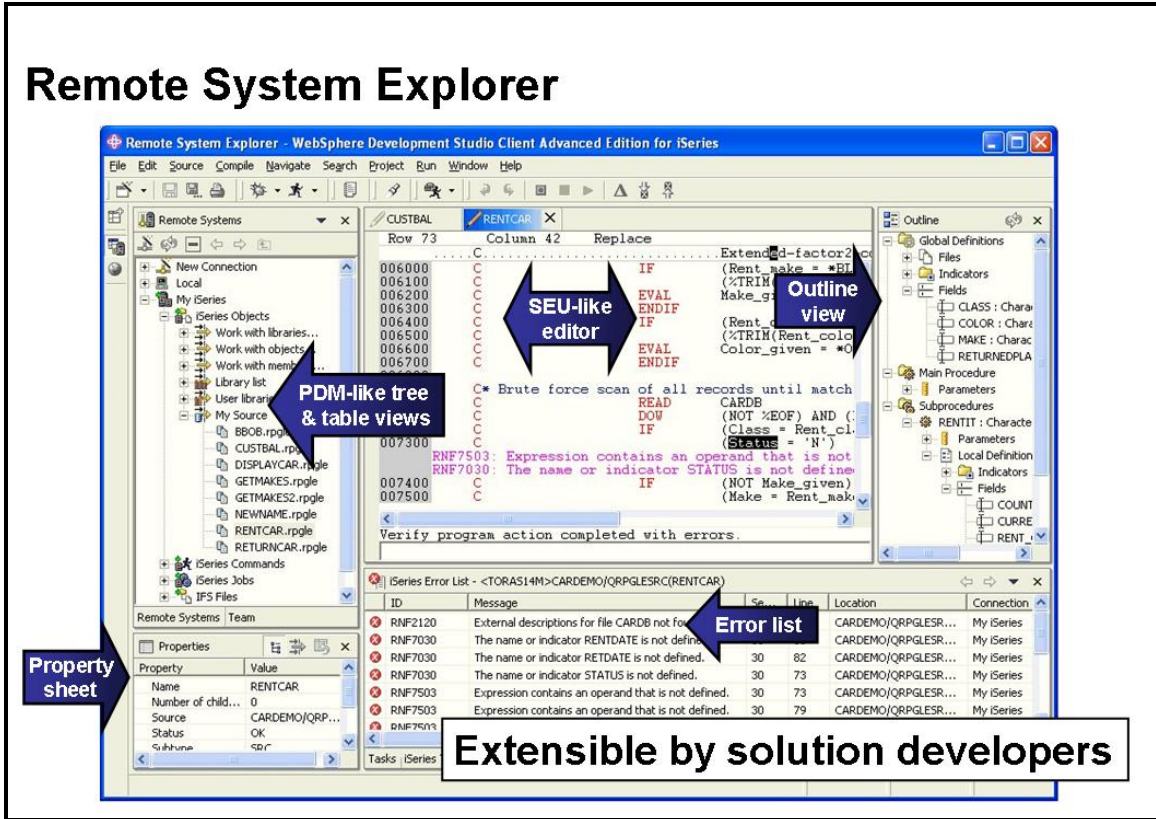
Improving programmer productivity

Moving the development staff to a graphical, integrated development environment (IDE) significantly improves productivity. This is not an immediate benefit, but one that will come quickly as programmers discover the added functions available in the desktop tools. Some functions intrinsic to a development environment, such as screen design and report layouts, are easier to manage graphically.

Many tool vendors have integrated components of their product to the IDE tools. This adds to the productivity benefit, as a developer can manage the total environment from one user interface on the desktop.

Skills that are developed while using the Remote System Explorer can be transferred to other development environments as there are many common behaviors and key sequences. This is the nature of the Eclipse IDE.

The final advantage of using the desktop tools available in WebSphere Development Studio Client is the intangible benefit of reducing the environment of rivalry that sometimes exists in a multilingual (RPG and Java™) shop. Because both groups of programmers are working in the more productive desktop environment, they are using the same tools to do similar tasks, even though they are producing code in different languages.



Remote System Explorer

The collection of views and editors in RSE is called the Remote System Explorer perspective. Developers can open and flip between multiple perspectives by using icons in the bar on the left side of the screen.

The primary view in the RSE perspective allows you to create connections with remote iSeries, UNIX®, Windows, or Linux™ servers. In this example, you can see a connection to an iSeries server where IBM OS/400® objects are expanded. Using 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 in the lower left of the screen shows information about the selected object (which, in this example, is RENTCAR). Conveniently, some of the object's property sheet information is directly editable in this window.

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 in this example. Beyond what IBM supplies, user-defined actions are possible. The editor is rich in function, far exceeding SEU (source entry utility), and retains features such as entering the letter D in the prefix area to delete a line. The editor also supports syntax-checking and cursor-sensitive F1 language help. It features a built-in program verifier for RPG, COBOL, and DDS that performs 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, (as shown here at numbers RNF7503 and RNF7030). The same feedback is used when the member is remotely compiled.

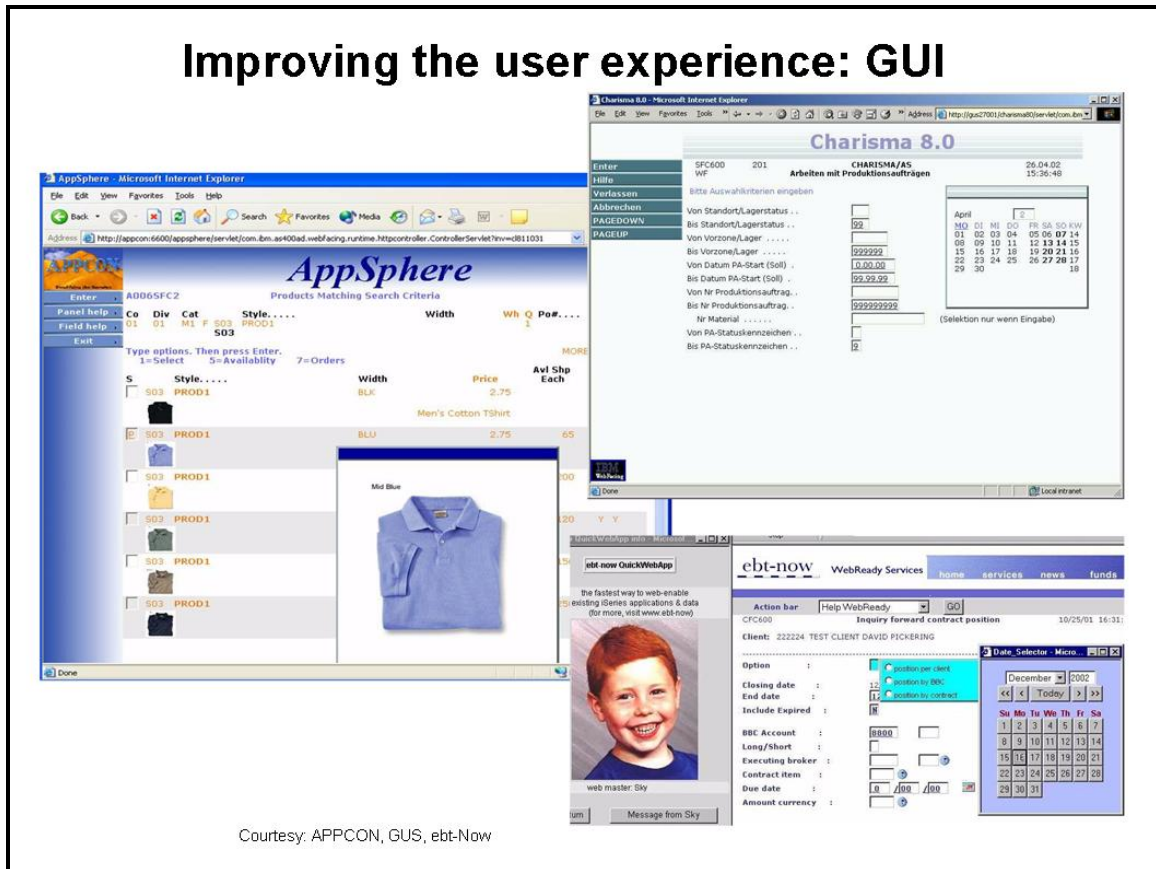
In the top right of the screen is the outline view, which offers an at-a-glance hierarchical picture of the source member currently active in the editor. This is convenient for program understanding and navigational purposes.

The editor supports content-assist for RPG, which allows you to press **ctrl + spacebar** to acquire a list of available options for the current cursor position. For example, in the opcode area, you get a list of valid opcodes that can be inserted at the present 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 table view that can be sorted. As with PDM, this is where the command line (CL) exists for entering and running OS/400 commands. The iSeries commands log shows all explicitly or implicitly run commands for this session.

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

Improving the user experience: GUI



Courtesy: APPCON, GUS, ebt-Now

Improving the user experience: GUI

Improving the user experience refers to changing the interface from a green-screen to a GUI. This step of the roadmap is noninvasive to the application code underneath, so no changes should be required to accomplish the task of adding the graphical interface. The GUI might be achieved by different means, including screen scraping, redirecting the user-interface data stream to use a different rendering, or hard coding a fat client that communicates with the server-side application. All of these vehicles create a more modern look and feel to the existing application code.

Portal technology begins to make sense for this purpose, too. Once an application has moved beyond the green-screen interface into a graphical one, it is possible to allow users to tailor their Web interfaces using portal products. Portals make possible the personalization of users' browser spaces, allowing each person to subset the browser window to show applications that are relevant only to their tasks within each window.

