



# The Mainstream

An article from the IBM @server zSeries software newsletter

Find the current version of The Mainstream here [ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

## Enterprise Transformation: Three scenarios for choosing retooling over rebuilding

*from The Mainstream, Issue 11 - 2004  
The IBM @server zSeries and S/390 software newsletter*

Corporate earnings are up, and management is finally warming to the idea of upgrading the organization's technology platform—to a point. Far from being provided with a blank check, IT executives are being challenged to upgrade their systems without a costly overhaul. They must maximize their current legacy hardware and applications before investing in new technology.

Of course, there's good reason to do so. Years of knowledge, business logic and business rules are programmed into your mainframe—rules that run virtually every aspect of the business. Trying to recreate these in new applications can be difficult, and even risky.

With that in mind, IT executives are choosing a prudent route: slowly evolving their mainframe systems from legacy to new applications, or integrating new systems with old. It's called enterprise transformation.

IBM Business Integration Reference Architecture is the technical framework for enterprise transformation. By making it possible for software to be delivered as reusable, shareable services, this architecture provides companies the ability to bridge disparate systems spread across the entire enterprise. And because its components are modular, implementation can start small and grow to cover all of a company's integration needs, both internal and external.

Here are three stories illustrating how IBM Business Integration Reference Architecture can be used to leverage and extend existing mainframe software and create new business value—without requiring a complete system overhaul. Although the scenarios are fictitious, the business challenges, project requirements and solutions are composites of real-world situations that are typical of today's mixed workload environments.



[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

### **Making legacy applications integration-ready— The online customer self-service story**

Global Distribution Inc. (GDI) needed to reduce its customer service department expenses. To do so, management decided to build a self-service Web site that would allow its resellers to place and track orders without human intervention.

The company set three key parameters for the project:

- *It had to be implemented within a limited budget.*
- *It had to be secure and reliable.*
- *It had to be flexible enough to be easily modified on an ongoing basis.*

GDI understood the benefits of business integration and knew that its legacy systems—which weren't originally designed for integration—were critical to powering the self-service Web site. If it could make the appropriate legacy systems integration-ready, GDI could create a Web self-service application at minimum cost. To accomplish this, GDI chose the IBM WebSphere® portfolio, which would enable it to combine business processes contained in its IBM CICS® systems with new business logic written in Java™.

#### *Designed to succeed*

To start, GDI created a high-level model of the new business process using WebSphere Business Integration Modeler. This enabled GDI to determine which business components could be extracted from its existing CICS systems and to create a unified picture of the new and existing business functions. The company identified 23 business functions, including those for order entry, inventory, invoicing, credit checks, payments, delivery options and triggering the necessary supply chain activities.

Developers then imported GDI's model into WebSphere Studio Application Developer Integration Edition, using Business Process Execution Language. This tool allowed GDI to refine its model and obtain a more detailed picture showing specific business processes and how they mapped to actual application code. As a result, they were able to identify reusable business functions within existing CICS applications and apply them to the new model.



[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

Since the majority of business logic necessary for the new application already existed on the company's mainframe, no changes were required to COBOL source code in GDI's legacy applications. For the remainder, the company wrote new business function, as well as a Web interface. For example, while its legacy system included no mainframe business logic to display images of products, the marketing department had maintained this information on a DB2® database. New logic was written in Java so that the images could be incorporated into the self-service Web site.

To manage the flow of data, WebSphere Business Integration Server Foundation provided the process choreography and CICS Transaction Gateway provided the bridging between the new Java logic running on WebSphere Application Server and the existing CICS applications.

### *Simplified processes delivered at minimal cost*

With the new system, a process that used to involve a customer service representative navigating through several CICS applications is now vastly simplified: A customer logs onto the Web site and is immediately presented with all the options necessary to complete a transaction—even if third-party business partners are involved. If a new customer makes a purchase request, for example, Web services—utilizing open standards—allow for the integration of a third-party credit check verification process.

Created at the fraction of what it would have cost to build a system from scratch, GDI's new self-service application went "live" in just three months. Because the new system relied on applications that had been running well for years, it was highly reliable from the beginning. And because the legacy applications have been made integration ready, extracted business components can be flexibly recombined as the company's needs change.

GDI's customers enjoy the ability to place orders 24 hours a day, seven days a week. The company reduced its budget without affecting customer satisfaction. It's a win-win—especially since new capabilities can be created by leveraging the value of an existing legacy system.



[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

### **Transforming a user interface improves productivity—**

#### **The call center story**

The large call center at Global Products Inc. (GPI) was missing its performance goals. Over budget for three straight quarters, it was also a source of serious customer dissatisfaction. Callers were frustrated by long wait times and by the length of calls. One survey respondent complained about the “long, uncomfortable pauses that occur each time the service representative needs to find some information about a customer’s account.”

At the time, GPI was using a legacy “green screen” call center mainframe application that contained 62 primary screens and hundreds of secondary screens—contributing to an average call time of about seven minutes. Much of that time was spent by the service representative searching through a myriad of screens to find answers to customer questions.

The reps were often as frustrated as the customers. With a learning curve of about three months just to become familiar with the system, many simply left the company before becoming proficient.

#### *Creating an interface to reuse existing applications*

At first, GPI looked into replacing its problematic call center application with a new Customer Relationship Management (CRM) application—until the projected cost was tallied. It would have put the center even further over budget. Plus, the existing legacy application had taken years to build and evolve, and completely replicating it would be difficult. It became clear that GPI needed to reuse and improve its current application.

