



# **Boost mainframe workload deployment with IBM System z Solution Edition for Application Development**

*By Jim Porell, Distinguished Engineer, IBM System z*

---

## Contents

---

- 2** *Executive summary*
- 3** *The operational value of the IBM mainframe today*
- 5** *The evolution of application development*
- 7** *How Rational software provides a consistent approach across platforms*
- 9** *How Rational Developer for System z makes the connection transparent*
- 10** *How System z Solution Edition for Application Development makes it affordable*
- 11** *Competitive value*
- 11** *Net result: Affordable hybrid computing*
- 12** *For more information*

***IBM Rational software enables programming consistency across all major operational environments.***

## Executive summary

The IBM System z® mainframe is well known for its operational value. Across a number of industries, including finance, government and retail, it is the de facto online transaction processing server and high-end database. Its operational qualities of service are well known, particularly for workload security, high availability, capacity management, business process integration and storage management.

However, the IBM mainframe is also perceived to be an older system for application development and new workload deployment. For many customers, the IBM mainframe is in operations mode. It's considered too expensive to develop for, and there is a perceived lack of skills for developing on the IBM mainframe due to its proprietary heritage as "green screen" and its batch programming interfaces. In addition, some customers fear that the IBM mainframe is unaffordable and inaccessible to application developers. The operating system and middleware is considered too expensive, and the IT operations team that manages the IBM mainframe may discourage system changes in order to preserve high availability. These perceptions don't bode well for new workload growth on the IBM mainframe.

IBM Rational® software provides an application lifecycle management solution that delivers a consistent front-end development interface across multiple operating systems. More importantly, it provides the same interface for the traditional mainframe transaction processing environment as it does for new workloads, such as Java™ and C/C++ applications. This programming consistency enables a business to utilize the same programming skills across all major operational environments, reducing the skills and knowledge necessary to work across multiple platforms.

---

## Highlights

---

Recently, IBM introduced the IBM System z Solution Edition for Application Development. This offering provides mainframe processing hardware, the virtualization hypervisor (IBM z/VM®), the IBM mainframe operating system (IBM z/OS®), and the majority of IBM middleware for z/OS at a fixed price that is intended to be price competitive with the development environments of other platforms. This offering enables individual developers to have their own operating system images for development and test purposes. The code developed in this environment is the same code that goes into production.

Using other development tools, a business might have one tool for Microsoft® Windows® and other tools for other platforms. They might also have tools that attempt to mimic the behavior of the mainframe in order to reduce development expense. Those tools may require changes in order to execute in the native mainframe operational environment, which may require additional skills and time to develop.

As a combination of offerings, IBM Rational Developer for System z and System z Solution Edition for Application Development are designed to provide an affordable development environment that provides cross-platform consistency while enabling developers to control their own efforts in delivering valuable applications for mainframe and distributed system deployment.

***The combination of IBM Rational Developer for System z and IBM System z Solution Edition for Application Development provides an affordable development environment for IBM mainframe and distributed system deployment.***

### **The operational value of the IBM mainframe today**

Businesses are constantly being given choices for the deployment of their mission-critical operations. While the PC is often the best choice for the front end of application lifecycle management, there are other choices at the deployment end of a workload. For operational value, the IBM mainframe provides significant advantages over alternative server implementations.

---

## Highlights

---

In a distributed environment, applications are typically in one server operating system image and the database is in another operating image. With the advent of virtualization in the distributed arena, via VMware, Xen, Microsoft Hyper-V, pHype and other virtualization providers, some consolidation of multiple operating system images can now occur on a single physical server. The hypervisors have added in some of the knowledge necessary to share across workloads; however, the operating systems themselves still continue to deploy a single workload at a time to reduce risk and maintain service level agreements.

