



Accelerating delivery of smarter products and services on System z

Many shops have a perennial problem of a shortage of production MIPS. With many mainframes running a 95 percent or greater utilization, there are few options for relatively inexpensive increases in MIPS availability. This is becoming a more visible issue for many companies. In interviews with our customers, many indicated that they felt limited by the resources they are assigned as their organization constantly shifts mainframe capacity towards production. Many shops seek to do more with less by reducing “Development MIPS,” for example by editing code on the workstation, moving applications to Java™ to take advantage of zAAP processors, or testing code off platform using alternate technologies.

The nature of mainframe applications and their development and delivery is changing. Nowadays, organizations are under increasing pressure to deliver new solutions in shorter time. This and other developments are driving mainframe application development to adopt some of the more flexible, open practices used in the development of distributed applications. Business pressures are causing release cycles to shorten, and there is increasing demand for accessibility to mainframe assets from traditional services, like CICS®, IMS™, DB2®, and more modern services like Java, and WAS.



In addition, newer development models such as agile and scrum, assume that “continuous builds”—builds that are run early and often—are standard procedure. This is in sharp contrast to the decades-old traditional methods of developing mainframe applications.

Agile development and the production environment

In production mainframe installations, the infrastructure necessary to support development and testing is often considered an expensive but necessary overhead—overhead that consumes MIPS, staff time, and other resources that could instead be used to run the production application portfolio that delivers value to the business. Nonproduction use of mainframe assets is scrutinized and rationed—by chargebacks and by restricting some noncritical development and test activities to off-peak hours. Clearly this conflicts with a development model that presumes continuous availability of mainframe resources and the ability to rapidly develop and deploy new workload solutions.

What has been needed is a dedicated, z/OS® environment for development and test work, which is available on demand to the Rational® Developer for System z® (RDz) developer. Even better would be the ability to provide this as a low-cost, easily available, z/OS development and test environment where developers can build and test application modifications that are functional (as opposed to performance related) without consuming production staff or more expensive production environment MIPS. The Rational Developer for System z Unit Test feature (RDz UT) was developed to provide just such a low-cost solution suitable for developing and unit testing by an individual developer. RDz UT is intended to be deployable by a single developer or small team onto an x86-based Linux® laptop/desktop or server.

Today, with RDz, it is possible to offload some development tasks such as syntax checking, editing, and analysis to the desktop. However, certain verification tasks such as validating a bind to CICS or DB2, or ensuring that a complete and correct message is sent to MQ, are best tested to an acceptable level of confidence in a live z/OS environment. The same applies to live debug, or getting an accurate characterization of storage utilization. These kinds of tasks cannot be done as reliably in a simulated environment.

Rational Developer for System z Unit Test Feature: A low-cost rapid prototyping solution for z/OS applications

To meet this requirement, IBM now offers the Rational Developer for System z Unit Test Feature (RDz UT) [already defined]. RDz UT is based on IBM's System z Personal Development Tool, (zPDT), which enables a user to run a personal System z development and test environment on an x86-based Linux environment.

How does the RDz UT work? Is it real z/OS?

The RDz UT hosts the actual z/OS operating system on an x86-based Linux platform. RDz UT also comes packaged with a preconfigured set of “typical” IBM z/OS software solutions which you are entitled to use only for development and test activities in the RDz UT environment. It is intended to provide the RDz z/OS application developer with the most accurate System z environment that can support rapid prototyping.

The solutions packaged with RDZ UT include:

- z/OS V1.10 and V1.11, including subfeatures
- WebSphere® Application Server for z/OS V7.0
- CICS Transaction Server V3.2 and V4.1
- IMS V10.1
- DB2 for z/OS V9.1
- WebSphere MQ for z/OS V7.0
- IBM Java SDK V5.0 and V6.0
- Enterprise COBOL V4.2
- Enterprise PL/I V3.8
- XL C++ V1.10 and V1.11
- IBM Debug Tool V9.1

The solutions in RDz UT are similar to the actual IBM mainframe software solutions. However the differences in hardware characteristics mean that the RDz UT is not appropriate for testing performance or applications which are precisely tuned to the performance characteristics of mainframe hardware.

As the name implies, the purpose of RDz UT is to support unit and functional testing activities only. RDz UT may not be used for production workloads of any kind, including, without limitation, production module builds, preproduction testing, stress testing or performance testing. If any of the products are needed for production purposes, full-function licenses must be purchased separately from IBM.