Improving the user experience: Creating a better interface

■ Create graphical user interface for 5250 applications

- ▶ Browser-based GUI is built.
 - IBM WebFacing Tool
 - WebSphere Host Access Transformation Services (HATS)
 - iSeries Access for Web
 - Portal interface
 - XML
- ▶ Underlying application logic is not affected.
- ▶ Functions can be added to the application using new technologies with no impact to existing programs.

5250	GUI
RPG/COBOL	
HTML/JSP	
Servlets	
Portlets	
XML	
Remote System Explorer	
Portal ToolKit	
Lotus WorkPlace	
IBM WebFacing Tool	
IBM WebSphere HATS	
iSeries Access for Web	
Vendor tools	

Improving the user experience: Creating a better interface

You can use one of three IBM options to create a better user interface (UI) for an existing application: the IBM WebFacing Tool for iSeries, WebSphere Host Access Transformation Services (HATS), and iSeries Access for Web. All three produce a Web user interface from a 5250 UI with no impact to underlying application logic. They produce interfaces that run on WebSphere Application Server - Express (V5 or higher) and on any operating system that can support WebSphere Application Server V5 or higher.

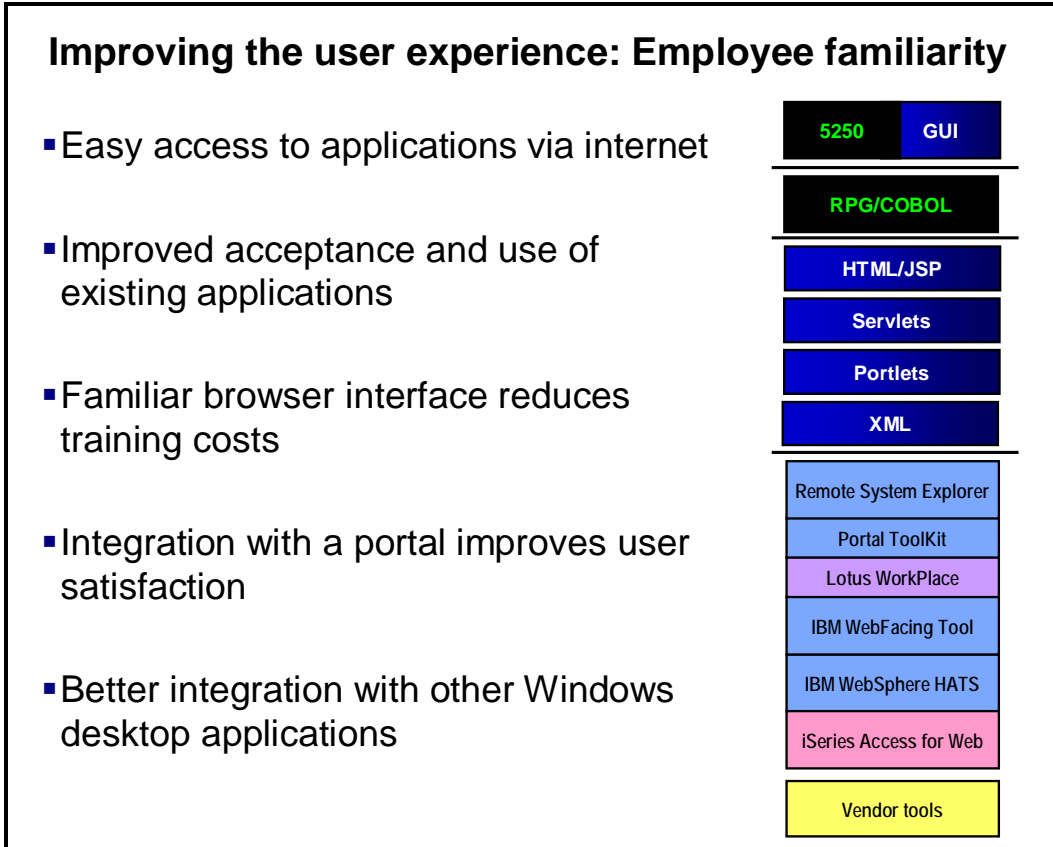
The IBM WebFacing Tool for iSeries converts DSPF DDS source, at development time, into a Web application that uses JSPs. The conversion is refined by the CODE Designer tool to add Web settings (via special comments) into the DDS source, which affects the result of the conversion. For example, fields can be hidden or replaced with HTML tags, where the tag contents are derived from the field contents. This means you can send an IFS image file name to a hidden 5250 field that is converted to an HTML image tag in order 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 data stream. Rather, the OS/400 runtime intercepts the data written by the application to the record formats, sending it to the IBM WebFacing Tool for iSeries servlet that inserts it directly into the generated JSPs™ (JavaServer Pages™). These applications do not use interactive

cycles when run on the iSeries 8x0 servers announced in January 2003. Also, there is no runtime fee for these applications.

HATS, the second refacing option from IBM, is part of Host Integration Solution for iSeries. It converts a 5250 or 3270 data stream, at runtime, to a browser-based interface that runs in the WebSphere Application Server. Because it is a runtime conversion, it instantly transforms screens to run in a Web application. HATS developers can easily refine the conversion results to improve the Web UI, and they can repeat the process whenever necessary. The HATS development environment plugs into WebSphere Development Studio Client for iSeries. The HATS development environment has been shipped with WebSphere Development Studio Client since V5.1.

iSeries Access for Web, at first glance, seems similar to HATS in implementation, as both perform 5250 to HTML conversion at execution time. However, the key strength of iSeries Access for Web is the many additional things that it does. There are several operational capabilities inside the iSeries Access for Web tool that allow a user to browse job and output queues, display message queues, and more. While browsing a spooled file, it is possible to see the output in .pdf format and then e-mail it other users. It is a powerful tool for remote operations.



Improving the user experience: Employee familiarity

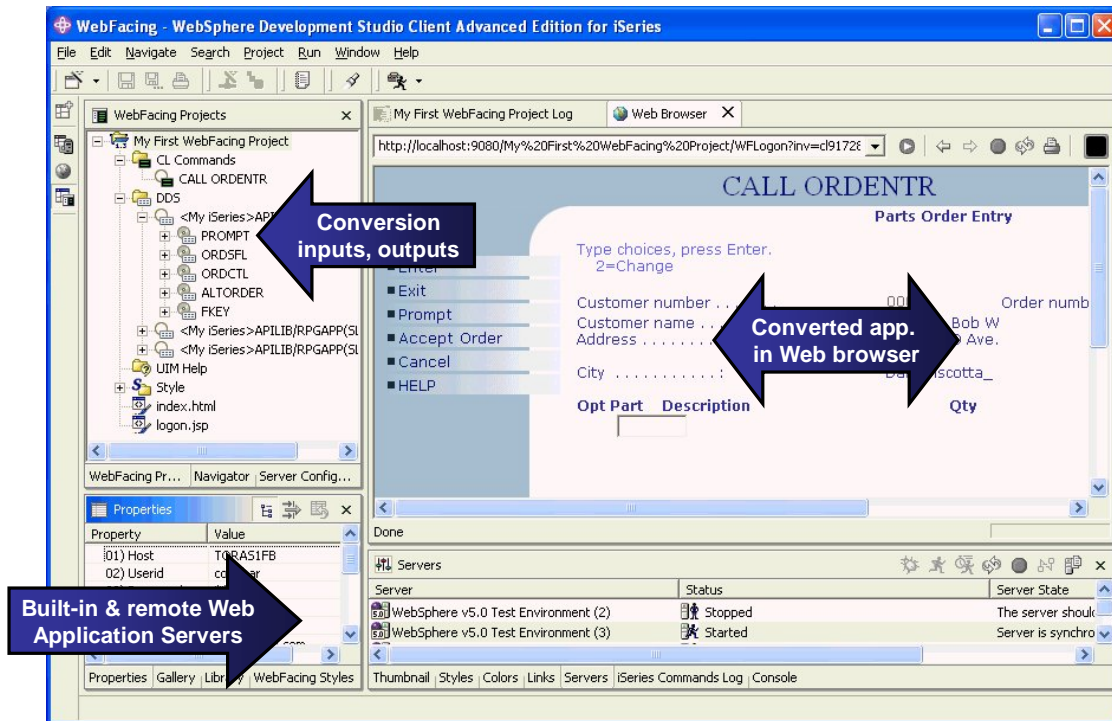
Employees are becoming savvier with using a mouse and clicking between multiple windows. They like the graphic interface that the IBM WebFacing Tool for iSeries, HATS, or iSeries Access can provide without any changes required to the underlying application. The browser is a familiar interface to most users, providing increased employee and customer satisfaction.

In addition, remote access is called for in the modern workplace at an increasing rate. Developers might choose to include remote access as a part of the modernization process, allowing remote employees, vendors, and partners to access the iSeries system. Doing so not only improves productivity, but it also lowers communications costs.

Solution providers might find that they are able to reach new customers who were previously uninterested in a green-screen application solution. These potential customers will discover that they can provide higher customer satisfaction and increased trading partner satisfaction.

