SOA solutions for System z
White paper

Unleash the power of mainframe assets into SOA.

By Michelle A. Cordes, Rational System z Ecosystem Team, mcordes@us.ibm.com; and Daniel Moul, Rational Product Line Manager, dmoul@us.ibm.com

September 2007

Contents

Executive overview

Organizations that run the IBM System z™ platform have unique business requirements. Their commitment to the high-availability server platform demonstrates their need for an environment that can support high-volume transaction processing with demanding batch windows and large, critical application portfolios. Organizations with System z environments typically have large numbers of developers with broad skill sets in various areas of business and technology, as well as a diverse set of applications, processes and standards. They also have the highest expectations for the quality of service and value they want their systems to deliver internally and externally to their customers. As a result, organizations face significant challenges, while standing to gain significant benefits, by making the transition to a service oriented architecture (SOA). This white paper examines these challenges and how System z clients can take advantage of the assets they already have.

Introduction: What you probably already know about the reality of today’s development environment

Here are some facts about your mainframe applications and environment that you have probably heard repeatedly over the last few years.

- *There are billions of lines of mainframe code running today’s businesses.*
- *Billions of lines of code are being added every year to mainframe applications.*
- *Some of these mainframe applications have been maintained over a period of decades, not just years.*
- *Over time, your organization’s significant investments have increased the value of your mainframe applications, which are critical to the success of your business. The applications are at the heart of your organization and enable you to manage and process the majority of your customer, product, supply-chain and critical business data.*
- *Some of these applications contain thousands, if not millions, of lines of code. Most of them are likely a mix of 3270 screen processing, as well as data I/O and the core of your application – your business rules.*

- *The developers who maintain your applications know them and your business inside and out. However, their numbers are decreasing. And often, your developers are spread across multiple locations, multiple cities and, in some cases, multiple countries.*
- *Budgets are constrained by increased costs, as well as cuts. Although hardware and software costs might be decreasing, labor costs are increasing even faster. And the business is demanding more and more from IT groups, so it's often up to the development groups to deliver more-agile applications. These applications must also be delivered under the constraints of increased governmental regulations.*

The challenges of complexity

Traditional mainframe applications are large, complex solutions targeted at addressing a specific business function. Design techniques, tools and processes are optimized to develop these kinds of systems. However, today's business systems increasingly involve composite applications, which are sets of related and integrated services that support a business process across existing application silos.

Although your established applications – along with the information and functions they contain – are invaluable to your business, they might be keeping your organization from realizing its true potential. Your complex, highly integrated portfolio of existing applications and processes might inhibit your ability to increase efficiency, differentiation and responsiveness. And faster time to market and improved agility can be difficult to achieve when you're running the business with large, heritage applications that support compartmentalized business processes and hide duplicate functions and information. This complexity increases application-maintenance expenses, diverting your valuable resources away from strategic business initiatives and innovation. In addition, the resulting long development cycles can lead to the perception that IT is not responsive to changing business requirements.

Don’t rewrite or replace your enterprise applications—repurpose them for today’s SOA environment

To meet the demand for business innovation and agility, organizations are looking to improve the structure, reusability, flexibility, quality and maintainability of existing applications. Unfortunately, traditional methods for doing so, such as rewriting existing applications, are expensive, time-consuming and risky. Replacing existing applications with package solutions also can be problematic, because you risk losing intellectual capital developed over the years that’s buried deep in your systems.

Repurposing existing mainframe applications instead of replacing or rewriting them is a time-tested approach to competitive advantage in your business. When you repurpose or modernize applications, as shown in Figure 1, you modify existing mainframe applications so that they can be used as part of a Web-service architecture, such as an SOA. After these applications are modernized, they offer greater agility, flexibility, robustness and productivity to respond to marketplace dynamics and changing business needs. Additionally, by streamlining application and operational processes, and lowering maintenance costs, you can free up more of your resources to focus on developing new business requirements and capabilities. But this repurposing or modernization should be done in an evolutionary manner, not a revolutionary one, so that risks are minimized.