The IBM mainframe provides a very different level of virtualization. The z/VM hypervisor was originally introduced in 1967. It has evolved to provide tremendous granularity with regard to processors, memory, storage and the networking infrastructure. Customers are capable of running thousands of virtual system images on the same physical server. Process Resource/Systems Manager (PR/SM) logical partition (LPAR) technology was introduced in 1989 to provide a higher level of abstraction and security compartmentalization so that up to 60 logical partitions can now be created on a single machine. The z/OS operating system is capable of running thousands of application and database servers in the same operating system image because its unique hardware, firmware, operating system, and middleware functionality provide an architected boundary between work units, ensuring that very high levels of service can be achieved.

***The z/OS operating system is capable of running thousands of application and database servers in the same operating system image.***

The different deployment models—fully distributed (original client server), virtually distributed (separate operating system images) and virtually shared (z/OS)—have completely different operational models. These operational

models change the cost of ownership for the deployment of new workloads. In particular, the management needs for security, availability/disaster recovery, capacity management, business process integration, and storage management will have completely different operational models when considered across the different deployment models.

For example, in the fully distributed model, a client will sign in to the applications server, which will in turn sign in to the database server. To identify the client that accessed the database at any given time, the logs of both the application server and the database server must be correlated. On the IBM mainframe, a single log can be accessed to capture this information. When capacity needs to be added to the distributed model, additional servers and networking may be required. On the IBM mainframe, an additional engine might be added and the capacity spread across all workloads. The IBM mainframe may also consume less floor space, cooling, electricity and full-time employees, helping to reduce the total cost of ownership.

### **The evolution of application development**

From the 1960's through the early 1980's, the traditional operational model was on a mainframe. Large machines with water-cooled processors, punch cards, and dumb 3270 terminals as front-end devices were running over a customer-owned SNA network. The IT group dealing with mainframe operations worked closely with the development teams that were satisfying new customer requirements.

The introduction of the IBM PC added some versatility to the front-end computing experience. Not only was the PC cheaper to acquire than the 3270 terminal, it added new functionality and value for the end user. It introduced a new IT requirement, as well. Software had to be configured for the

PC in order to operate in conjunction with the mainframe. This PC setup was initially the responsibility of the end user, but most large businesses created small PC service teams to handle these tasks.

The beginning of a new programming paradigm began with the creation of screen scraping applications within the 3270 emulator sessions. No longer was the end user responsible for entering all the input into a command line or text field. They could now take advantage of PC graphical capabilities such as with scroll bars and point-and-click operations. In this way, a list of choices could be selected by the end user. This provided a productivity enhancement for the end user, and it reduced the number of syntactical or spelling errors that needed to be processed on the mainframe. To simplify the deployment model for PCs, local area networks (LANs) were created so that the information on the screens could be shared by an organization and not require modification of PC setups. The TN3270 terminal was introduced so that commodity PCs could be deployed and plugged into a corporate network with little or no setup by the end user. Web servers were also added to the LAN environment, providing new channels for consumers to enter transactions instead of having to go through a customer service agent.

All of these innovations led to increased use of the mainframe on the operations side but a dramatic reduction in the amount of development that was targeted for deployment on the mainframe. In essence, the mainframe evolved to maintenance mode for existing applications.

The largest change to mainframe operations began when portals and desktop interfaces attempted to capture the input for multiple mainframe transactions via a single interface and run them as stateful transactions across a distributed environment. This greatly added to the instruction path length when computing across a network. Locks are held between systems that inhibit other transactions from executing until the locked unit of work completes. The result is that the overall capacity of the end-to-end environment diminishes while waiting for the distributed environment to complete the transaction because the instruction path length is much longer. Laboratory testing has shown that this path length can be increased two to five times in the distributed environment versus on z/OS. This can increase the cost of computing.

The mainframe has evolved to process this workload in a single operating system image. This enables Web application and traditional transaction processing to occur in a single operating system, which processes this stateful transaction in memory instead of across a network. In doing so, the processing power consumed on the mainframe is remarkably similar to the original 3270 transactions but with a better end-user experience.

### **How Rational software provides a consistent approach across platforms**

Rational software provides an integrated development environment (IDE) that manages the development of software from requirements through deployment (see Figure 1). Like most IDEs, the Rational Suite begins on a desktop. It leverages the Eclipse structure so that other vendors can contribute toward application lifecycle management (ALM) as well.

