

WebSphere Studio for iSeries Case Study

by
Dave Slater
Worldwide iSeries AD Market Manager

Jim Mason
President ebt-now

ebt-now was engaged by a utility company to architect, design and build a customer self-service e-business billing inquiry application. The application would allow customers to register as web users to securely view their own bills and payment status online. Future plans by the utility company include online processing of credit card payments for bills using an existing iSeries credit card payment application written in RPG. IBM's WebSphere Studio for iSeries allowed ebt-now to rapidly develop, test and deploy this application.

Company Background

ebt-now, is an iSeries service provider delivering Java WebSphere Quickstart services for iSeries-based companies: training, engineering, development, migration and implementation of Java web applications.

Application Concepts

The web-enabled billing inquiry application would allow customers to:

- register as web users for the application
- register to access their own accounts
- login as a web user
- select a registered account
- search, browse and view bills and payment status for the selected account

The self service billing inquiry application was to provide customers with the following benefits:

- easy, online access to current bills and payment status
- elimination of the need to maintain paper files on old bills
- see changes in usage over time to better control costs

ebt-now was responsible for the following:

- creating an application deployment architecture
- building a project plan
- creating an application design to meet requirements
- building the application
- testing the application
- providing customized mentoring and skills transfer
- assisting the IT staff with deployment of the application to WebSphere

WebSphere Studio for iSeries Case Study

Project challenges / requirements

Challenges we faced on the web application project included:

1. The Web application had to be intuitive; easy to use, simple to learn
Customers would be expected to come to the web site, enroll and then access their bills and payment information online, without needing support.
2. The Web application had to be easy to maintain and deploy .The IT staff at the company provided first class IT services and applications on the iSeries server but were new to: running web applications, WebSphere, Web application development and Java.
3. WebSphere Application Server installation, configuration and tuning would be complex .The web application would have to access a corporate network through a firewall and the company was part of a larger corporate network with remote DNS servers.
4. Reuse existing iSeries database and RPG applications where possible. The web application accessed existing production databases heavily. The only new tables added were for web user and account registration. The next phase with online payments will reuse existing RPG applications to handle online credit payments. I tested RPG calls with the Web Interaction wizard.
5. Remote development. To speed development and control costs, development was done at ebt-now's web lab. ebt-now required test copies of production data and applications.
6. Tight Schedules. As is normal in application development, the schedules and budgets were very tight forcing ebt-now to choose tools to build Java servlets, JSPs and Java beans quickly.

Selecting the right tools: WebSphere Studio for iSeries and VisualAge for Java for iSeries

Some of the development tool selection criteria for this project included:

- Rapid development and maintenance features
- Easy to use, productive test and deployment features
- Easy to learn, productive for experienced iSeries developers moving to Web development
- Build Java servlets, JSPs and Java beans to access existing data and RPG applications
- An open standards based development toolset that can build portable applications

WebSphere Studio for iSeries Case Study

Web application tools reviewed included:

Tool	Features	Requirements fit
IBM Host on Demand	provides traditional 'screen scrape' for 5250 apps on the web	Not the right tool for delivering custom, simple to use web apps
IBM WebFacing tool	generates Java web applications to front end existing 5250 apps	Not a fit since existing 5250 applications are not used here
IBM WebSphere Studio for iSeries	builds, customizes Java web applications to access iSeries applications and databases.	Excellent for visual development, editing, test, deployment of Java web apps with wizards for generating basic SQL, servlets, JSPs and beans to access iSeries data and programs. Special iSeries deployment wizard.
IBM VisualAge for Java	builds Enterprise Java applications, EJBs, applets and Java beans	Excellent easy to use, visual builder tools for Java beans, EJBs and integrated WebSphere Test Environment for testing web apps. Great support for reusable Java component development.
IBM VisualAge RPG	builds client / server applications using RPG logic, can generate Java applications and applets	Creates a rich Windows GUI not suitable for a Web application
Microsoft Visual Basic	builds client / server and web applications using Basic. Used commonly for building Windows server applications.	Not as productive for visual development, testing and deployment for Java web applications as Studio.
Other Java IDE's	build Enterprise Java applications, primarily for other servers. Requires experienced Java developers.	Lack VisualAge for Java's easy visual builder tools for Java beans and integrated WebSphere Test Environment for testing web apps.
IBM Lotus Domino	build Domino web applications in Domino Designer. Good for Notes document management, workflow apps.	Doesn't handle development of flexible database applications and host RPG program integration well.