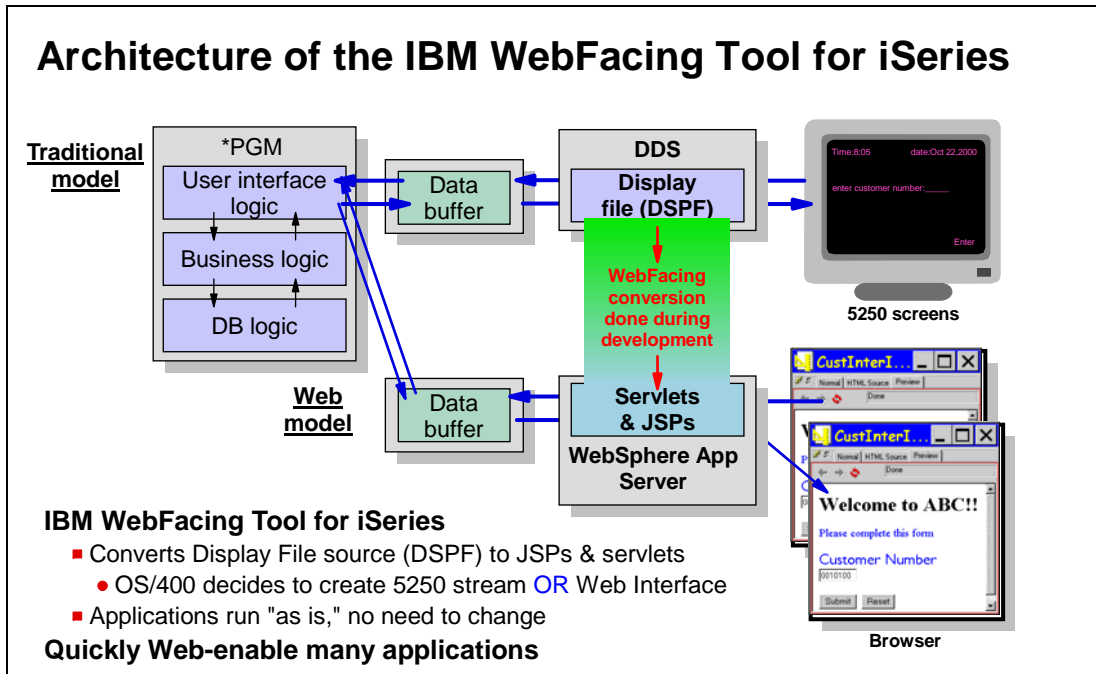
Optionally, including portal support in the user interface solution will extend the familiarity and reach of existing applications into an even more modern environment. All three IBM refacing products that improve the user experience (IBM WebFacing Tool for iSeries, HATS, and iSeries Access) provide portlets for new browser-based UIs to be included in an enterprise portal.

IBM WebFacing Tool for iSeries



IBM WebFacing Tool for iSeries

As stated previously, the IBM WebFacing Tool for iSeries uses perspectives that the developer can flip through using icons on the left of the screen. Developers use a wizard to create a project. They specify which of the DDS and the UIM (user interface member) help panel members are 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 the **Run on Server** option. This executes the preconfigured copy of WebSphere Application Server within the desktop test environment, as shown in the bottom right window on the screen capture.



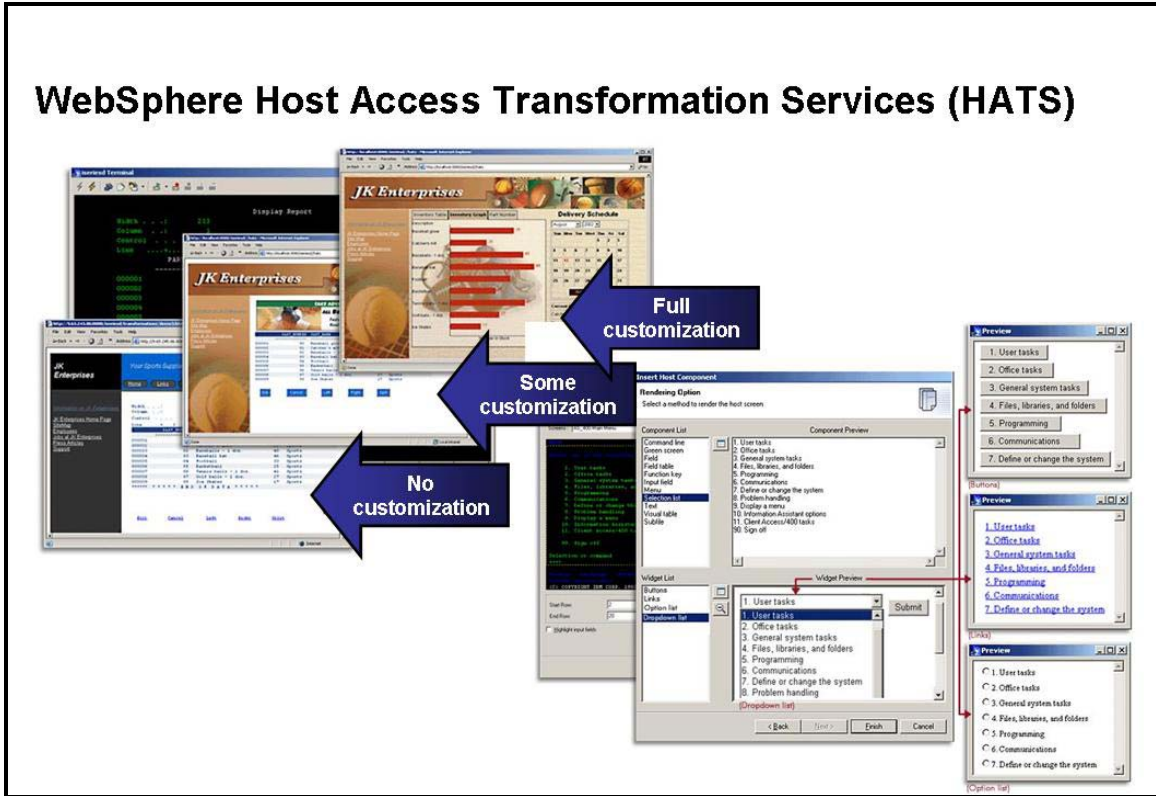
Architecture of the IBM WebFacing Tool for iSeries

Before explaining the architecture of the IBM WebFacing Tool for iSeries, it is best to review the architecture of a traditional green-screen application.

When a program executes a WRITE operation, it passes the contents of its I/O buffer to the operating system, which then merges the buffer with the DSPF object. This produces a 5250 data stream that is sent to the green screen terminal.

Using the IBM WebFacing Tool for iSeries, you can point the tool to the DSPF source code. The converter then reads the source and creates a corresponding JSP for each display file format, which is placed into an application server. When a request comes from a browser to invoke the program, the application server initiates the request to start the program. The unchanged program executes, and when the WRITE operation occurs, the program dumps the contents of the program I/O buffer to the operating system (just as in the green-screen example). However, the operating system interrogates the status of a switch that designates whether the request was initiated from a browser or from a green screen. If the request came from a browser, then the I/O buffer is handed back to the application server and to the servlet. This data is merged with the JSP, and HTML is sent to the browser.

It is possible to have two clerks working side by side running the same application code: one working on a green screen and one working from a browser.

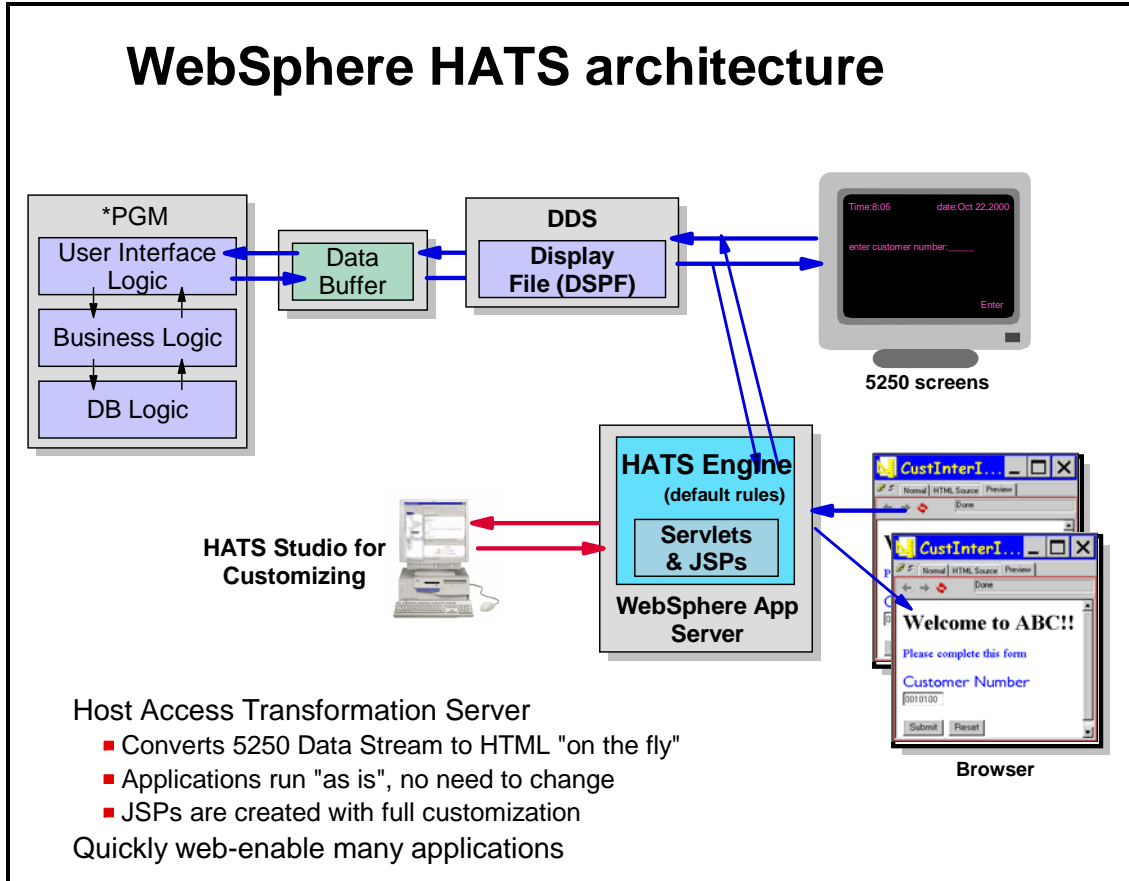


WebSphere Host Access Transformation Services (HATS)

The set of four progressive screen captures on the left side of this screen shows the results obtained from doing various levels of customization using HATS. The original green screen is in the background, with the customization increasing from the bottom left picture to the top right.

