

Arkansas Data Services accelerates Web and SOA development using EGL.

Overview

■ Challenge

ADS sought to expand its development service offerings to include Web applications and solutions based on SOA. The company wanted to continue to leverage the reliability and scalability of its existing applications and deployment platform, as well as the expertise and skills of its development team

■ Solution

ADS used IBM Rational Business Developer software and EGL technology to build business solutions that run on IBM WebSphere® Application Server Express software and the IBM System i platform.

■ Key Benefits

EGL helped reduce development time on the company's first Web-enablement project by an estimated 75 percent. Development time for SOA-specific code was cut by 90 percent. With the ability to modernize legacy business systems and deliver rich Web-based solutions, ADS has solidified its existing customer base and positioned itself to tap new revenue streams.

As an IBM Advanced Business Partner, Arkansas Data Services (ADS) provides medical office software and custom development services using IBM technologies. The company's skilled programmers and system analysts have years of experience developing business systems in RPG for the IBM System i® platform.

To address customer requests for Web-based business solutions, ADS began looking for a cost-effective way to modernize existing applications and expand its custom development offerings to include Web applications and service-oriented architecture (SOA) solutions. "We had wanted to move to Java™-based development for some time, but we also wanted to retain the reliability, security and affordability of the IBM System i platform," explains Rusty Gadberry, co-founder of ADS. "We know the learning curve for Java can be steep, so we were looking for

a solution that would enable our existing team to develop Web applications while reusing as much of our existing code base as possible. We didn't want to have to reinvent the wheel just to provide a Web interface."

After initially considering the IBM WebFacing tool, ADS concluded that it needed a complete rich Web development approach to meet the company's application and user interface (UI) requirements.

ADS adopted Enterprise Generation Language (EGL) technology and IBM Rational® Business Developer software to modernize legacy business applications and develop new Web- and SOA-based applications. "Moving to EGL was a strategic decision for us, and we see EGL as central to our development capabilities going forward," says Gadberry.

Key Components

Software

- Enterprise Generation Language
- IBM DB2
- IBM Rational Business Developer
- IBM WebSphere Application Server Express
- IBM i

Hardware

- IBM System i 520

“In the past I’ve developed code for handling the Simple Object Access Protocol (SOAP) envelope and it’s not easy. With Rational Business Developer it is straightforward, and I finished in about 10 percent of the time that it would have taken to hand code.”

—Dan Darnell, an independent consultant working with ADS

Accessing Web services in an SOA

The first two ADS projects completed using EGL were for a wholesale men’s formalwear business that serves retail rental shops. Dan Darnell, an independent consultant working with ADS, led both projects after getting up to speed using materials from a distance learning course on EGL conducted by IBM. On the first project, Darnell used Rational Business Developer to create the interface code needed to link a back-office system with a third-party Web service. “We thought that interfacing our back-office systems with the hosted Web site would be a significant challenge,” Darnell explains. “Our systems were written in RPG and hosted on IBM System i servers running IBM i and the Web host was using nothing but Microsoft® solutions. Our common language was SOA. We used Rational Business Developer wizards to import the WSDL [Web Services Description Language] document that the host provided to describe their Web service. Rational Business Developer created all the supporting EGL types and library functions for the Web service, which enabled us to focus on the business process—synchronizing the Web site with the back-office data. In the past I’ve developed code for handling the Simple Object Access Protocol (SOAP) envelope and it’s not easy. With Rational Business Developer it is straightforward, and I finished in about 10 percent of the time that it would have taken to hand code.”

ADS created a mechanism that triggered real-time updates to the host whenever the back-office data changed. Darnell notes, “Triggers on our IBM DB2® database tables send requests to [IBM] i5/OS® data queues. Our EGL programs monitor the data queues using the IBM i support in Rational Business Developer and EGL, and send updates to the host when inventory changes.”

During the project, staff from IBM Software Services for Rational helped ADS keep the effort on schedule. “We hit a sticking point because we couldn’t authenticate to the Web host,” Darnell recalls. “The support we received from IBM Rational was exceptional. They addressed the problem and we were able to continue moving forward in a matter of hours. They showed a strong desire to make sure we could accomplish what we were trying to do. That commitment also shows in the development community at EGL Cafe.”

Web-enabling an existing system

The second project was to develop a Web interface that enables retail rental shops to place orders via the Internet, instead of phoning them in to a call center. For this project, the challenge was to develop an easy-to-use Web interface to the sophisticated pricing, inventory management and inventory prediction algorithms that had already been implemented on a back-office system. “If we had to redevelop these elements, we would not have been able to complete the project for an acceptable cost in a reasonable amount of time,” says Darnell. “EGL enabled us to leverage elements of the existing back-office solution. In fact, we did not have to change a single line of code in the existing system in order to use it in our Web-based order-entry application.”

EGL also simplified development of the front end. “I am much more productive working with EGL than with Java and its various frameworks,” says Darnell. “I have found that EGL, with its support for JavaServer Faces technology, reduces the complexity of creating rich and complex Web user interfaces by at least 40 percent.”

Reducing development time by 75 percent

The first and second projects required approximately 100 and 160 hours of development effort respectively, including analysis, design, coding, testing and initial implementation. “Without EGL, we would have needed four to six times longer to complete the second project,” says Darnell. That time savings is due in part to increased productivity and automatic generation of Java code. “The Rational tools are first-rate. Along with the editors and visual designers, the integrated application server test environment really speeds the deploy, test, debug and edit cycles. Also, I don’t look at the Java code; I think and work in EGL. With the EGL debugger I debug at a level that is more meaningful—the level of the application’s business logic.”

Taking the next steps

The first two EGL applications were well-received by ADS’s customer. The formalwear wholesaler is seeing sub-second response times while running on IBM WebSphere Application Server Express software and the IBM System i platform.



“Thanks to the Java performance gains of the 32-bit JVM delivered in i5/OS V5R4 our customer is able to host these applications on an IBM System i 520 server with a 500 CPW [Commercial Processing Workload] rating,” says Darnell. “The first two projects proved to the customer that we could deploy solutions quickly and cost-effectively using EGL,” adds Gadberry.

ADS continues to accelerate development using EGL, and the team recently completed a Web-based payment entry system for a water utility. “If it was our first project with EGL, it would have taken two or three weeks to finish, but with the experience we now have we delivered it in one,” says Gadberry.

ADS is now training more of its developers in EGL before updating its core medical office solutions with the technology. Gadberry concludes, “With EGL, we can now offer Web-based solutions that we never could before—and that represents a new revenue stream for us. In addition, EGL is helping us retain clients, because we can show them that their IBM System i 520 platform is capable of running their Web and application servers. We can modernize their business applications piece-by-piece, deploy Web services, and keep them on a reliable, secure platform. That provides real value to our customers, and it is a big selling point for us.”

For more information

To learn more about EGL, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/awdtools/developer/application

© Copyright IBM Corporation 2008

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
October 2008
All Rights Reserved

IBM, the IBM logo, ibm.com and Rational are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at ibm.com/legal/copytrade.shtml

Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.

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

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

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind, express or implied. In addition, this information is based on IBM’s current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

IBM customers are responsible for ensuring their own compliance with legal requirements. It is the customer’s sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws.

This document illustrates how one organization uses IBM products. Many factors have contributed to the results and benefits described; IBM does not guarantee comparable results elsewhere.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.