With RDz UT, the developer can prototype their application in a live z/OS environment, without consuming production MIPS or cycles from support staff.

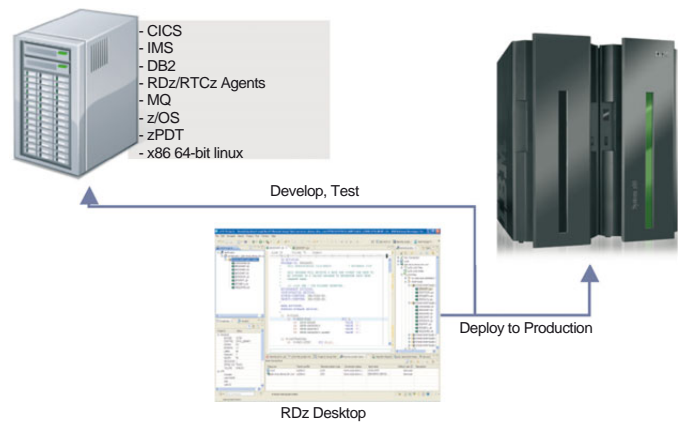


Figure 1: Using RDz UT for rapid prototyping

Getting started with the RDz UT

As mentioned earlier, the RDz UT is based on x86-based Linux environment. You will need a system with adequate memory and processor capacity to run a 64-bit Linux distribution. The supported Linux distributions are SuSE and Red Hat.

In addition to the RDz UT software, the product requires the existence of a hardware USB key which is provided with the solution. You should plan to use a dedicated physical machine to run your RDz UT environment. Other workloads, especially those that are graphics or I/O intensive should not be run on the same hardware.

RDz UT comes in two configurations. RDz UT Standard is a single engine solution. You can also choose the RDz UT Specialty offering, which runs up to three engines. You will have the best performance if you plan for at least one more Intel® Core processor than the number of engines you intend to deploy. For example, the RDz Unit Test Standard Configuration would run best on a dual-core machine. If you have applications that are intended for zAAP, or zIIP, you can use a four-core Intel processor machine and configure each of the three remaining engines to run the appropriate combination of the target environments.

You will need approximately 80 GB of free disc space. Once you have identified your hardware platform, you install your chosen Linux distribution, then RDz UT.

There are obvious physical limitations to running z/OS on desktop hardware. There are some testing tasks that must be run natively in the production environment. Only a subset of devices is available with RDz UT. So a test that presumes a particular network device, for example, or on the memory speed of System z would not be valid on RDz UT. That said, prototyping work which would typically only require a simple compile, link, bind and run, can now be performed with RDz UT instead.

Why use RDz UT?

Unlike other mainframe test solutions, which attempt to simulate z/OS, RDz UT runs the actual z/OS operating system. The software solutions included are the actual IBM software releases currently available. Using the actual

software environment ensures the highest possible fidelity to a production environment. This means that the risk of false test results or spurious errors is minimized when the application is moved back to the mainframe for more comprehensive testing.

The RDz UT feature is intended to be deployed and managed at the team or individual level, which simplifies configuring of multiple test environments.

Knowledgeable developers and testers can tailor the environment as required without waiting on support from production. If needed, for example, to validate forward or backward compatibility of your solution, you can run parallel RDz UT environments at different software levels.

System Programming staff can be relieved of the burden of handling requests for nonproduction resources, and development chargebacks can diminish significantly. At the same time, more host resources can become available for production.

Taking your mainframe development skills to the next level

The Rational Developer for System z Unit Test feature is a key component of IBM's Enterprise Modernization (EM) Workbench. The EM Workbench is a set of solutions, which enable rapid analysis and discovery, integrated development and testing, and comprehensive life-cycle management and deployment of mixed applications.