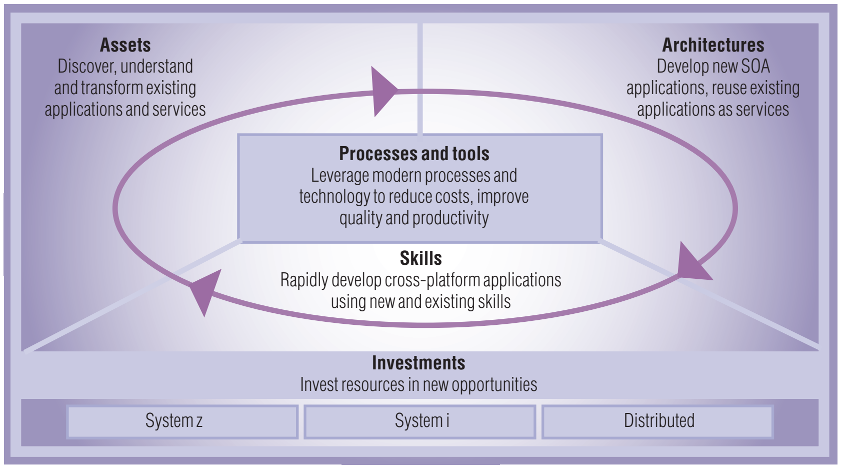

Assets
Discover, understand and transform existing applications and services
Architectures
Develop new SOA applications, reuse existing applications as services
Processes and tools
Leverage modern processes and technology to reduce costs, improve quality and productivity
Skills
Rapidly develop cross-platform applications using new and existing skills
Investments
Invest resources in new opportunities
System z
System i
Distributed

Figure 1. The IBM enterprise modernization solutions delivered through the IBM Rational® Software Delivery Platform help organizations modernize existing enterprise assets, architectures, skills, processes and tools.

IBM enterprise modernization solutions

IBM enterprise modernization solutions (**ibm.com**/software/info/developer/solutions/em) are designed to address today's critical enterprise modernization issues of maintenance costs, leveraging existing assets, architectural complexity, siloed teams, cross-platform support and skills. With decades of leadership in enterprise modernization, IBM is well positioned to provide the capabilities that your organization requires to cost-effectively and incrementally evolve your enterprise systems toward modern architectures and technologies. IBM enterprise modernization solutions can help your organization adapt business processes quickly and flexibly by reusing existing applications and data. Developers can unite and leverage their skills in disparate programming languages and work within a single, integrated development environment. This forms the foundation for your enterprise modernization initiatives. IBM provides offerings in the following enterprise modernization categories:

- Asset modernization – *IBM solutions provide analysts, developers, architects and operations teams with detailed reports and visualizations that enable a rich understanding of their existing applications, regardless of whether the applications were developed for the mainframe or the distributed world.*
- Architecture modernization – *Service-based integration is the most-common approach to exposing the business functions buried in existing systems. You can tap into the business value in your current systems by using IBM offerings to craft modular components from existing code written in COBOL, PL/I or Java™, and by assembling and deploying services as part of an SOA. This can also position your IT for rapid future changes to the business model.*
- Skills modernization – *Organizations are struggling to determine how to tap into the decades of experience and domain knowledge of traditional IT professionals. IBM solutions can help them modernize current enterprise applications to take advantage of newer architectures and technologies.*

- Process and tools modernization – *Organizations have traditionally managed mainframe development separately from other development. Each development silo or organization maintained its own set of development processes and tools. To help bridge the gap between traditional mainframe and distributed development teams, IBM provides an effective change-and-release-management solution. It automates and enforces development processes, and enhances collaboration and productivity across distributed and mainframe operating platforms at every stage of the application life cycle.*
- Development-investment modernization – *Organizations can find it difficult to attract and retain development talent, because many IT professionals are unfamiliar with modern tools. To stay competitive, they need to make the transition to modern architectures and technologies without incurring the substantial risk and high costs of manually migrating their applications.*

In the next few pages, you will read about IBM offerings that fall into these enterprise modernization categories. This will not be an all-encompassing list of offerings, but it will cover those that can help you take advantage of and extract value from your existing assets.

How do you leverage your existing assets in an SOA?

Briefly, let's run through the steps required to create a service-oriented, composite application, as shown in Figure 2. (Note: This is not a complete description, but one used for representative purposes.)

