August 2012

IBM Integrated Solution for System z Development

Accelerating multiplatform development

Highlights

- Build IBM System z business products and services with a complete solution for multiplatform application development
- Take advantage of a modern, integrated development environment for software delivery
- Reduce development analysis time and risk by understanding the impact of change upfront
- Lower your total cost of software delivery by deploying a common team infrastructure that automates and enforces any development process
- Free up mainframe capacity for higher value production workload and eliminate delays with a low-cost development and test environment

Today, the demand for a new generation of applications has increased the challenges of developing innovative, multiplatform, enterprise software applications. It is not easy to provide customers with modern applications while maintaining core applications. The skills, tooling and processes for the different programming languages and platforms necessary to complete a typical, multi-tier enterprise application project are often diverse and disconnected.

For example, the typical mainframe developer uses 3270-based technologies, such as Interactive System Productivity Facility (ISPF), which is text-based and can be difficult to use and learn. Productivity might suffer due to communication barriers or the use of manual processes. Testing and development can become a burden on the infrastructure and the ability to innovate.

Fortunately, development teams now have a solution that can help them create the applications customers want and their companies need. The IBM Integrated Solution for System z® Development is a modern, integrated, efficient platform for analysis, development, team collaboration, unit test, change management, and software configuration management. A combination of integrated IBM® Rational® products, this solution helps companies build enterprise applications by addressing development challenges with a complete environment for multiplatform development teams. With support for IBM CICS®, IMS™, DB2® and WebSphere® Application Server, as well as batch processing, the solution accelerates the development of mainframe and cross-platform applications to help increase productivity and reduce costs.

One of the easiest ways to understand what the IBM Integrated Solution for System z Development offers is through an example. The scenario that follows demonstrates how this solution can help your development teams.

A core application, JK Account Management, requires an update

It's no secret that large organizations in financial services, insurance, government, retail and other industries run their core business applications on System z. It's also no secret that customers prefer to have several options for doing business—one day they might want to use a browser on their home computer, but on another they might want to use a smart phone or a tablet computer. Today's mainframe applications and software need to meet these demands.

Our hypothetical development team faces these challenges. They are responsible for an application called JK Account Management. To continue to improve their customer experience with innovative, self-service offerings, they are delivering an application update that is part of a larger, mobile application project that allows customers to access account information using their smart phones.

The development organization was pragmatic in its approach to building the mobile application. With all the focus on reducing costs and risk, they made the most of their existing assets and processes. At the core of the JK Account Management application is a CICS application written in COBOL that runs on System z. The application started originally with a character-based screen user interface. The account management features of the new mobile application were achieved by extending the existing account functions of the CICS application as standard web services.

The activities in this scenario are initiated when a request to update the JK Account Management application is received by the development team. Now that customers can access the application from anywhere using their smart phones, showing an unobscured account number is a security exposure. The team has received a requirement to mask the account number wherever it is output.

For the purposes of demonstrating how the IBM Integrated Solution for System z Development works, we introduce these fictional members of the JK Account Management application development team: Cameron Murray, Analyst; Debra Hazrithi, COBOL Developer, Robin Alemaine, Team Lead and Julio Nourinho, Java Developer.

The solution in action

The requirement has been received, and now it's time to go to work.

Step 1: Analyzing the scope and risk of the requirement

After IT receives the request for the account number to be masked, they assign it to Cameron Murray, their analyst. To reduce project resource uncertainty and prevent the scope of the project from increasing, Cameron uses IBM Rational Asset Analyzer—a component of the IBM Integrated Solution for System z Development. With Rational Asset Analyzer, he not only gains a better understanding of the required modifications, but he is also able to assess the possible effects the changes might have on other application components (Figure 1).