The team decided to revamp the application’s user interface so that critical information for most calls would fit neatly onto four well-designed screens. They used IBM Host Access Transformation Services (HATS), which is a plug-in for WebSphere Studio, to design the improved application model and begin creating the new interface.



[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

The team started by mapping out the core functions of the application and isolating the redundant and cumbersome processes responsible for the call center's performance issues. With the HATS screen customization wizard, developers were able to create the four primary screens and establish global variables based upon key fields within the host application, eliminating unnecessary screens. By extracting required fields from the original legacy application and building composite screens to display only those fields, GPI was able to leave the underlying business logic untouched.

For the hundreds of secondary screens, developers defined a set of default rules. For example, a rule stated that all selection lists should be represented as hot links. These rules eliminated the need to customize individual screens and significantly reduced development time.

### *Improved workflow restores customer satisfaction*

In just eight weeks, the team developed, tested and delivered an intuitive interface that vastly streamlined service representatives' tasks to access, modify and interpret information. Improvements included:

- *Eliminating the need to enter the same data multiple times; global variables condense into a single view many inefficient, multi-screen sub queries—such as product detail—and populate the relevant fields automatically on all successive requests*
- *Providing simple pull down menus to replace fields that had required service representatives to key in complex alpha numeric codes*
- *Organizing frequently accessed information in tabbed folders, enabling representatives to locate data and answer questions without having to navigate through numerous screens*
- *Replacing all date input fields with calendars*
- *Adding the ability to view customer orders in real-time by providing access to the company's supply chain application, which had previously only been used by the distribution center*

Another improvement was the replacement of text entry fields with valid value lists—that is, drop-down or pop-up lists of the valid entries allowable in any text entry field. By no longer requiring representatives to enter these complex codes, the entire process was accelerated, and many common syntax and key entry errors were avoided.



[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

In addition to streamlining the workflow of the application, GPI improved the entire look and feel of the interface. Though no changes were made to the application source code, service representatives are now able to navigate modern, “point-and-click” Web pages.

Because the company chose a solution that involved no source code modification, the original “green screen” interface continued to be accessible, assuring zero downtime. The finished project was deployed to WebSphere Application Server using a simple three-step wizard. Once deployed, the site was immediately accessible to call center employees from their Web browsers.

GPI’s new system can be learned in just two weeks by the average customer service representative, compared to the previous learning curve of three months. Average call time has been reduced from seven minutes to three—an improvement of over 130%. Employees are happier, turnover is lower and the call center is exceeding its performance goals. Most importantly, GPI’s customers are once again satisfied with the service they receive.

After seeing the benefits of a streamlined call center application, the company decided to move other applications to the Web as well. The next rollout will include WebSphere Portal, which is tightly integrated with HATS and will ultimately provide a common interface for everyone in the company.

### **Integrating COBOL and Java teams: The new applications development story**

Consolidated Financial Products (CFP) was trailing its competition in both revenue growth and time-to-market. Competitors had been growing rapidly and taking advantage of the resulting economies of scale and scope to quickly develop new products and expand their customer base.

As part of its response to the competitive environment, CFP began readying new products—including a Web-based loan product. To ensure its success and that of future new offerings, CFP needed to revisit its IT strategy to better utilize existing resources.





[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

CFP selected the IBM WebSphere portfolio to enable its COBOL and Java development teams to work more collaboratively and productively to meet its goals for growth, starting with its new Web-based loan product.

To begin, developers first used IBM WebSphere Studio Asset Analyzer to look at the company's existing CICS applications and determine which functions could be re-used as is and which would need to be optimized or created from scratch. With the resulting blueprint of the company's existing systems, developers had a graphical view of key business functions that accelerated the task of re-architecting existing business logic to deliver the new product.

Next, the team used WebSphere development tools to switch their "green screens" to new application interfaces. COBOL developers used WebSphere Studio Enterprise Developer to optimize the required CICS processes, processes that would bear the load of thousands of simultaneous users. "Peeling away" the 3270 presentation layer enabled them to expose the underlying business logic for direct access by other applications.

They also used one of the program's wizards to create a connector to integrate the CICS systems with WebSphere Application Server for z/OS™. An additional connection was made to an outside business partner using the SOAP feature that is a part of CICS, while WebSphere Business Integration Adapters provided similar connectivity to CFP's own third party applications.

Meanwhile, the Java team was working on integrating the back-end components into a single, unified mixed-workload application. Using WebSphere Studio, they created the new loan product's Web interface and the Java logic to assemble all of the loan product information.

WebSphere Business Integration Message Broker enabled the teams to model the various message structures and formats so that Web services could then be transformed and routed to all the various systems.





## The Mainstream

9

[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

The application was deployed to WebSphere Application Server and tested to ensure it would perform under real-world conditions. With WebSphere Studio Workload Simulator, the teams simulated thousands of virtual users and the associated traffic they would create. The teams also used WebSphere Studio Application Monitor to perform real-time monitoring of their systems to identify sources of bottlenecks. The resulting information was used to audit the development process itself and make iterative improvements.

### *Ready in months, new offering sets stage for competitive advances*

CFP went live with its high-performance, reliable, scalable product in a matter of months. As a result, the company did more than merely catch up with its competitors. It bypassed them, sending a clear message to its Board of Directors, marketplace and the analyst community: CFP was again a leader in rapid development of new and innovative products and services.

© Copyright IBM Corporation 2004

All Rights Reserved

CICS, DB2, IBM, IMS, the On Demand logo, WebSphere and z/OS are trademarks or registered trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Java is a trademark of Sun Microsystems, Inc. in the United States and other countries.

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

**ON DEMAND BUSINESS™**