1) Identify a business need. *In our example, we'll say that we are the world's largest special-order rare-book company, and we find that our competition is encroaching into our market. We decide that we need to increase sales without adding to our phone staff. We determine that the best way to do this is to make our ordering system available over the Web.*
2) Decompose the business need into required functions and services. *We will need to create a Web site and provide security for logging on, inquiry access to our mainframe-based rare-book inventory system and access to individual customer information, which is also mainframe-based. We'll also need to provide an ordering form that can interface to our mainframe-based ordering system.*
3) Confirm the planned composite application for online customer ordering. *For this step, we verify that the requirements that are identified meet the business need.*

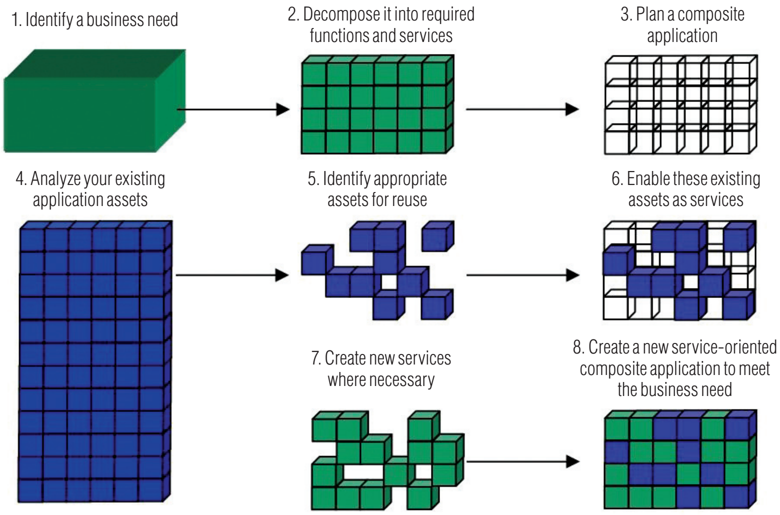

1. Identify a business need
2. Decompose it into required functions and services
3. Plan a composite application
4. Analyze your existing application assets
5. Identify appropriate assets for reuse
6. Enable these existing assets as services
7. Create new services where necessary
8. Create a new service-oriented composite application to meet the business need

Figure 2. Creating a new service-oriented application by combining parts of existing applications with newly created services

4) Analyze our existing applications. *From our analysis of the business need, we know that our mainframe-based inventory, customer records and ordering systems will be involved in this new application.*

5) Identify the actual programs and data items that are part of the existing systems that we might also be able to reuse in this new application. *We find that some of the programs that we want to use in our new application include 3270 terminal maps and error handling, as well as the data access and business code.*

6) Enable our existing code as services. *This could be done in a number of ways. We could use a tool to basically turn our existing 3270 screens into a Web-browser interface. Or, because we are an IBM CICS® shop, we could wrap the existing transaction and make it into a Web service. The third option would be to extract the code from the existing application and make it a stand-alone program that can then be used by both the existing 3270-based application as well as the new Web-based application. We use a combination in this particular case.*

7) Create some new services. *We do need to do some coding, in particular, for security and to provide several of the new interfaces.*
8) Deliver our new service-oriented ordering system for our customers.

The rest of this paper will focus on Step 4 through Step 7: analyzing our existing applications, identifying which assets to reuse, enabling those assets as services and creating new services. Later in this paper, we will also describe IBM offerings that can aid in these steps. All of these steps enable us to unleash the power of our existing assets into the SOA world, as well as deploy our complete application.

Steps 4 and 5

In order to analyze our existing applications and identify those assets that we can reuse as services, we must create an application-metadata repository to provide a foundation for application insight. This will help us understand our application assets and their dependencies, and it will allow us to have current documentation that speeds knowledge transfer. The centralized repository will help us find sets of programs and related assets that embody specific business tasks. A centralized knowledgebase will also allow project leaders to gain visibility into code quality as well as complexity issues. It will help the project leaders to better size projects by determining the effect of a change before making the change and estimating resource requirements based on current application metrics. All of this knowledge will help us plan for the modernization of our online ordering system and continued maintenance efforts. We will be able to identify our business rules so that we can more quickly service-enable our various assets. The IBM product offerings that aided us in these efforts are:

- *IBM WebSphere® Studio Asset Analyzer*
- *IBM Asset Transformation Workbench*
- *IBM CICS Interdependency Analyzer*

These products and the products mentioned in Steps 6 and 7 are described in greater detail later in this paper.

From the analysis that we performed in Steps 4 and 5, we've identified the following items as being part of our modernization effort:

Application system	Inventory	Customer	Ordering
Programs	20	10	18
CICS transactions	8	2	6
Basic mapping support (BMS) maps	12	3	9
Virtual storage access method (VSAM) files	3	0	4
IBM DB2® tables	9	5	10
Method of modernization	IBM WebSphere Host Access Transformation Services	Service-flow modeler in IBM Rational Developer for System z	Restructure with Asset Transformation Workbench and Rational Developer for System z
New items required	Security logon page	Customer inquiry page	Order form

Table 1. Hypothetical results from performing steps 4 and 5.