As 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.

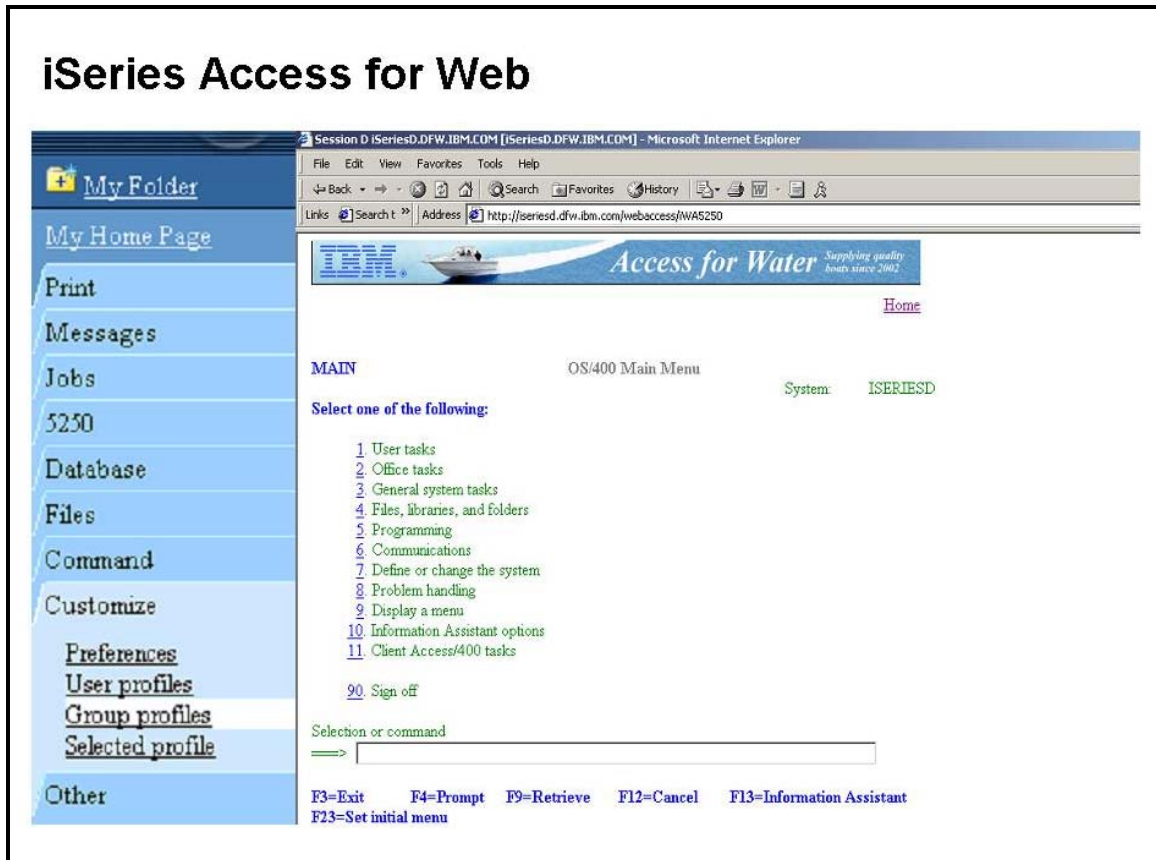
The three screen captures on the right show the Eclipse development environment for HATS (used for the customization of the generated GUIs). Alterations require no HTML skills and are accomplished using intuitive wizards and dialogs. As with the IBM WebFacing Tool for iSeries, HATS leverages the built-in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



WebSphere HATS architecture

Using IBM WebSphere Host Access Transformation Services, you can define a few rules for transformation, including such things as how menus will be handled and function keys. When a program executes a WRITE operation, it passes the contents of its I/O buffer to the operating system, which merges the buffer with the DSPF object. This produces a 5250 data stream. The HATS runtime engine then intercepts the data stream and transforms it according to the rules previously defined, producing HTML that is then sent to the browser.

For more detailed customization, you can use the HATS Studio tool to review the specific screen images, create notebook tabs, and redefine the flow of the application. These detailed customizations are stored as individual JSPs in an application server. When the intercept routine grabs the 5250 data stream, it first looks for a corresponding JSP for detailed customization. If there is no specific JSP for the screen image, the HATS runtime engine applies the default rules. From this point on, HATS operates the same way the IBM WebFacing Tool for iSeries does by determining whether the request was initiated from a browser or from a green screen, then handling the request as necessary.



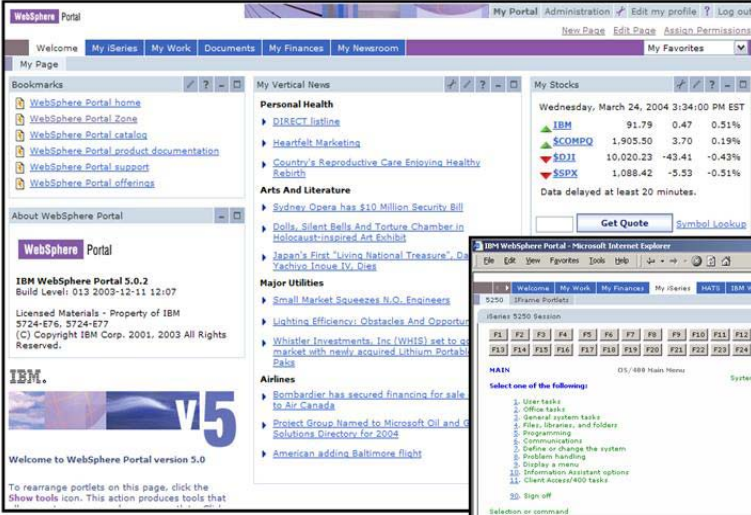
iSeries Access for Web

iSeries Access for Web transforms the 5250 data stream into HTML at execution time.

The folder of available options is visible on the left side of the screen. The options that have been authorized for an individual user are displayed here. iSeries Access for Web offers many more functions than 5250 data stream-to-browser conversion. Some of the other key functions include:

- Displaying and working with messages
- Displaying output queues and spooled files
- Dynamically creating database inquiries via SQL
- Executing OS/400 commands

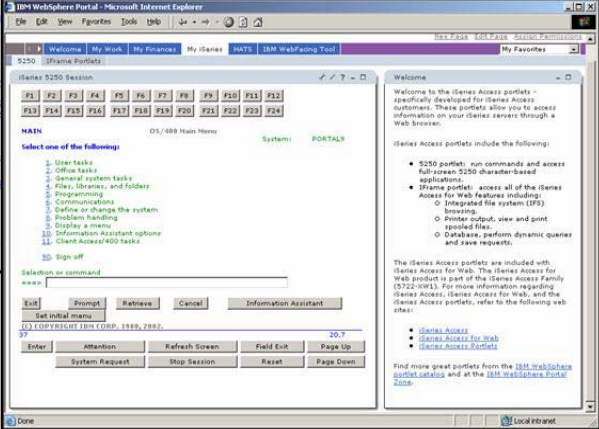
WebSphere Portal



The screenshot shows the WebSphere Portal interface with several portlets. On the left, there are navigation links and a 'WebSphere Portal' section with version information (5.0.2). The main content area is divided into three columns: 'My Vertical News', 'Personal Health', and 'My Stocks'. The 'My Stocks' portlet displays a table of stock prices for IBM, SCMPDQ, \$DJI, and \$SPX as of Wednesday, March 24, 2004.