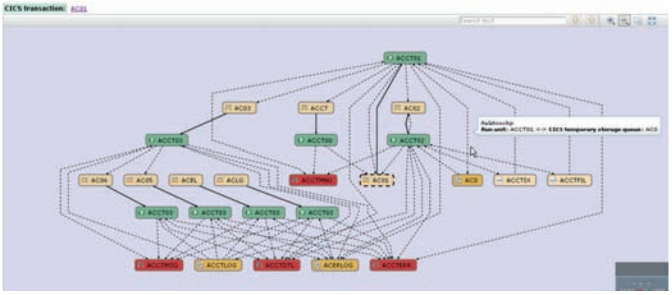


Figure 1: Rational Asset Analyzer helps Cameron understand multiple aspects of the program to determine the impacts of the requested change.

Experienced developers and new team members alike now have insight into the structure of the application and its interactions with business data. This shared application knowledge can be used throughout the project life cycle.

> *"IBM Rational Asset Analyzer software is an excellent tool for portfolio management. It saves a considerable amount of analysis effort and it is a very useful tool for large maintenance and production support projects and teams."*
>
> —Annapoorna Venkateswaran, Head of Computer Engineering, Wipro IT Business[1]

Based on his analysis, Cameron determines that the developers must modify both the CICS COBOL application and WebSphere Java code. He updates the original request with the results of his analysis, and the COBOL change and the Java change become two subtasks. The developers working on each will use the IBM Integrated Solution for System z Development.

Step 2: Updating the project status and reviewing the tasks

To update the project status and review the tasks, Debra Hazrithi (COBOL developer) and the rest of the team will use IBM Rational Team Concert™ which is integrated with the Rational Developer for System z Integrated Development Environment (IDE)—both key components of the IBM Integrated Solution for System z Development.

The team can take advantage of how Rational Team Concert combines change management, work item planning, source control, and build management into a single application. This gives the developers involved with the task a complete and real-time view of ongoing activities whenever they want it. Debra and the team quickly discover that they can accomplish more in a significantly shorter period of time, given Rational Team Concert's complete collaborative environment that facilitates both traditional and agile development and planning, project health, integrated reporting and process support. Many of the administrative tasks that they used to do manually are now automated. This means the team can concentrate more time on productive work such as coding and testing, instead of spending effort attending meetings and providing status reports.

Step 3: Understanding the program that will be modified and identifying the effects of modification

To make the update to the existing application to mask the account number, Debra starts with the analysis provided by Cameron. This directs her to what program(s) to look at for further development analysis. Using capability in the Rational Developer for System z IDE, she is able to do deeper analysis, including visualizing program control flows, listing data elements and dependent copy books, plus the ability to open called programs from within source code.

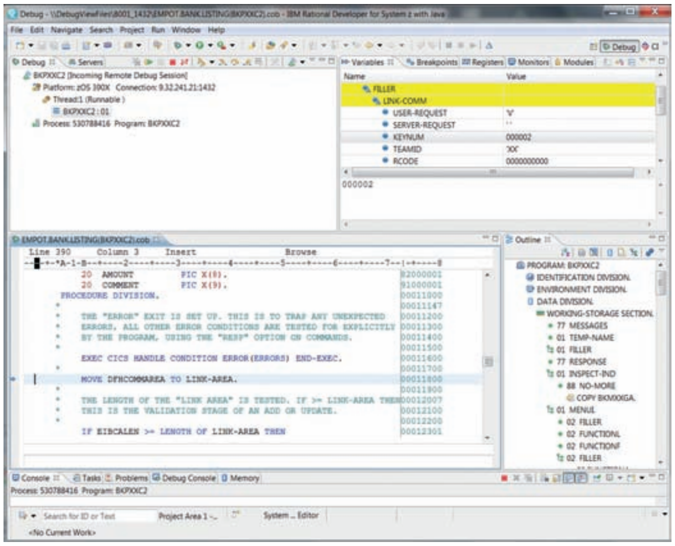
Based on the additional analysis, Debra is confident that she has identified all of the application artifacts that need to be updated to mask the account number.

Step 4: Checking out, changing and unit testing the COBOL code

Integration between Rational Team Concert and Rational Developer for System z IDE makes it possible for Debra to check out the COBOL code directly from the software configuration management (SCM) repository in Rational Team

Concert. Rational Developer for System z can integrate with a variety of System z SCM solutions, including Rational Team Concert, CA Endevor® and Serena® ChangeMan®. In addition, Rational Team Concert provides extension points so that the change management process can be integrated with existing System z SCM solutions.