Step 6

You can enable existing code as services in one of several ways. Each of the programs that we identified as required for our new Web-based system will have to be evaluated to determine the method best used to accomplish this task. Because of the complexity of our inventory system, we'll use one of the quickest ways to enable an existing program as a Web service. That is, to use WebSphere Host Access Transformation Server (HATS) to transform existing 3270 screens into a Web-browser interface.

Our customer-data system is a CICS application. As a user of CICS Transformation Server, Version 3, we have several options, but in our case, we're able to create adapters to integrate our existing customer applications into our new Web applications. We'll use a combination of the CICS Service Flow Feature of CICS Transaction Server and the Service Flow Modeler in Rational Developer for System z.

In the case of our ordering system, it makes better sense for us to restructure the application. That is, we extract the code from the existing application and make it a stand-alone component that can then be used by both the existing 3270-based application, as well as the new Web-based application. We can then refactor our ordering system using IBM Asset Transformation Workbench, IBM Rational Developer for System z and IBM Enterprise COBOL.

Step 7

We can create new services and integrate them with the repurposed assets from the previous steps using IBM Rational Developer for System z, IBM Rational Business Developer Extension or both. We'll use IBM Rational Business Developer Extension to create the new Web pages we need and to integrate them with our Web services.

Unleash the power of your existing assets

IBM provides a number of solutions that can help you extract greater value from your existing assets and repurpose them for Web services.

WebSphere Studio Asset Analyzer

WebSphere Studio Asset Analyzer scans mainframe and distributed software assets, storing related application information in an IBM DB2 repository that resides on the mainframe. You do not have to download your IBM System z application inventory to a workstation, and you don't have to upload your distributed application inventory to a mainframe. WebSphere Studio Asset Analyzer scans source where it lives, whether that's in partitioned data set (PDS) or partitioned data set extended (PDSE) data sets on the mainframe, directories on Microsoft® Windows® or IBM AIX®, or one of a number of software-configuration management systems (SCMs). In addition to source code, WebSphere Studio Asset Analyzer scans mainframe CICS and IBM IMS™ subsystems, the DB2 catalog and distributed IBM WebSphere Application Server domains. The results are then accessible using a Web browser by developers, project leaders, managers, testing personnel and DB2 administrators (see Figure 3). With WebSphere Studio Asset Analyzer, you can find mainframe-application relationships in seconds that might otherwise take much more time.

Enterprises might have hundreds of project teams making changes to application code and the structure of shared data at the same time. These changes occur across hundreds of thousands of programs and batch jobs and touch thousands of DB2 tables, columns, and VSAM files. To avoid system outages caused by these changes, a project team must:

- *Identify affected source within its application.*
- *Identify other affected project teams and existing applications.*
- *Deploy these modifications into quality-assurance and production environments in a highly synchronized manner.*

For our project team, it's not trivial to identify all the potential impacts of a change in our own application. In our simple example, we identified 48 programs, 7 VSAM files and 24 DB2 tables that would be affected by our changes. This is just the tip of the iceberg. We didn't even consider any other application systems that might be affected by our new online system. There could be hundreds of data elements, more programs, and other factors that need to be addressed. This requires a detailed impact-dependency analysis across all of our programs, copybooks, map sets and so on. This might include tens of thousands of files and millions of lines of code.

WebSphere Studio Asset Analyzer helps analysts, developers, architects and operations teams cut through the complexity of their applications and interdependencies to work faster and with better quality. WebSphere Studio Asset Analyzer helps you:

- *Understand multiple virtual storage (MVS™) and Java Enterprise Edition (Java EE) application assets and their relationships through interactive textual and graphical reports.*
- *Identify the artifacts that a proposed code or operational change affects, and make changes with more confidence, greater speed and less risk.*
- *Identify information about connectors for Web-enabled applications.*
- *Identify business processes and flows.*
- *Identify application dependencies at multiple levels, including:*
  - *Jobs and transactions.*
  - *Files and databases.*
  - *Programs and applications:*
    - Written in COBOL, PL/I, Assembler, Java EE and various distributed text formats, including HTML and XML.
    - For CICS, IMS, WebSphere Application Server, IBM WebSphere MQ,and DB2.
    - On IBM z/OS®, IBM System p™, AIX, and Windows operating systems.
  - *Composite applications that span mainframe and distributed systems.*