WebSphere Studio for iSeries Case Study

Net.Data	older macro library for building iSeries database access for HTML pages	Not as productive as WebSphere Studio. Doesn't build reusable Java beans. Not as easy to learn: requires Net.Data macro language and deeper HTML skills.
RPG CGI programming	traditional web development model.	Not as productive as WebSphere Studio. Doesn't build reusable Java beans. Not as easy to learn. Requires deeper web server, HTML skills. Applications not as fast as Java typically.
Other Web Site editors	build, edit web sites and web pages. Examples include: Netscape, Front Page etc	Good to build basic HTML pages graphically only. Then, import pages and resources into Studio.

Other advantages for using WebSphere Studio and VisualAge for Java included:

- no runtime charges for the tools
- development tools are components of IBM WebSphere Development Studio for iSeries . WebSphere Development Studio for iSeries is a no charge upgrade for current AD customers with software subscription
- builds pure Java server-side, thin-client, web applications using servlets, JSPs and Java beans technology for WebSphere Application Server
- supports most existing web browsers well

Using the tools right for the job

Key features of WebSphere Studio used in this project include:

1. Fast development of database access and update web applications with SQL, database, Java bean wizards that quickly generate Java servlets, JSPs and beans for database access, updates, inserts, deletes and calls to RPG programs via stored procedures.
2. Rapid visual development of HTML, JSP pages with Page Designer's support for HTML and JSP tags with visual edit, direct code edit and page preview views simultaneously available for a page. The Design Time controls make it easy to create input controls like list boxes etc on HTML forms.
3. Easy creation and maintenance of hyperlinks in pages with Page Designer's ability to define visually dynamic hyperlinks with parameters prompted from Java beans and Studio's file relations view to visually edit hyperlinks as needed.

WebSphere Studio for iSeries Case Study

4. Visually building web pages that call RPG programs with the Web Interaction wizard that has a visual editor to generate PCML (Program Call Markup Language) calls to ILE RPG programs that have an input page for parameters and a results page for output.
5. Simple publishing of web applications to any server with Studio's 'smart' management of hyperlinks in HTML pages and JSPs that automatically convert URLs in all files to the target server during publishing and Studio's global find and replace text to map applications to new configurations quickly and the iSeries publishing wizard.
6. Management of all the files in a project with Studio's check in / check out control to manage changes to project files, import of other web pages and resources and the Project Integrity Report to check all the project's files and their relationships (eg identify any broken links easily between pages etc).
7. Quick creation of graphic images, logos and animations for web pages with Studio's Web Art Designer to quickly create and modify graphic images, logos, headers etc for our web site

Key features of VisualAge for Java used in this project include:

- Bean Info page of the Visual Composition editor to visually build custom, reusable Java beans:
 - Jsp_DefaultError** – handles customizable runtime error control logic for any web application error
 - NumericHelper** – provided mod 10 check digit support for data entry of some key numeric fields (IBM's WebFacing tool includes javascript libraries that provide this and more data entry editing)
- IBM WebSphere Test Environment: easily configured test web application and started and stopped the test WebSphere server
- Debugging runtime applications EASILY using both VisualAge's: Debug facility to step through and change executing Java code
JSP execution monitor to see what JSP statements were being executed as Java code concurrently
- VisualAge's export connection to Studio to 'publish' Java bean versions directly to the project in Studio.

WebSphere Studio for iSeries Case Study

Lessons learned on the project

Based on our experience, WebSphere Studio and VisualAge for Java are the right tools to rapidly develop Java web applications and train new developers on Java web development.

Try to make your application easy to use.

Add online help pages that accessible from any page for the user. Provide Customer Service or Support line contact information that is also accessible from any page with hours of operation, phone numbers and email addresses. You would not believe how many “large” corporate sites fail to do this well. Oh wait. You would believe it ...

If a JSP fails to compile at runtime:

Review the error messages displayed in the web browser first. Usually, the error only tells you the page that failed to compile, not why.

Next, check the correct target directory or repository to find the Java generated source code for the servlet that didn't compile as a Java class. Potential reasons for a failed compile include: incorrect Java, HTML or JSP code in the page or a resource that wasn't accessible at compile time (eg a Java class not found on the classpath). If you started with WebSphere Test Environment servlet engine with the option to load servlets externally, the generated Java source code for a JSP is the latest file in a sub directory of the Test Environment (by default): temp/JSP1_0. If you select the option to load servlets externally, the generated Java source is in a Project in VisualAge: JSP compiled code. You can view the source from the VisualAge workbench.