WebSphere Portal and Portal Express Plus

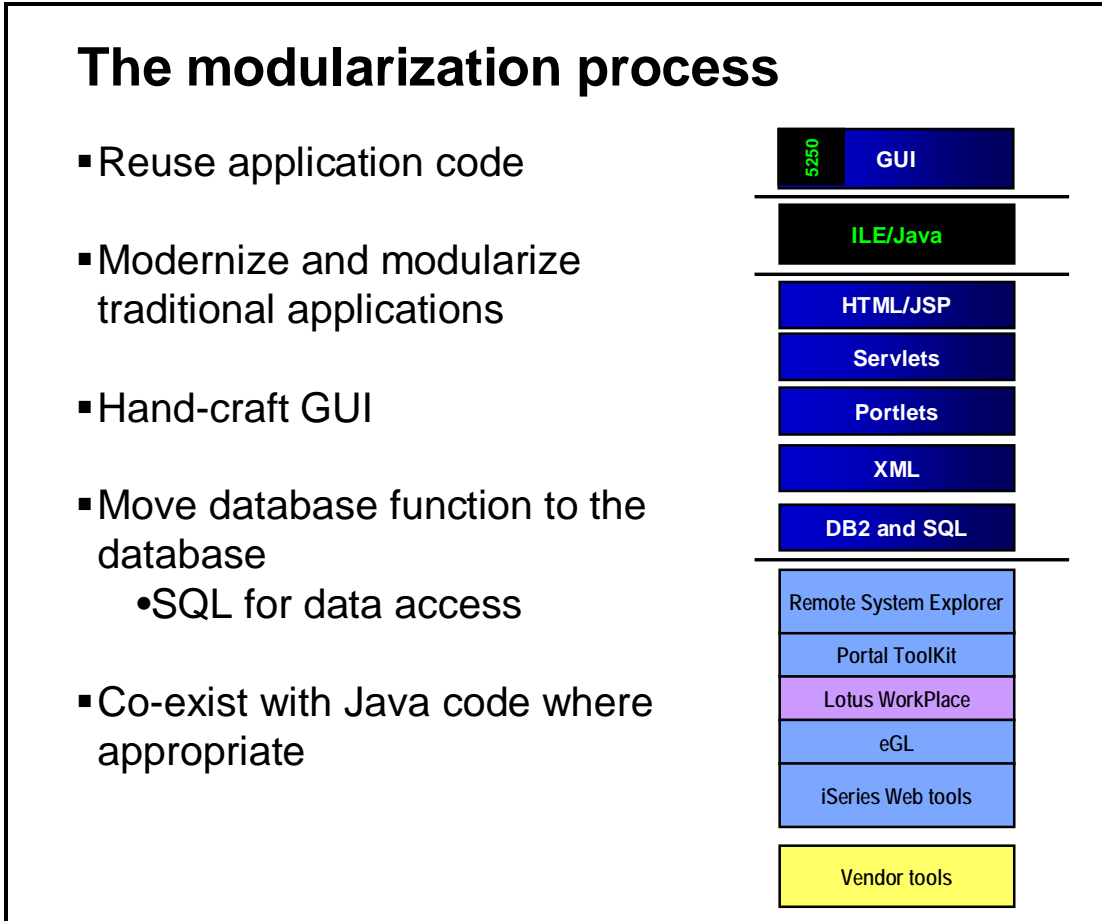
- Subset browser real estate
- Activate multiple applications at once



This screenshot shows the IBM WebSphere Portal administration console. It features a navigation pane on the left with options like 'User tasks', 'Files, libraries, and folders', and 'Programming'. The main area displays a 'Select one of the following' list of tasks, including 'ClientAccess400 tasks'. At the bottom, there are buttons for 'Exit', 'System Request', 'Refresh Screen', 'Field Exit', 'Page Up', 'Stop Session', 'Reset', and 'Page Down'.

WebSphere Portal

There are multiple ways to incorporate WebSphere Portal technology. The most basic function is to use a portal as a vehicle for subsetting the real estate of the browser. This will allow multiple applications to appear on the browser at one time. This is similar to having multiple items from the favorites list active on the browser window at one time, each contained within separate boxes (IBM calls these portlets). In order to use portlets, an application needs to be explicitly enabled to be accessed via portal technology. All three IBM tools create browser interfaces that support portal enablement.



The modularization process

The modularization process is the first roadmap step that involves reworking the application code. In fact, new business logic might have to be developed as well. This is a time to move the application code into the most current form of the compilers: RPG IV and ILE COBOL. It is also a time to identify and separate various functions, paying special attention to those that have historically been hard coded within the application. It is now possible to move these to the database, using structured query language (SQL) for data access. Examples of this are referential integrity and stored-procedure functions.

By re-architecting the application into a modular one, you also allow for the replacement and addition of modern technologies (such as browser interfaces and distributed database activity). As the reworking progresses, it might also become apparent that some of the business-logic modules would be better served by being exposed to trading partners in the form of Web services. This is not feasible with large, multi-functioned, structured programs.

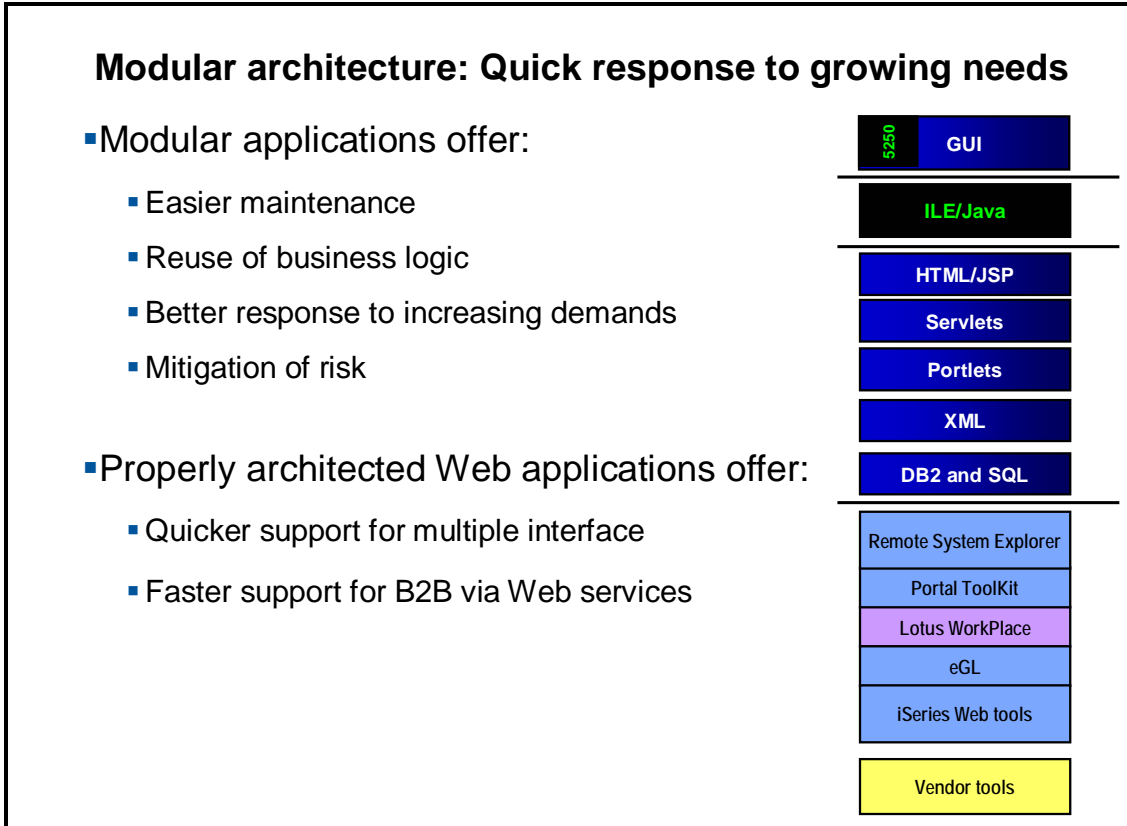
Modernizing and modularizing the application code can still be done to a limited extent by using Remote System Explorer (RSE). As you might remember from the first step on the roadmap, this tool is a component of WebSphere Development Studio Client for iSeries. However, although the editor function is there and is extremely productive, it still requires that you build the code by handcrafting. Building the new browser-based user

interface is done using the Web tool component of the same toolset. Included in the Web tool component are the necessary wizards and editors to help develop, expose, and publish modules of business logic, thereby creating Web services. One important thing to note is that a Web service allows functions (in the form of a remote function call) to be exposed to others who can execute them from another server, without knowing or caring about the language in which the service is written or the operating system on which it runs. Web services can be implemented within a company to allow complete reuse of key functionality, or they 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 parameters of a program or procedure. It also generates the Web application code 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 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, and entry fields support validity checking and even database referencing. The necessary HTML and JavaScript™ is generated automatically.

Another key feature of this now modular application is the ability to incorporate modules written in different languages, such as Java. As the application requirements are identified, it might become obvious that some key pieces of the application should be written in the language appropriate to do the task. In the future, applications will no longer be written in a singular language, but rather the language will be chosen based on the task requirements.



Modular architecture: Quick response to growing needs

Modularizing applications brings with it a tremendous value proposition for the enterprise. Modular applications allow for a much faster response to the changing requirements of the business. As application users request enhancements and extensions to the existing code, the development or maintenance project can be distributed among multiple programmers. It is faster and more reliable to replace only a few lines of a small component of an application, rather than reworking a 2,000-line program. Mitigating risk and providing faster response to business needs should be key components to an ongoing IT strategy.

As new technologies emerge, a modular application can plug in a new user interface or a new request to access data from a remote system much faster.