Debra uses Rational Developer for System z to make the necessary changes. The Rational Developer for System z interface makes it easier for Debra to see different views at once: debug view, variable view, debug console view and editor (Figure 2). Debra is pleased to discover that IBM Rational Developer for System z software makes it easier to change applications for the mainframe. She is a relatively new mainframe developer, just learning COBOL, and she was worried that making the update would be difficult. Fortunately, Rational Developer for System z quickly allays her fears because it offers a single solution for developing, debugging, testing and deploying an enterprise application—from the display of HTML and web application logic at the mid-tier, to web services that extend CICS or IMS business logic to DB2 database access on the mainframe.


Debug - \\DebugViewFiles\8001_1432\EMPOT.BANK.LISTING(BKPXXC2).cob - IBM Rational Developer for System z with Java
File Edit Navigate Search Project Run Window Help
Debug
Debug
Servers
Variables
Breakpoints
Registers
Monitors
Modules
EMPOT.BANK.LISTING(BKPXXC2).cob
Outline
Console
Tasks
Problems
Debug Console
Memory
Search for ID or Text
Project Area 1 -...
System ... Editor
<No Current Work>

```
BKPXXC2 [Incoming Remote Debug Session]
  Platform: zOS 390X  Connection: 9.32.241.21:1432
    Thread:1 (Runnable )
      BKPXXC2 : 01
  Process: 530788416  Program: BKPXXC2
```

Name	Value
FILLER	
LINK-COMM	
USER-REQUEST	'V'
SERVER-REQUEST	' '
KEYNUM	000002
TEAMID	'XX'
RCODE	0000000000

000002

```
 Line 390     Column 3     Insert                  Browse
--█-+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7--|-+----8
           20  AMOUNT          PIC X(8).                              82000001
           20  COMMENT         PIC X(9).                              91000001
       PROCEDURE DIVISION.                                            00011000
      *                                                               00011147
      *    THE "ERROR" EXIT IS SET UP. THIS IS TO TRAP ANY UNEXPECTED 00011200
      *    ERRORS. ALL OTHER ERROR CONDITIONS ARE TESTED FOR EXPLICITLY 00011300
      *    BY THE PROGRAM. USING THE "RESP" OPTION ON COMMANDS.       00011400
      *                                                               00011500
           EXEC CICS HANDLE CONDITION ERROR(ERRORS) END-EXEC.         00011600
      *                                                               00011700
           MOVE DFHCOMMAREA TO LINK-AREA.                             00011800
      *                                                               00011900
      *    THE LENGTH OF THE "LINK AREA" IS TESTED. IF >= LINK-AREA THEN00012007
      *    THIS IS THE VALIDATION STAGE OF AN ADD OR UPDATE.          00012100
      *                                                               00012200
           IF EIBCALEN >= LENGTH OF LINK-AREA THEN                    00012301
```

```
PROGRAM: BKPXXC2
  IDENTIFICATION DIVISION.
  ENVIRONMENT DIVISION.
  DATA DIVISION.
    WORKING-STORAGE SECTION.
      77 MESSAGES
      01 TEMP-NAME
      01 FILLER
      77 RESPONSE
      01 INSPECT-IND
        88 NO-MORE
          COPY BKMXXGA.
      01 MENUL
        02 FILLER
        02 FUNCTIONL
        02 FUNCTIONF
        02 FILLER
```

Process: 530788416 Program: BKPXXC2

Figure 2: Using Rational Developer for System z IDE, Debra debugs the COBOL code.

As a result, the development environment integrates well into traditional mainframe development practices while allowing for a transition to a twenty-first-century development suite. Debra notices that her productivity has improved during this project because everything she needs to do her job is at her fingertips and many tasks that usually take significant effort are automated or simplified.