If a runtime error occurs in a JSP:

Review the browser page for any error messages. In some cases you may see a default error page that doesn't include any specific error information. If this is the case, see if the JSP has a generated page directive for a default error page. If it does, comment this out by changing the prefix `<%` to `<% //`. Now, running the JSP, errors will be dumped to the current browser page and not handled by a default page. Since WebSphere Test Environment lacks good logging facilities compared to WebSphere itself (where logging is excellent), consider either putting the generated servlet in debug mode in VisualAge or using a scriptlet such as the one below to write messages to the generated browser page:

```
<% pageContext.getOut().println(" this shows a Java variable = " + myVariable); %>
```

Use JSP model for generating database wizard applications.

For simple applications, you can create a “similar” MVC (Model-View-Controller) concept in a JSP by creating a “before” block of control that executes before you do output in the JSP. You can edit the inputs, do the processing and check for errors before producing output. Most of this work is typically done in Java beans so you don't have large JSPs with thousands of lines of code (a very bad thing !). The reason to choose the JSP model with the resulting tsx database tags is that they are easier to maintain visually in Studio for new developers. This is important when you are

WebSphere Studio for iSeries Case Study

transferring an application to a customer who isn't experienced in maintaining Java applications.

Produce test use cases for any complex application function

You can easily build Java "tester" beans that test complex logic separately from the application as a whole. In our case, the NumericHelper bean that did mod 10 checking on input numeric values was tested in VisualAge using a "tester" bean we created with specific use cases. We were sure the bean worked perfectly before we added it to Studio as part of our application.

Get help to resolve networking problems

For this billing application, ebt-now received good support from the IT network staff to reconfigure how names were resolved using their DNS servers. Originally, our WebSphere Application Server just timed out whenever we sent a request to start the application from the browser. By running a test Java application, we discovered that we could ping the WebSphere server by address instantly. When we attempted to connect using a configured DNS name through the firewall we timed out because the DNS request was forwarded to a corporate DNS server thousands of miles away. The network staff reconfigured the processing sequence for DNS servers to use the local server first and our timeout problems ended. We could have reconfigured the application timeout value in WebSphere, of course, but that would not have solved the real problem.

Remember to remap your JDBC driver when deploying on WebSphere

If you develop and test your application in VisualAge's WebSphere Test Environment you will use the AS/400 Java toolbox JDBC driver which can remotely access the iSeries database. When you deploy to WebSphere on iSeries you can change to use the native JDBC driver that comes with the Java JSDK on the iSeries for better performance. The toolbox driver will work directly on the iSeries but the native driver is faster.

Speed remote development

We have some automated tools we've built that help us rapidly reconstruct and modify customer's databases on our servers. In addition, we can easily check to see if supplied RPG and CL test programs have all the associated application objects we need for testing. This helped us setup our project on our server in an hour or two versus days.

WebSphere Studio for iSeries Case Study

Studio recommendations

Plan how to use tools

Spend time PLANNING HOW to use the tool for your specific project. It helps to have an experienced lead developer or consultant direct how the tools will be used to build project components, test and deploy the application.

Build application prototypes

If your company is new to Java and WebSphere, consider building an application prototype of your first application. It is NOT the application itself BUT a prototype only that would include:

- representative user interfaces and navigation to get some user feedback on the application design
- key technical features to be used (eg using tsx database JSP tags, stored procedures etc) in the real application
- key usage scenarios for how the tools will be used to solve specific needs: build and test JSP page, create application links with parameters to other pages, etc.

Remember, REAL prototypes are just that. Prototypes. Nothing more. They are not the application. They are short in duration. Useful for learning and validating design concepts as well as testing key technical and tool features. One of my customers described the incorrect use of prototypes best when he said: “Here we only deploy our prototypes. We never actually build the real applications.”

Get latest fix packs

Ensure you have the latest PTFs applied. We hit a few errors in the tools because we didn't have the latest fix packs applied. None of our problems were difficult to identify or resolve though.

Use IBM documentation

Leverage Online help and IBM redbooks for application development help and examples. The online help in both Studio and VisualAge is quite good. The real catch here is that you have to read it! Like many people we are prone to hit buttons first and then, when things aren't working, search the online help...