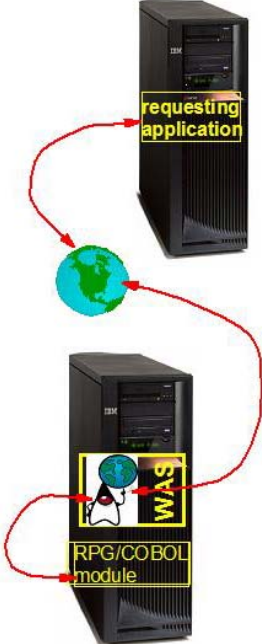
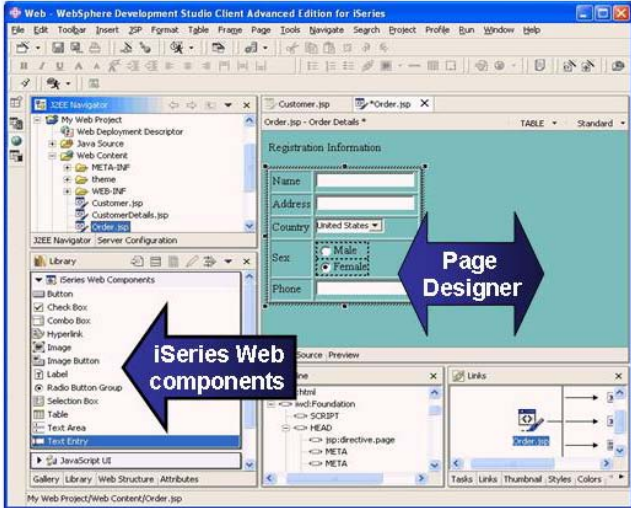
WebSphere Development Studio Client provides many pieces to assist in the modernizing efforts. Web tool wizards assist in the creation of a Web interaction so the programmer does not get wrapped up in the technology, but stays focused on the business function instead.

Many companies have found that exposing some business logic as Web services has improved the relationships they have established with trading partners and customers. Providing a more immediate response with more current information is a primary concern of business relationships in today's marketplace.

Modular architecture: Web-based applications

iSeries Web Tools

- Tools can build graphical front end
- Tools can assist in creation of Web services

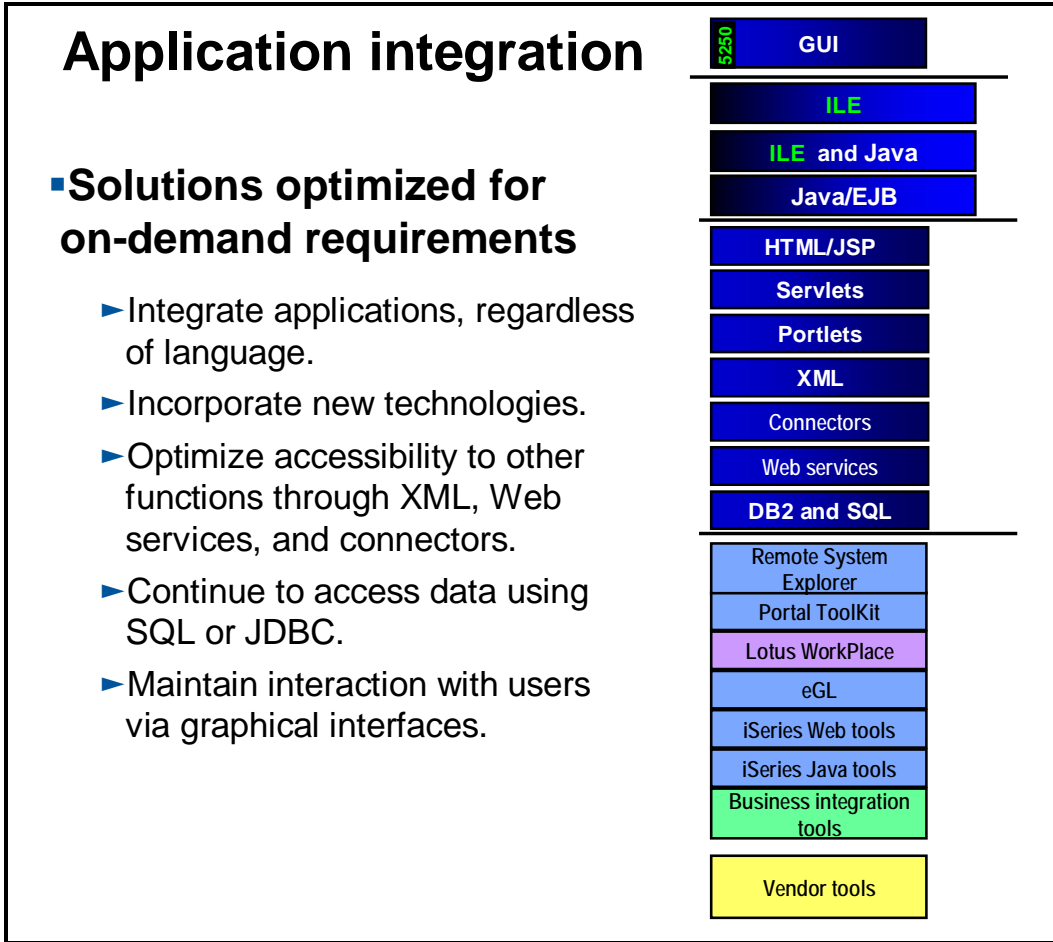


The screenshot shows the WebSphere Development Studio Client Advanced Edition for iSeries. The main window displays a web form titled 'Registration Information' with fields for Name, Address, Country (set to United States), Sex, and Phone. A 'Page Designer' arrow points to the form. On the left, a 'Library' pane shows 'iSeries Web components' including Button, Check Box, Combo Box, Hyperlink, Image, Image Button, Label, Radio Button Group, Selection Box, Table, Text Area, and Text Entry. A 'Server Configuration' pane is also visible. The diagram on the right illustrates a 'requesting application' server connected to an 'RPG/COBOL module' server via a globe, with a WAS icon on the module server.

Modular architecture: Web-based applications

Once an application has been moved from multi-function structured code (monolithic programs) to smaller functional modules, it is possible to use IBM Web Tools (a component of WDSC) to create a Web-based application. The wizards and smartguides will simplify the tying of code modules to Web images, exposing functions to the Web as required.

These series of screen captures illustrate some of the functions provided by the iSeries Web tools. The image on the right represents the function of the Web services wizard that will wrapper modules of RPG or COBOL code with Java, allowing the module to become exposed as a Web service.

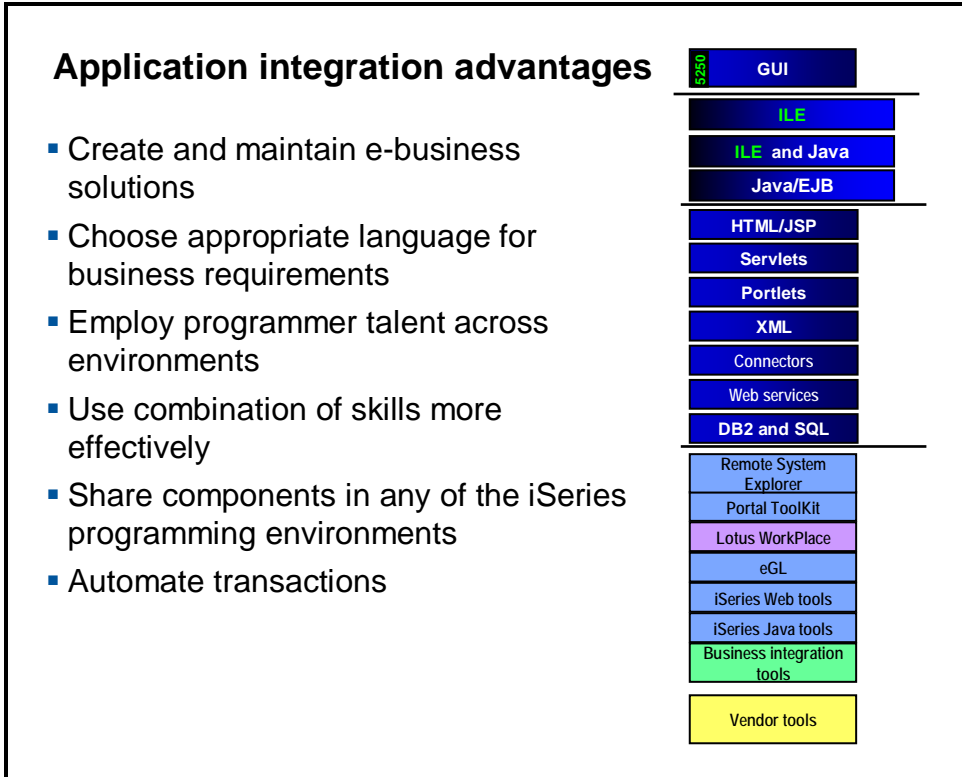


Application integration

This step of the roadmap focuses on the move toward integration of applications within an enterprise. The key technologies to accomplish this are components such as robust Web services, making use of the capabilities of XML and connectors between various application components. This is the first step toward complete business process integration.