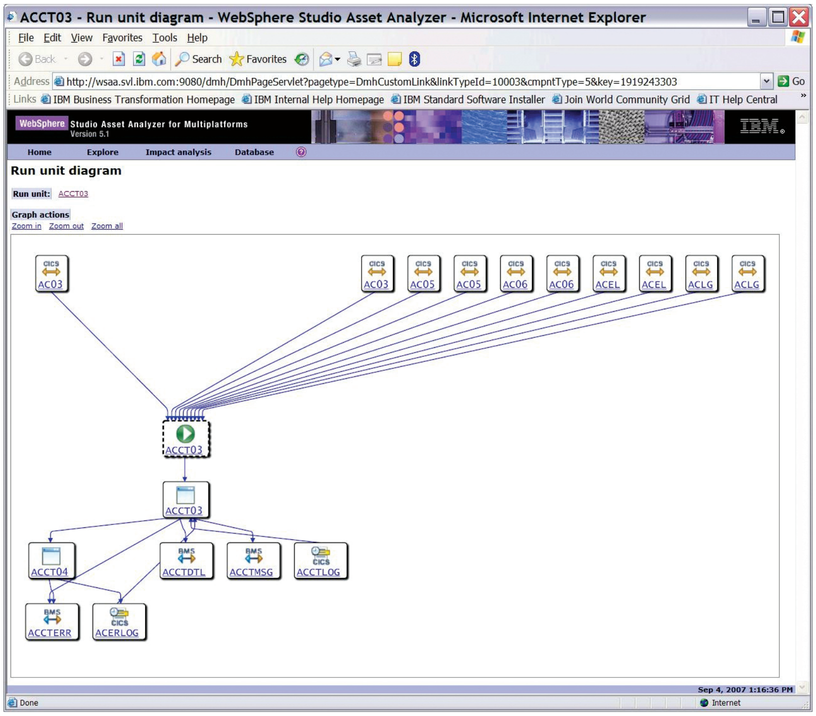

ACCT03 - Run unit diagram - WebSphere Studio Asset Analyzer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://wsaa.svl.ibm.com:9080/dmh/DmhPageServlet?pagetype=DmhCustomLink&linkTypeId=10003&cmpntType=5&key=1919243303
Go
Links IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer Join World Community Grid IT Help Central
WebSphere Studio Asset Analyzer for Multiplatforms Version 5.1
IBM
Home Explore Impact analysis Database
Run unit diagram
Run unit: ACCT03
Graph actions
Zoom in Zoom out Zoom all
CICS AC03
CICS AC03
CICS AC05
CICS AC05
CICS AC06
CICS AC06
CICS ACEL
CICS ACEL
CICS ACLG
CICS ACLG
ACCT03
ACCT03
ACCT04
BMS ACCTDTL
BMS ACCTMSG
CICS ACCTLOG
BMS ACCTERR
CICS ACERLOG
Sep 4, 2007 1:16:36 PM
Done
Internet

Figure 3. WebSphere Studio Asset Analyzer presents relationships between assets in an easy-to-understand, visual manner.

Asset Transformation Workbench

The foundation of Asset Transformation Workbench is a knowledgebase that contains rich, current insight into application portfolios. Detailed reports, metrics, documentation and visualizations of the enterprise applications are readily accessible to project leaders and architects using the workbench. Other team members can use the browser-based module of Asset Transformation Workbench to view the reports that are generated. Integration with IBM WebSphere Studio Asset Analyzer allows users to perform high-level analysis in WebSphere Studio Asset Analyzer and then pass the application insight through a software bridge for use in Asset Transformation Workbench.

Asset Transformation Workbench can provide application-development teams with a set of highly integrated workstation-based modules that use the application metadata in the application knowledgebase to help accelerate strategic and tactical initiatives. Many development, analysis and project-management activities are streamlined with Asset Transformation Workbench:

- *Powerful analysis and assessment tools can help accelerate ongoing maintenance and enhancements.*
- *Tools can expose and help manage business rules, which can simplify application reuse for SOA initiatives (through the optional Business Rules Extension).*
- *Rearchitecting tools can help increase the productivity of teams restructuring and componentizing applications (through the optional Application Architect extension).*
- *The Reuse Analyzer for Asset Transformation Workbench technical preview can help you quickly assess an application's suitability for reuse in an SOA.*

Asset Transformation Workbench Analyzer for Eclipse

IBM Asset Transformation Workbench Analyzer for Eclipse provides core application-analysis capabilities for the developer inside the integrated development environment (IDE) of Rational Developer for System z. This perspective includes five integrated and synchronized Eclipse views:

- *Program context (this program and the objects to which it relates)*
- *Impact analysis (this program and other programs and data that make up an application)*
- *Program flow (paragraph-level flow within a program)*
- *Flowchart diagrams (statement level within a COBOL paragraph)*
- *Repository browser*

By generating multiple views of the structure and flow of programs, Asset Transformation Workbench Analyzer for Eclipse helps developers to quickly get up to speed on the nature of their programs. In addition, the impact-analysis view reveals elements that will be affected by a modification to the code.