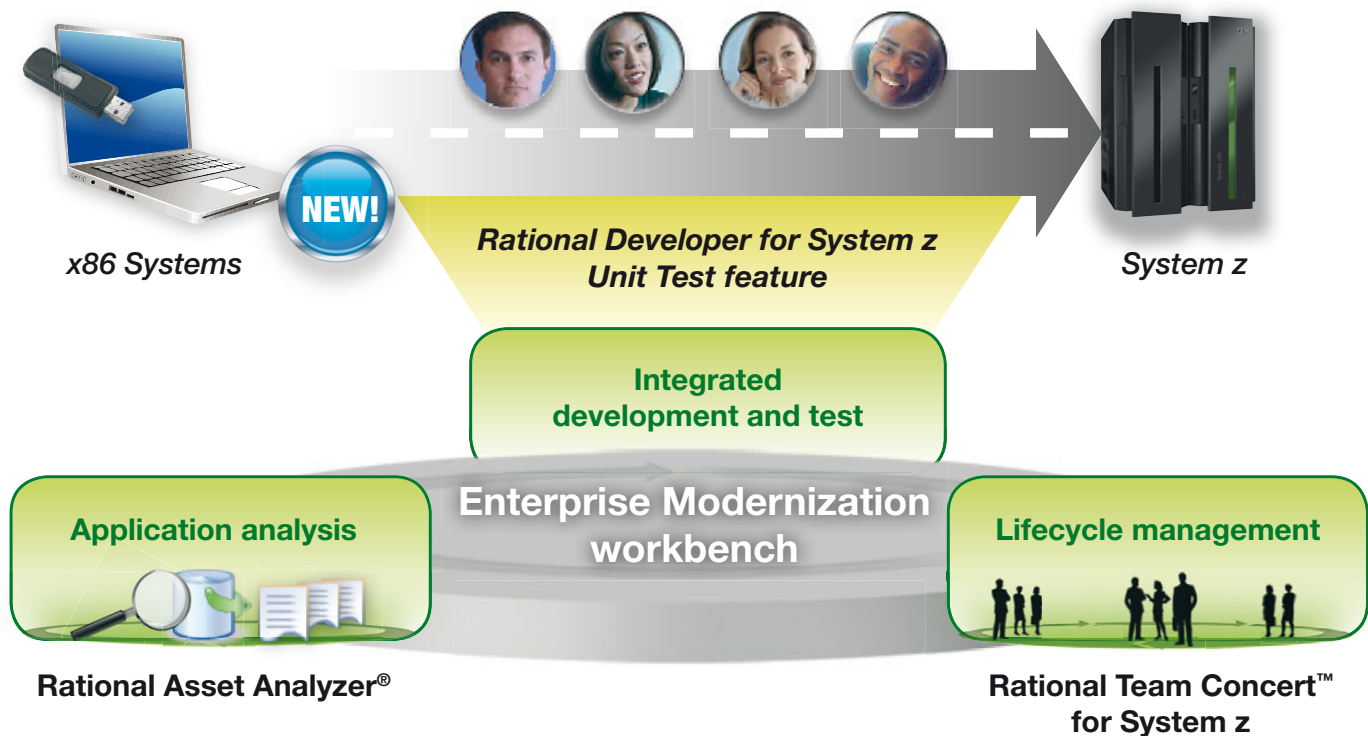


Figure 2: The IBM Enterprise Modernization Workbench

Summary

The IBM Rational Developer for System z Unit Test feature can greatly enhance the way organizations develop, maintain and test mainframe applications.

The new Rational Developer for System z Unit Test feature helps you:

- Liberate your developers to rapidly prototype new applications
- Build and test new IBM System z applications virtually anytime!

- Provide an easily accessible z/OS development and test environment to a wider range of mainframe developers and testers
- Take advantage of a more modern and agile team workbench that accelerates delivery of new workloads which span multiple platforms.
- Free up mainframe development MIPS for production capacity

The RDz UT feature is the definitive solution for the problem of providing a low-cost, instantly available z/OS development and test environment appropriate for individual development.

For more information

To learn more about the RDz UT or the Enterprise Modernization Workbench, please contact your IBM marketing representative or IBM Business Partner, or visit the following website:

ibm.com/software/rational/announce/modernization/

Additionally, financing solutions from IBM Global Financing can enable effective cash management, protection from technology obsolescence, improved total cost of ownership and return on investment. Also, our Global Asset Recovery Services help address environmental concerns with new, more energy-efficient solutions. For more information on IBM Global Financing, visit: ibm.com/financing



© Copyright IBM Corporation 2010

Produced in the United States of America
May 2010
All Rights Reserved

IBM, the IBM logo, ibm.com CICS, DB2, IMS and System z 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

Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

Note: This Program is licensed only for development and test of applications that run on IBM z/OS. The Program may not be used to run production workloads of any kind, nor more robust development workloads including without limitation production module builds, preproduction testing, stress testing, or performance testing.



Please Recycle