UniCredit modernizes mainframe development and testing

UniCredit Group is one of the largest financial services organizations in Europe. When UniCredit Group merged with HypoVereinsbank (HVB), the HVB Information Services (HVB IS) organization continued its mission of developing IT solutions for modern banking operations. HVB IS adopted Rational Developer for System z software to accelerate mainframe development, improve the efficiency of its COBOL development teams and move to a similar environment for mainframe and web development. More than 200 developers at HVB IS began using Rational Developer for System z software to develop mainframe applications for IBM System z servers. These developers are seeing an estimated 15 percent increase in development productivity and 10 percent increase in test productivity.[2]

After Debra completes the code changes, she moves the code to the IBM Rational Development and Test Environment for System z, an IBM z/OS® testing environment, for unit testing. This virtual System z architecture environment makes it possible for mainframe operating systems, middleware, and software to run on Intel and Intel-compatible platforms. The benefit here is that Debra does not have to use any System z mainframe hardware for her initial activities, and has more control of the test environment and conditions than is normally allowed by the system programmers on the shared mainframe test environment.

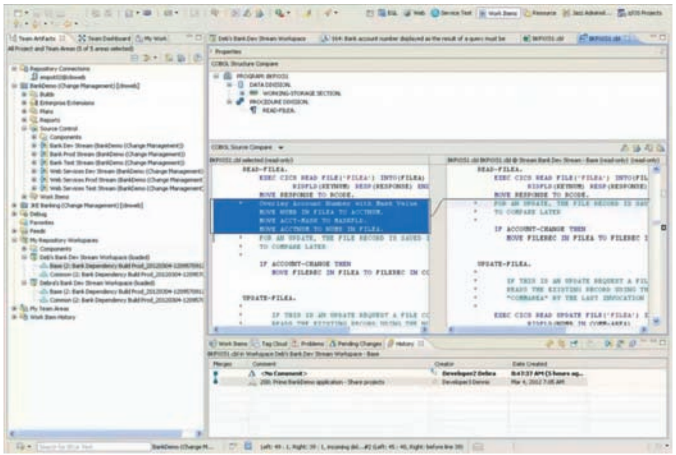
Debra notes that she can validate and test her application changes more quickly than she could for her previous projects. This is because she can make changes to the Rational Development and Test Environment for System z on her own without waiting for system programmers to evaluate and approve system wide changes that might affect the shared test environment. Before this project, waiting for mainframe change requests sometimes added weeks to delivery cycles. In addition, if she chooses, Debra can take advantage of current levels of z/OS middleware for development and testing purposes, along with greater access to new runtime capabilities. As a result, she can test the code more quickly and thoroughly without affecting the company's production environment, which not only makes the entire IT staff happy but frees up capacity on the mainframe.

Debra checks the code into the Rational Team Concert SCM repository and moves the source code from the Rational Development and Test Environment for System z to the staging area in Rational Team Concert.

Step 5: Reviewing the code

The development process requires that code created by junior developers be reviewed before it can be delivered to quality assurance. The team lead, Robin Alemaine, reviews the code (Figure 3). He can also use Rational Developer for System z capabilities such as Code Review to ensure that Debra followed the defined coding standards and Code Coverage to ensure that she unit tested the new and modified code.