---

Highlights

---

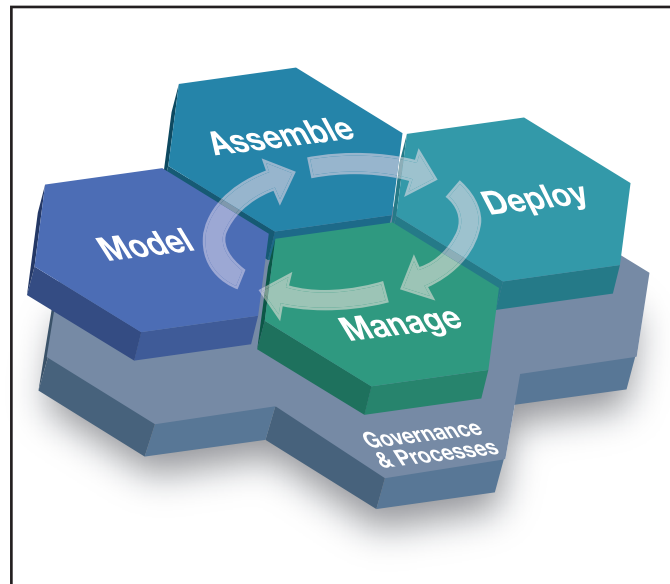


Figure 1: Rational software provides an integrated development environment that manages the development of software from requirements through deployment.

***The Rational Suite provides tooling that will generate or manage code in a wide variety of traditional and next-generation programming languages, including Java, C, C++, COBOL, PL/I, and EGL.***

The Rational Suite provides tooling that will generate or manage code in a wide variety of traditional and next-generation programming languages, including Java, C, C++, COBOL and PL/I. In addition, a higher level language, Enterprise Generation Language (EGL), has been introduced to enable a single developer to generate code in a high-level abstract language that can be easily translated, through tooling, to any of the other popular languages.

The Rational Suite also enables applications to run in a variety of transaction processing environments on the IBM mainframe and other platforms. It is this consistency of development and common tooling across platforms that enables skills portability and ease of deployment, reducing the level of effort required and enabling developers to target the platforms that best fit the particular application being deployed.



**Highlights**

*Rational Developer for System z provides a specific set of IBM mainframe-centric tooling that enables a developer to build and test applications using the mainframe, while not requiring unique IBM mainframe knowledge.*

**How Rational Developer for System z makes the connection transparent**

Rational Developer for System z is the on-ramp for bringing the Rational Suite of tools to the IBM mainframe. Rational Developer for System z provides a specific set of IBM mainframe-centric tooling that enables a developer to build and test applications using the mainframe, while not requiring unique IBM mainframe knowledge. Applications can be developed and maintained within the Rational Suite of tools, while changes can be made on the desktop in a high-productivity environment. The code is transparently accessed on System z for compilation and function testing using the same IDE tools, using traditional mainframe runtimes as a deployment and testing platform. This provides an additional level of skills portability while at the same time allowing the code to inherit the operational capabilities of the mainframe when it is put into production.