CICS Interdependency Analyzer

IBM CICS Interdependency Analyzer for z/OS is a runtime tool that automates detection of runtime relationships within your CICS system, records this data in a DB2 database and provides a user-friendly GUI to help you analyze the collected information, build a relationship road map and use this data in your daily operations. Information provided by IBM CICS Interdependency Analyzer can help you to:

- *Use CICS resources more efficiently.*
- *Make an informed decision about the best way to split workload and move applications to more CICS regions.*
- *Improve the speed and reduce the cost of application maintenance.*
- *Reduce time for problem resolution.*
- *Improve CICS systems and application design.*
- *Implement the workload-management capabilities of IBM CICSPlex® System Manager.*

WebSphere Host Access Transformation Services

With IBM WebSphere Host Access Transformation Service (HATS), you can create Web and rich client applications that provide a user-friendly GUI for your character-based 3270 and 5250 host applications, as shown in Figure 4. HATS is designed to:

- *Help improve the workflow and navigation of your host applications, without any access or modification to application source code.*
- *Transform host-screen components in real time.*
- *Add lists, hot links, tables, buttons, valid-value lists, tabbed folders, graphs and other elements, such as logos, graphics and backgrounds.*
- *Create programmed navigation through multiple terminal screens to improve the productivity and usability of your host applications.*
- *Integrate with the Eclipse-based IBM Software Delivery Platform and deploy to WebSphere Application Server, WebSphere Portal and Eclipse Rich Client Platform runtime platforms.*
- *Create SOA assets from logic contained in your terminal applications.*
- *Support Eclipse Rich Client Platform and IBM Lotus® Expeditor Client environments.*
- *Support these operating systems: AIX, HP UX, IBM i5/OS®, Linux®, Sun UNIX®, Windows and z/OS.*

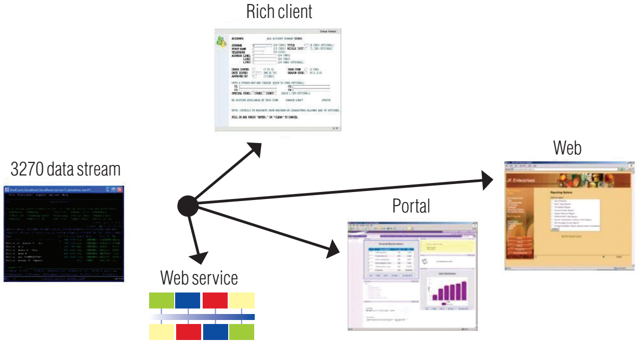


Figure 4. With WebSphere Host Access Tranformation Services, you can extend host applications quickly to the Web, portal or a rich client platform.

Rational Developer for System z

IBM Rational Developer for System z includes capabilities that can help make traditional mainframe development, Web development and integrated SOA-based composite development fast and efficient. Rational Developer for System z supports industry standards that can simplify the development of rich client interfaces, Web-user interfaces, traditional terminal interfaces and back-end, business logic. For business-oriented developers who are new to Java 2, Enterprise Edition (J2EE) and Java, it also includes high-productivity tools that integrate IBM System z processing through Web services with powerful Web-user and session management.

IBM Rational Developer for System z is optimized for IBM WebSphere software and System z deployment environments, and supports multivendor runtime environments. It is powered by the Eclipse open source platform so developers can adapt and extend their development environment to match their needs and increase their productivity. It can help accelerate the development of:

- *SOA, including services deployable to CICS, IMS, WebSphere Application Server and DB2 stored procedure-based environments.*
- *Dynamic Web applications and front ends, including HTML, Java Server Faces (JSFs), Java Server Pages (JSPs), Java and J2EE, including J2EE Connector Architecture (JCA).*
- *COBOL and PL/I applications that can be deployed to IBM z/OS, Microsoft Windows, CICS, IMS, batch, UNIX System Services, and DB2 stored procedure-based environments.*
- *C, C++, and High Level Assembler applications that can be deployed to z/OS, CICS, IMS, batch, UNIX System Services, and DB2 stored procedure-based environments.*
- *Web-service development using new or existing CICS applications, including:*
  – Development of a Web service with a CICS application as the application component.
  – Development of a comprehensive Web service that collects and processes data from terminal and non-terminal applications and Web services.
  – Development of a CICS application that accesses a z/OS DB2 database, and invokes it from a comprehensive Web service.