He marks the code review complete and approves the change for delivery into the official Rational Team Concert build. As code is delivered, the built-in process enforcement of Rational Team Concert ensures that the development process is followed.



```
       READ-FILEA.
           EXEC CICS READ FILE('FILEA') INTO(FILEA)
                RIDFLD(KEYNUM) RESP(RESPONSE) END
           MOVE RESPONSE TO RCODE.
      *    Overlay Account Number with Mask Value
           MOVE NUMB IN FILEA TO ACCTNUM.
           MOVE ACCT-MASK TO MASKFLD.
           MOVE ACCTNUM TO NUMB IN FILEA.
      *    FOR AN UPDATE, THE FILE RECORD IS SAVED I
      *    TO COMPARE LATER
      *
           IF ACCOUNT-CHANGE THEN
              MOVE FILEREC IN FILEA TO FILEREC IN CO


       UPDATE-FILEA.
      *
      *       IF THIS IS AN UPDATE REQUEST A FILE CO
      *       READS THE EXISTING RECORD USING THE NU
```

```
       READ-FILEA.
           EXEC CICS READ FILE('FILEA') INTO(FIL
                RIDFLD(KEYNUM) RESP(RESPONSE)
           MOVE RESPONSE TO RCODE.
      *    FOR AN UPDATE, THE FILE RECORD IS SAV
      *    TO COMPARE LATER
      *
           IF ACCOUNT-CHANGE THEN
              MOVE FILEREC IN FILEA TO FILEREC I


       UPDATE-FILEA.
      *
      *       IF THIS IS AN UPDATE REQUEST A FIL
      *       READS THE EXISTING RECORD USING TH
      *       "COMMAREA" BY THE LAST INVOCATION
      *
           EXEC CICS READ UPDATE FILE('FILEA') I
                RIDFLD(NUMB IN COMM-AREA)
```

Merges	Comment	Creator	Date Created
	<No Comment>	Developer2 Debra	8:47:37 AM (3 hours ag...
	200: Prime BankDemo application - Share projects	Developer1 Dennis	Mar 4, 2012 7:05 AM

Figure 3: Using Rational Team Concert, Robin, the team lead, is able to compare files to verify the changes that are about to be delivered.

> *"Rational Developer and Test Environment for System z… shortens our time to delivery, shortens our project cycles, makes our customers happy and increases our margins… At the price… it's a no brainer."*
>
> —Eric Simone, CEO, ClearBlade

Changing the web application, reviewing the code and completing final tasks

Meanwhile, the team's Java developer, Julio Nourinho, completes the web development using Rational Developer for System z. Due to the multiplatform, Eclipse-based capabilities included in Rational Developer for System z, Julio is able to also take advantage of standard, consistent capabilities for Java development, including integrated debug, code review, and code coverage. Both subtasks, the COBOL development and the Java development, are delivered to quality assurance together, where the changes are tested. It turns out there is a minor problem with the Java portion of the change. So Julio uses the Rational Developer for System z integrated debug environment to fix the problem.

When the testing is complete, the application is delivered to production. Using Rational Team Concert, the project manager and auditors can see that the code review was done and view all the changes that were made to complete the account mask enhancement.

How the IBM Integrated Solution for System z Development brings value

As our scenario has demonstrated, the IBM Integrated Solution for System z Development provides a highly productive environment that meets the needs of experienced mainframe developers and developers accustomed to an integrated, visual, automated development platform. Table 1 provides information about how it addresses common development requirements and challenges.

Conclusion

Existing systems, competing technologies and a new generation of applications have increased the challenges of developing innovative, multiplatform, enterprise software applications. The IBM Integrated Solution for System z Development is a modern, integrated platform for mainframe and cross-platform development, team collaboration, unit testing and analysis that can accelerate the development of enterprise applications.

The IBM Integrated Solution for System z Development can help your development teams increase productivity and improve quality, reducing the time it takes to deliver software, helping to keep your company innovative and competitive. It supports modernization initiatives with tools for creating:

- Mobile applications
- Cross-platform web services
- Transactional processing
- Web 2.0 applications
- EGL support for flexible development of modern applications and services
- Java Enterprise Edition applications for deployment to WebSphere Application Server

It also provides accurate, comprehensive visual dependency analysis so your teams can export and share analysis results for high quality maintenance and new development. The solution provides a built-in process enforcement capability to help you meet audit and compliance mandates. In addition, it offers team reports and web dashboards to help track the health of your projects at all levels of detail, along with team communication features such as event feeds, integrated chat, automated traceability and simple email notifications.

Like all IBM solutions, the IBM Integrated Solution System z Development is supported by an experienced worldwide staff of professional services consultants. We can implement solutions and support your staff throughout the entire application modernization process from discovery, analysis, modeling and design to assembly, test, deployment and management for your System z platform.