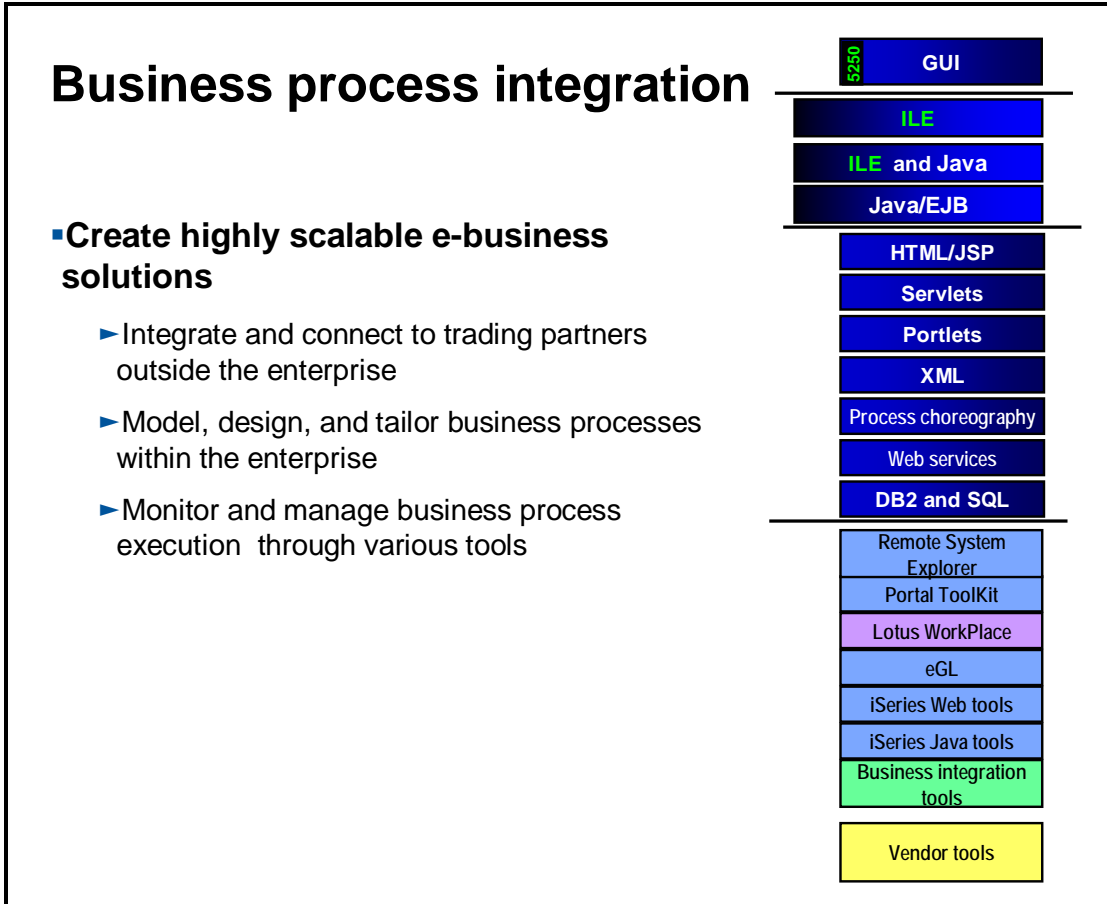
the next step on the roadmap is to externally define the connectors to the actual applications and then code interfaces to these standards. This allows independent applications to talk to each other regardless of language or business function. It also completes the vision of reusability of application code and avoids the replication of business logic in multiple applications and languages. ILE provides a consistent environment for designing applications so that different languages can be used as appropriate. Similarly, data can be accessed through either SQL or Java Database Connectivity (JDBC™) as appropriate.

Using a connector type of architecture provides an enormous benefit in reusability, even for maintenance activities. As components of an application are updated and evolve, a programmer is only required to ensure that the updates continue to conform to the coding standards of the connector interface. There is no need to change additional components.



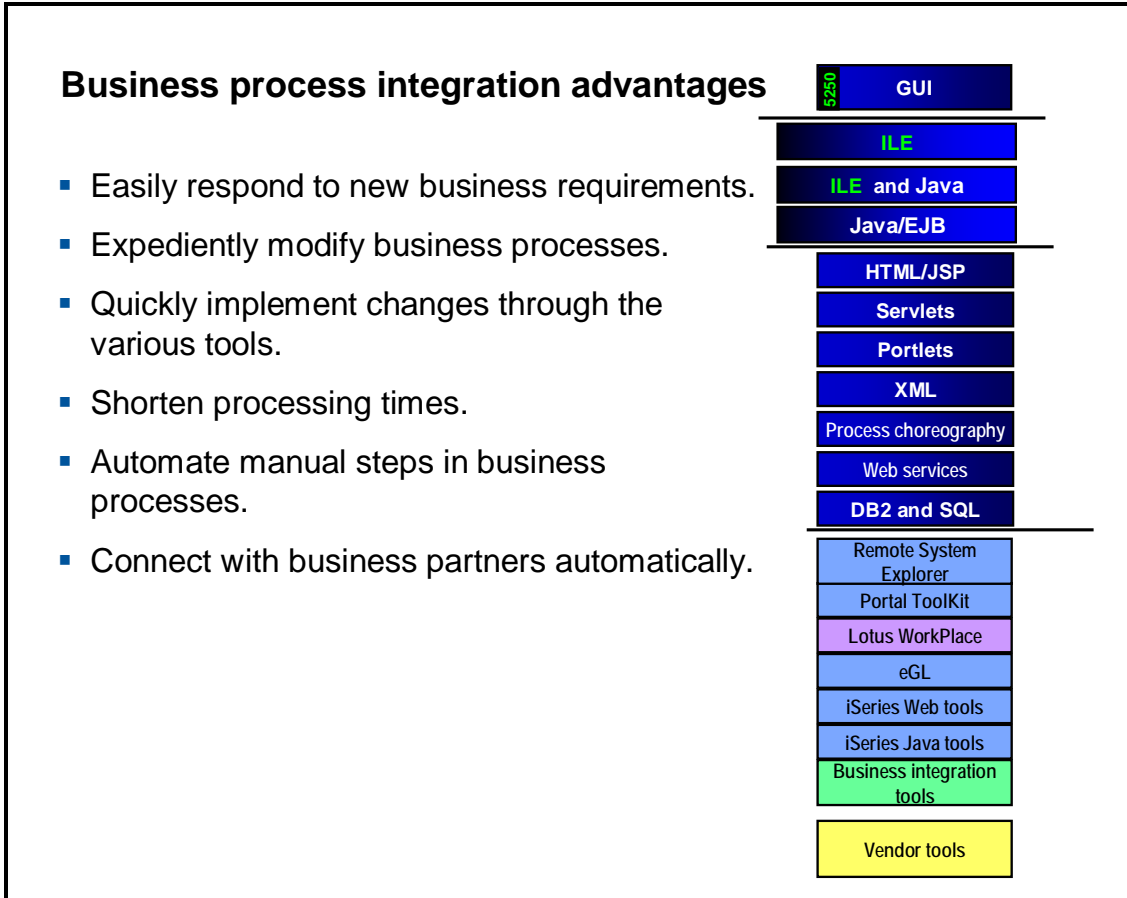
Application integration advantages

There are financial advantages to implementing full application integration. The most obvious are: eliminating code redundancy and the ability to code business functions according to whatever language makes the most sense. Development shops can use whatever skills are necessary to accomplish the requirements of a modern business application. Code needs only to be created and maintained in one place, and yet provides the necessary interfaces to allow other applications to use the function. This also speeds the delivery of the enhancements to all participating applications and reduces the risks of implementing the changes. Today, many companies have applications that are dependent upon each other for the transfer of data. These systems are not always linked together, however. In many cases, manual intervention, such as re-entry of data from one place to another, is required. The automation of this process can significantly improve response times and reduce error in data entry.



Business process integration

The last step on the roadmap focuses on business-to-business (B2B) integration. Trusted trading partners can integrate their code across the network, maintaining key business logic in one location and using technology to monitor and maintain the most efficient links between the application components. As in the previous step, code is modular in nature and is written to a specific integration standard called a connector. Each enterprise can code its application to the standard connector architecture, and the business process integration software can monitor, maintain, and manage the integration across enterprises. Good business-process integration can also help to keep track of information flow by helping to automate and manage the flow of data and knowledge from enterprise-to-enterprise.

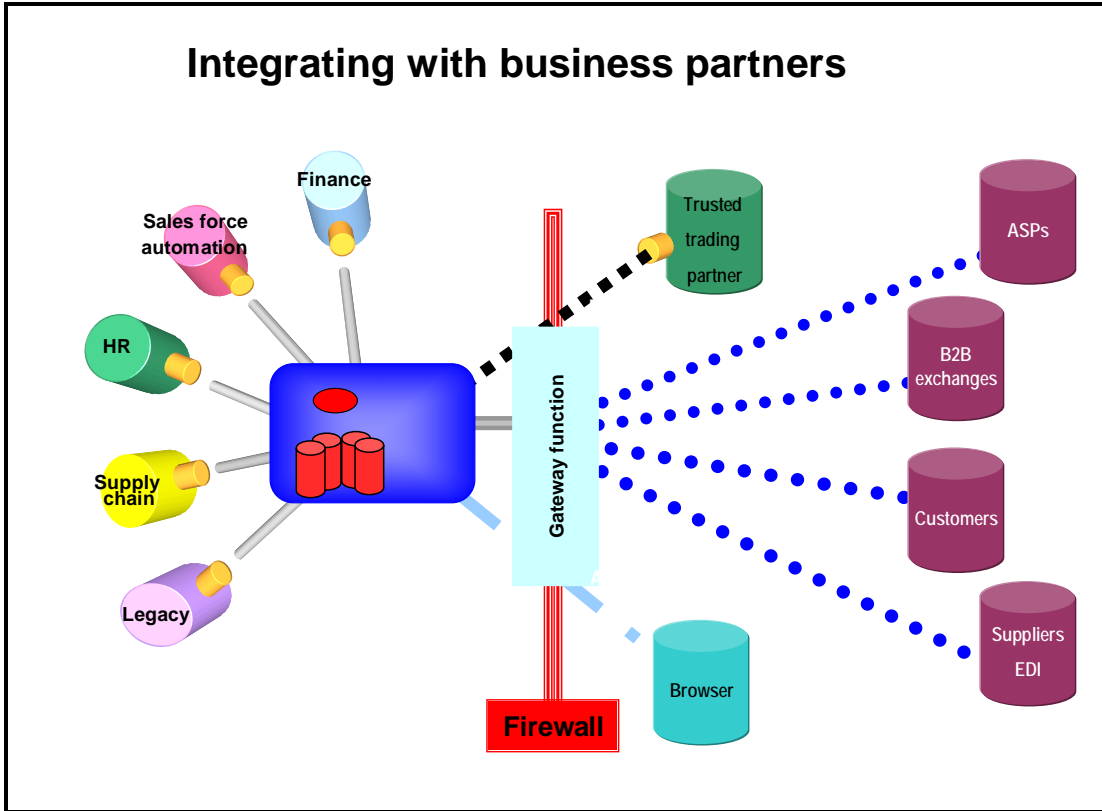


Business process integration advantages

There is great value to be gained in having this level of integration between enterprises. Business processes can be modeled, enhanced, and exploited to ensure maximum return to the partners involved. New processes can be quickly added, allowing multiple enterprises to see immediate benefit. Changes to existing processes can be quickly implemented, in response to business requirements.