In the newest release of Rational Developer for System z, several key features were added, including several in support of SOA and Web services:

- *Support for CICS Transaction Server, Version 3.2 Web Service run time was introduced, including support for Web Services Description Language (WSDL) 2.0 and SOAP 1.2 standards.*
- *Tools were introduced for for generating outbound callout codes used by Web services in IMS, Version 10.*
- *PL/I language support in Web-services wizards was introduced for generating Web services from existing PL/I programs for CICS, IMS and batch processing.*

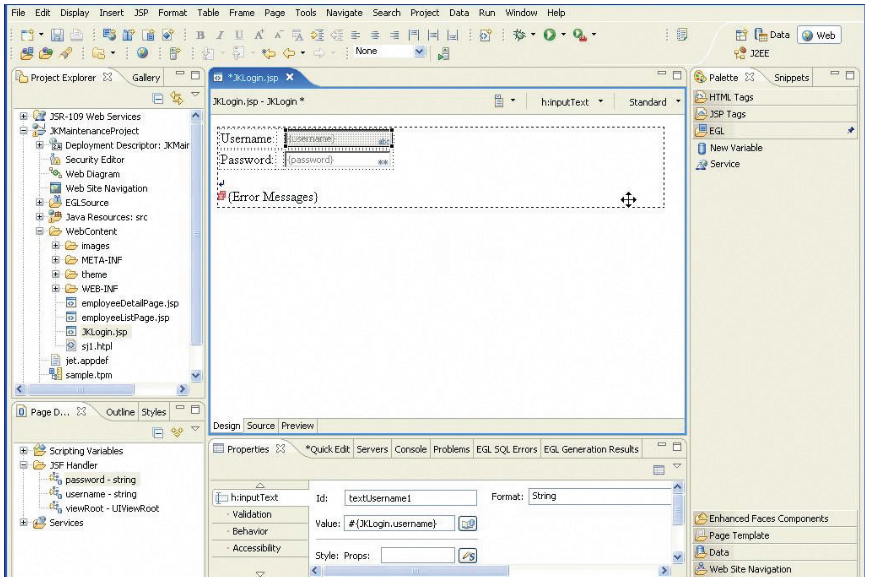
Also introduced were the integration of key IBM problem-determination tools with Rational Developer for System z:

- *File Manager integration enables access to keyed sequence data set (KSDS) files from the IBM Rational Developer for System z workbench, including the ability to browse and update data sets from the workbench.*
- *Fault Analyzer integration that allows users to browse Fault Analyzer abend reports on in CICS, IMS, batch, Java, WebSphere and other environments.*

Rational Business Developer Extension

IBM Rational Business Developer Extension provides a rapid-development workbench for delivering cross-platform Web and service-oriented applications (see Figure 5). Based on Enterprise Generation Language (EGL), the product's innovative end-to-end rapid-development approach provides:

- *Higher development productivity through a powerful, platform-neutral, business-oriented specification and a wealth of rapid-development tools and wizards.*
- *Rapid and simplified SOA support. EGL includes the notion of service. Rational Business Developer Extension delivers the tools for quickly defining, testing and deploying the service to all supported platforms, including automated services generation from models.*
- *Seamless and rapid end-to-end Web application development.*
- *Ability to deploy applications and services to a broad variety of platforms, including J2EE application servers as well as traditional mainframe transactional environments, such as CICS for System z and IBM i5/OS for IBM System i™ technology.*
- *Ease of learning. Typically, developers with general programming skills can learn EGL and become productive with Rational Business Developer Extension in a matter of weeks.*
- *Extension and modernization of existing applications. EGL's built-in interoperability with COBOL, RPG, PL/I or other existing program is excellent for rapid reuse of existing investments as new services or Web systems.*
- *Flexibility and responsiveness. The platform-neutral development approach eliminates skills silos, creating a unified pool of business-oriented developers that can be freely shifted across projects according to business demands.*



Id: textUsername1

Format: String

Value: #{JKLogin.username}

Figure 5. We use Rational Business Developer Extension to easily create our new Web customer logon page.