Challenge	Offering	Business Benefit
Productivity obstacles with separate, character-based tools	IBM Integrated Solution for System z Development, with integrated collaboration, change management, SCM, analysis, development and unit test	• Rational Development for System z software can improve mainframe developer productivity by 15 to 44 percent when analyzing, coding, compiling, testing and debugging[3]
High development MIPS consumption	Rational Development and Test Environment for System z, harnessing the processing power of distributed systems	• Companies can see a compile cost reduction of as much as 15 percent and 10 percent less cost for TSO sessions[4] • MIPS capacity is freed up for higher value, production workload
• Long waits for mainframe test cycles • Short test windows because the mainframe is running at or near capacity	Rational Development and Test Environment for System z, installed on a shared server	• Test cycles are available when they are needed for as long as necessary • There is no waiting for the operations team for support for simple environment changes
• Manual analysis that typically takes up to 45 percent of a developer's time	• Rational Asset Analyzer • Rational Developer for System z	• Up front, overall analysis with Rational Asset Analyzer cuts analysis—sometimes from months to days[5] • Rational Developer for System z can potentially improve analysis time by 40 percent[6]
Large proportion of development time spent on business and technical communications (for example, email, status meetings, information gathering and reporting)	Rational Team Concert	• Geographical, technological and organizational challenges can be overcome with one place to go to view and collaborate on projects • Automation and improvement of the development processes can lead to a reduction in administrative overhead of over 20 percent[7]
Difficult to enforce, manual processes	Rational Team Concert	Process enforcement is part of Rational Team Concert to help customers meet audit and compliance mandates
Isolated developers, tooling and processes for different platforms and programming languages	Common team infrastructure based on the IBM Jazz™ platform	Team productivity can increase by as much as 50 percent[8]
Pressure to modernize 20+ year-old core applications that represent billions of dollars of investments	IBM Integrated Solution for System z Development: capabilities like application inventory, business rules modernization, analysis, and application refactoring and modernization on a common, Jazz-based team infrastructure	On average, reusing an existing, proven application is five times less expensive than rewriting[9], plus results in faster time to market and less risk
Difficulty attracting new developers: 80 percent of mainframe developers are older than 50 and considering retirement	IBM Integrated Solution for System z Development: with integrated collaboration, change management, SCM, analysis, development and unit testing	A highly productive environment meets the needs of developers accustomed to an integrated, visual, automated development platform

Table 1: How the IBM Integrated Solution for System z Development addresses challenges and requirements.

For more information

To learn more about the IBM Integrated Solution for System z Development, please contact your IBM marketing representative or IBM Business Partner, or visit: **ibm.com**/software/rational/integrated/systemz-development/

Try it online at no charge

The IBM Enterprise Modernization Sandbox for System z serves as a hands-on environment for trying Rational design, development and lifecycle management solutions for IBM System z applications with short, guided exercises, online demos and best practice documentation in a free, cloud-based environment.

ibm.com/developerworks/downloads/emsandbox_systemz/

Additionally, IBM Global Financing can help you acquire the software capabilities that your business needs in the most cost-effective and strategic way possible. We'll partner with credit-qualified clients to customize a financing solution to suit your business and development goals, enable effective cash management, and improve your total cost of ownership. Fund your critical IT investment and propel your business forward with IBM Global Financing. For more information, visit: **ibm.com**/financing

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

Produced in the United States of America
August 2012

[1] From the Wipro reference, http://www-01.ibm.com/software/success/cssdb.nsf/CS/JJAE-8EHRRL?OpenDocument&Site=software&cty=en_us

[2] To learn more, visit: http://www-01.ibm.com/software/success/cssdb.nsf/CS/WJBN-7K7P8B?OpenDocument&Site=rational&cty=en_us

[3] These numbers are based on IBM customer surveys. Your results may vary.

[4] These numbers are based on IBM customer surveys. Your results may vary.

[5] These numbers are based on IBM customer surveys. Your results may vary.

[6] These numbers are based on IBM customer surveys. Your results may vary.

[7] These numbers are based on IBM customer surveys. Your results may vary.

[8] These numbers are based on IBM customer surveys. Your results may vary.

[9] Software Productivity Research (SPR) study