Get help configuring WebSphere

A significant challenge was getting WebSphere Application Server correctly setup and configured for our applications. If you need to, get outside help to get going with WebSphere. IBM has a FREE 'first customer install' service for assisting on iSeries WebSphere. The email address is: rchfica@us.ibm.com or visit the iSeries WebSphere home page below.

WebSphere Studio for iSeries Case Study

Get outside consulting services when needed

For additional training on WebSphere Development Studio for iSeries look to IBM Learning Services or training partners like us focused on the iSeries and WebSphere.

If you have pressing e-business requirements or need to be more productive in your move to e-business, consider getting outside consulting help from IBM or a business partner.

Simplifying user navigation in JSPs with reusable Java beans

With WebSphere Studio and VisualAge for Java, you can build web transaction applications that have the same sophistication as your existing RPG applications. One of Java's significant advantages over other languages is how complex application processing can be simplified by building reusable beans that provide the function to any client application. In the example below, the logic for how to navigate the application in the event of any error is handled by a "customizable" error handling bean (the Jsp_DefaultError class built in VisualAge).

Below is a portion of the JSP user login page in an application frame showing:

- common application navigation options
- the user login entry form
- page specific navigation options for the user

Home	Help	User Login	Search View Bills	User R
Billing Inquiry - User Login				
If you are already registered as a user for an account, Login by entering your user id and password below.				
User id	Password			
<input type="text" value="JEM"/>	<input type="password" value="*****"/>			
<input type="button" value="Login"/>	<input type="button" value="Reset"/>			
For assistance, please call customer service at: 999-999-9999 (hours: Monday - Friday, 8:30 am to 5:00 pm)				
<input type="button" value="Main menu"/>	<input type="button" value="Registration menu"/>	<input type="button" value="Logout user"/>		

The page above includes direct database access to edit the user login information and directs the user to a customized error page below if there is a login error:

WebSphere Studio for iSeries Case Study

Home	Help	User Login	Search View Bills	User Registration
------	------	------------	-------------------	-------------------

Error - user not registered and logged in

Explanation: **You are not logged in with a valid user id.**

Press Continue to return to the main menu and login

You must have a valid, registered user id to access the system.

If you are a Billing System customer,
You can register as a new user and, then,
register for a specific account to access the Online Billing System

If you still have difficulty, please contact our support center.
Support Center is 999-999-9999. Support Center hours are Monday - Friday, 8:30 am -

Page login error occurred on:

The error beans automatically set the URLs for where the user needs to return to in the application when the Continue button is pressed.

Conclusions

WebSphere Studio is an excellent tool for building or customizing Java servlets, JSPs, HTML pages and related resources as part of a web project. Java application components and beans can be built and tested productively in VisualAge for Java. If you need to reuse existing business logic in RPG 5250 applications, the WebFacing tool can generate JSPs, servlets and Java beans that can be further customized in WebSphere Studio.

IBM iSeries server's have state-of-the-art, built in web application server runtime environments. In addition, IBM's WebSphere Development Studio for iSeries is a great tool suite for building and maintaining Java web applications very productively.

WebSphere Studio for iSeries Case Study

Details about the tooling

The development environment was IBM WebShere Development Studio for iSeries, version 5.1. The runtime deployment environment was targeted for WebSphere Application Server – Advanced edition version 3.5.3 on an iSeries 270 server.

WebSphere Studio is a component of IBM's WebSphere Development Studio for iSeries (5722-WDS) which includes all of IBM's iSeries application development tools:

Host Tools

- ILE RPG
- ILE COBOL
- ILE C
- ILE C++
- Application Development ToolSet

Workstation Tools (with unlimited workstation licenses per server)

- WebFacing Tool
- WebSphere Studio for iSeries'
- CODE
- VisualAge for Java for iSeries
- VisualAge RPG

IBM is offering the entire package at NO ADDITIONAL CHARGE to current iSeries AD customers on Software Subscription.

Site	URL
IBM WebSphere Development Studio for iSeries	http://www.ibm.com/software/ad/wds400
IBM iSeries WebSphere home page	http://www.ibm.com/servers/eserver/series/software/websphere/wsappserver/
ebt-now iSeries WebSphere Quickstart services	www.ebt-now.com

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, AS/400, e-business, Integrated Language Environment, iSeries, MQSeries, OS/400, VisualAge, and WebSphere.

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

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