Enterprise COBOL for z/OS

IBM Enterprise COBOL for z/OS integrates COBOL applications with Web-oriented business processes and simplifies the componentization of COBOL programs. In addition to building on decades of COBOL language optimization, Enterprise COBOL also helps integrate COBOL applications with Web applications. It helps to:

- *Enable interoperability with Java components across distributed applications*
- *Promote the exchange and usage of data in standardized formats including XML and Unicode*
- *Support Java interoperability by new object-oriented syntax*
- *Provide XML and enhanced Unicode support*
- *Facilitate the reuse of existing applications in WebSphere and traditional z/OS environments*

CICS Transaction Server

With its reuse and Web-service generation capabilities, CICS plays a major role in the deployment stage of SOA. The capability to create and reuse services, with improved time to value, plus better access to core applications, is a compelling business proposition. The concept of reuse is often described as the capability to reuse existing code. This is part of it, but reuse also helps maximize existing skills while building new skills to address future challenges. Created components can be reused in other composite applications to benefit existing and future projects. CICS Transaction Server plays a key role in this, extending application reuse, easing application connectivity and improving service management to make full use of application resources and assets.

CICS application reuse enables you to create more-flexible and configurable components from existing applications for use in new applications. CICS Transaction Server for z/OS helps in a number of ways by providing the middleware to help build new applications from existing assets and take advantage of new business opportunities quickly.

Without technology tools, an attempt to create processes that could be linked together to create a composite application would have been an arduous task at best—if it were attempted. And yet, to be able to orchestrate a flow of disparate processes together within a CICS transaction offers tremendous business value.

CICS Service Flow Feature is a business-service integration adapter for all CICS applications. It offers both tooling and runtime capability in a single offering. These capabilities enable enterprise solution architects, enterprise integration developers and CICS application specialists to create CICS business services for integration in SOAs, business-process collaborations or enterprise solutions that take advantage of a loose coupling approach by visually composing existing CICS 3270 or COMMAREA applications.

CICS Service Flow Feature takes advantage of the tooling for the Service Flow Modeler in Rational Developer for System z. The tooling provides an integrated development environment for graphical modeling that enables the visual composition of CICS applications into a CICS business-service flow. CICS Service Flow Feature implements runtime adapters that provide access to existing CICS transaction and application interfaces using noninvasive techniques. As a result, the CICS application assets, orchestrated by the service flow, do not have to be altered to support the CICS business-service flow. This enables the fast reuse of existing assets while minimizing the risk of the new implementation.

Websphere Application Server for z/OS

While not specifically mentioned in our earlier discussion of solving our need for increasing our sales, IBM WebSphere Application Server for z/OS is a key player in our solution. Many of our Web services, newly created or repurposed, will be deployed to WebSphere Application Server for z/OS.

The combination of WebSphere Application Server for z/OS and the System z platform provides unique value in today's challenging and complex operating environments. WebSphere Application Server for z/OS is a sophisticated J2EE application server that provides a scalable, cost-effective and security-rich deployment environment for our SOA. WebSphere Application Server for z/OS is a proven and strategic platform for the mainframe. It combines the features of WebSphere Application Server with the reliability and availability of System z mainframe products. It provides the qualities of service of the WebSphere Application Server platform, such as close proximity to data, intense scalability and rich security. It integrates with the System z portfolio of hardware and z/OS software assets, and it provides significant single-tier value. WebSphere Application Server for z/OS running on the System z platform allows us to reduce our costs in establishing a high-performance, high-reliability infrastructure for our new SOA applications

Conclusion

IBM provides solutions that can help you to save time and money, and reduce the risk of improving your existing applications. These enterprise modernization solutions are designed to help you with the essential steps of repurposing your applications as part of an SOA for competitive advantage: including analyzing existing applications, enabling existing code as services, creating new services and integrating them with the repurposed assets. After you've achieved these goals, IBM CICS Transaction Server and IBM WebSphere Application Server deliver a powerful solution to deploy your new SOA.

For more information

To learn more about unleashing your System z assets for an SOA, contact your IBM representative or visit:

ibm.com/software/awdtools/wsaa

ibm.com/software/awdtools/atw

ibm.com/software/awdtools/devzseries

ibm.com/software/awdtools/cobol/zos

ibm.com/software/webservers/hats

ibm.com/software/htp/cics/tserver/v31/

ibm.com/software/awdtools/developer/business/index.html

ibm.com/software/htp/cics/ianaly

ibm.com/software/info/developer/solutions/em

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

Produced in the United States of America
09-07