Business integration is a key element in shortening the processing time inherent in doing business across multiple enterprises. Even the basics of connecting and staying connected can be automated.

One of the key advantages that tools such as WebSphere Business Integration can offer is the concept of modeling. This lets analysts review the entire business process, including manual steps and tasks that are not working properly. Models can be built even if some of the function will ultimately run on a remote site or a trusted partner site. The model can then be updated as processes are improved, and in some cases, the update can be done automatically, eliminating manual intervention.



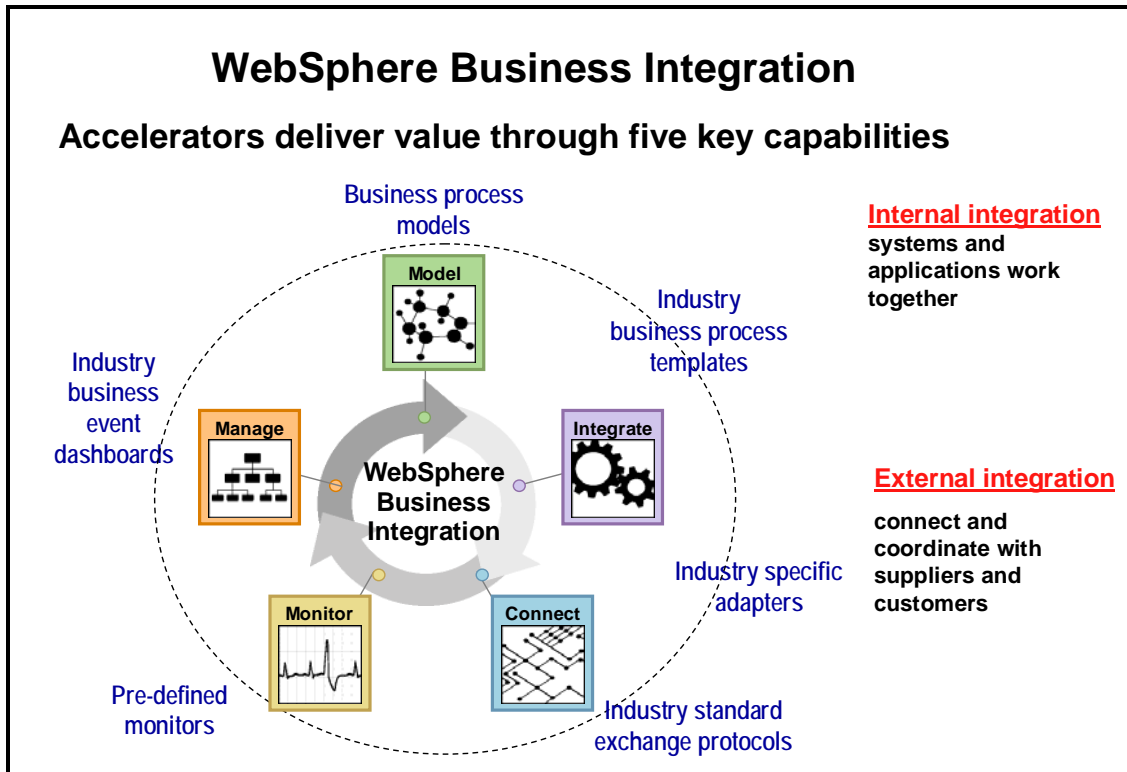
Integrating with business partners

The gateway function is a mechanism to receive or send data across the network to other entities. The WebSphere Business Integration Solution provides the mechanisms for many different gateway implementations.

One such gateway is WebSphere Business Integration Connect, which allows information in many different data formats (XML, EDI, text, and others) to enter a system through many different protocols (FTP, POP, or HTTP, for example). WebSphere Business Integration Connect can take this information and run a business solution to automate the process.

Trusted trading partners can communicate over a firewall with out-of-the-box adapters. Another gateway is Access Framework, which allows users on browsers or client software to access back office applications.

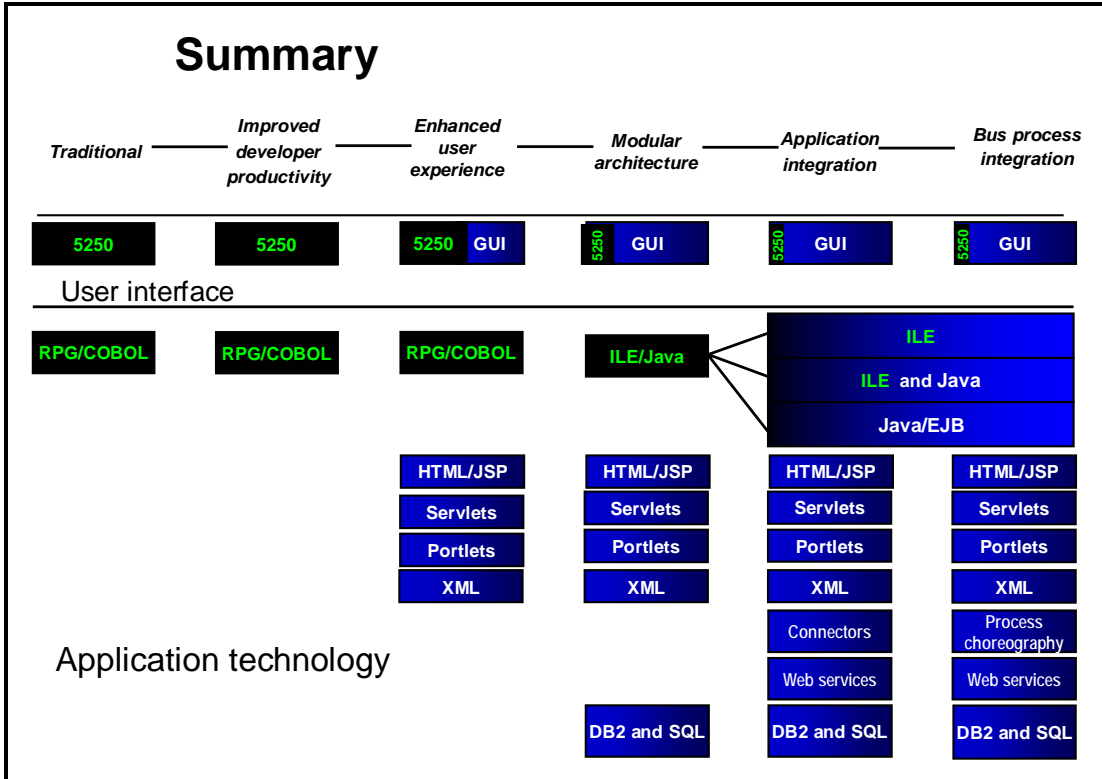
It should be noted that some of the components, such as maps and adapters, can be used in an interface with an outside connection. This component reuse is a major advantage of the hub and spoke architecture.



WebSphere Business Integration

Internal integration helps a company's systems and applications work together to achieve efficiencies. This, in turn, allows them to provide better customer service.

In addition, external integration supports companies in connecting and coordinating with their suppliers and customers in order to achieve efficiency and improved communication.



Summary

The major goal of the iSeries Developer Roadmap is to provide a suggested path for modernization, including everything from user interfaces to business-to-business (B2B) connectivity. It is important to keep in mind that modernization efforts can have three main points of view.

First, applications need to be modernized; perhaps each in a different way. Secondly, the method should be dependent upon the requirements of the business. Developers have to update their skills by utilizing new tools and more sophisticated (and more productive) technologies. Finally, development shops need to update their total development environment to encompass advanced technologies and tools. Enterprise modernization pertains to all of these points of view.

The goal of the roadmap is therefore to recommend an approach for each of these areas. Every step of the roadmap might prove to be a destination, a resting point, or an interim step as applications, programmers, and shops modernize.

Additional information

- The IBM eServer Tools Network
ibm.com/servers/enable/tools
- Follow a roadmap
ibm.com/server/enable/site/roadmap/ques/index.html?isvg
Request an iSeries solution technical assessment
- iSeries Developer Roadmap
ibm.com/series/roadmap
- Access hardware: Virtual loaner program
ibm.com/servers/enable/site/vlp
- IBM Innovation Centers for Business Partners
www.developer.ibm.com/iic
- PartnerWorld
ibm.com/partnerworld
- Virtual Innovation Center for Hardware
www.developer.ibm.com/vic/hardware
- IBM Independent software vendors
ibm.com/isv

Trademarks and special notices

© IBM Corporation 1994-2005. 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: IBM, the IBM logo, eServer, iSeries, WebSphere, AS/400, OS/400, and ibm.com

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.

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

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