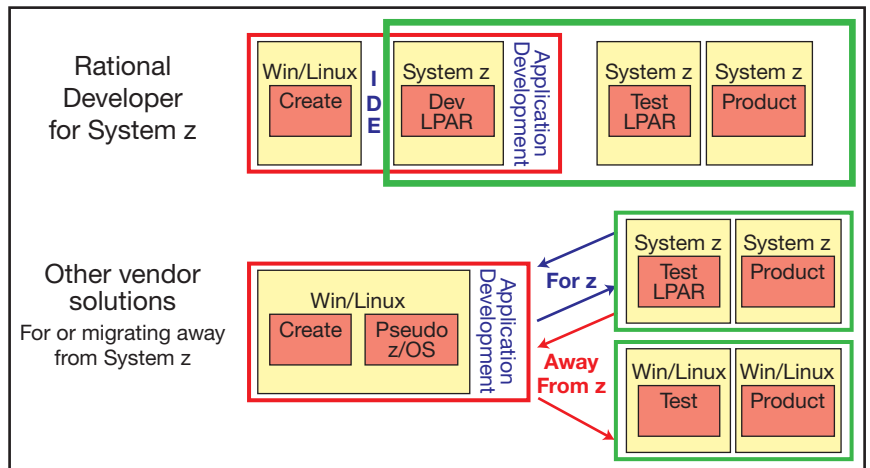


Figure 2: Rational Developer for System z supports the application development lifecycle by providing an affordable and simple development environment and deployment container.

---

**Highlights**

---

***System z Solution Edition for Application Development provides a mainframe hardware container, the z/VM hypervisor, the z/OS operating system, and most of the IBM middleware available for z/OS.***

**System z Solution Edition for Application Development is cost competitive with any platform**

System z Solution Edition for Application Development completely changes the acquisition costs for IBM mainframes that will be used for development, proof of concept, and function testing purposes. The solution includes a mainframe hardware container, the z/VM hypervisor, the z/OS operating system, and most of the IBM middleware available for z/OS (e.g. IBM DB2®, IBM CICS®, IBM IMS™, IBM WebSphere®).

The goal of System z Solution Edition for Application Development is to enable application developers to have their own mainframe images through which they can build and test z/OS applications. Each developer can then modify that system image without impacting other developer system images, providing new levels of development flexibility and control.

By enabling the IBM mainframe to be a target deployment environment, developers can experience the operational value of having applications and data co-located in the same operating system image. System z Solution Edition for Application Development allows developers to functionally test their code in an environment that mimics the production environment.

The major benefit of this solution is that it provides a valuable cost-competitive development environment for IBM mainframe customers. Current z/OS offerings are priced on machine capacity. As the capacity grows, so does the cost per seat for each developer. System z Solution Edition for Application Development allows you to avoid this by providing a separate hardware container with the operating system and middleware, for a significantly lower-cost solution for development.

---

## Highlights

---

*The System z Solution Edition for Application Development and Rational Developer for System z provide an ultra-low cost, high-productivity environment for creating and maintaining applications on IBM mainframe systems.*

The System z Solution Edition for Application Development and Rational Developer for System z offerings complement each other to provide an ultra-low cost, high-productivity environment for creating and maintaining applications on IBM mainframe systems as well as exploring new applications and architectures. Rational Developer for System z makes the coordination of work between the two environments seamless.

### **Competitive value**

Customers have choices for their development and deployment environments. Some development platforms only target a specific deployment environment. Other development tools attempt to mimic the environment of the mainframe to attempt to reduce development costs. But in doing so, they don't provide the same operational experience that would occur in production on the mainframe.

By combining Rational Developer for System z and System z Solution Edition for Application Development, IT organizations can provide a development environment that facilitates deployment on many of the server platforms that the business may have deployed.

### **Net result: Affordable hybrid computing**

Businesses are run on any combination of computing systems. In addition, one business may interoperate with another business, so there must be standardization of formats and protocols in order to distribute applications and data between systems. It becomes critical that standardized programming tools be made available to meet these intra- and inter-system business needs.

The combination of Rational Developer for System z and System z Solution Edition for Application Development provides a development environment that enables cross-platform deployment operations, operational fidelity with the target deployment platforms, and a fixed cost for each application developer, providing a much more competitive offering for application lifecycle management.



## For more information

To learn more about System z Solution Edition for Application Development, please contact your IBM marketing representative or IBM Business Partner, or visit: [ibm.com/systems/z/solutions/editions](http://ibm.com/systems/z/solutions/editions)

© Copyright IBM Corporation 2009

IBM Corporation  
IBM Systems and Technology Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
November 2009  
All Rights Reserved

IBM, the IBM logo, ibm.com 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](http://ibm.com/legal/copytrade.shtml)

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

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice.

IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

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



Recyclable, please recycle

ZSW03135-USEN-00