



**Rational** software

## **Extending the value of RPG applications on the IBM System i platform.**

---

**Contents**

---

- 2 Introduction**
- 3 When does it make sense to modernize RPG applications?**
- 6 Protecting IT investments**
- 6 Reasons to consider migrating to IBM Rational Business Developer software**
- 8 Potential application transformation scenarios**
- 11 Conclusion**

**Introduction**

Applications based on the RPG programming language running on the IBM System i™ platform are the cornerstone of many companies' business processes. While early RPG applications still run the way they always have, these applications have grown more comprehensive and complex with time. And because these existing programs have always been executable in new versions of the IBM i5/OS® operating system without having to make any changes, a number of them were developed with technology that was state of the art 15 or 20 years ago. Similarly, there are programs that use the latest RPG IV features and a seemingly endless range of feature variations.

Developed in parallel with System i and RPG technologies, new platforms with modern GUIs have been created, such as the Java™ and Microsoft® .NET environments. These platforms have quickly established themselves as standards, representing a potent generational change in application development. To maintain a competitive edge in the marketplace by exploiting advances in modern architectures and middleware, today's businesses will have to make this transformational shift. According to a recent press release from Gartner, "Although IT management teams may well be able to keep systems on life support for some time, there is a finite limit to the willingness of business users to keep on using solutions that fail to deliver modern standards of functionality and agility."<sup>1</sup>

<sup>1</sup> Gartner, "Gartner Says IT Leaders Must Place IT Modernization at the Core of Their 2008 Objectives," press release, February 26, 2008, <http://www.gartner.com/it/page.jsp?id=611507>.

---

## Highlights

---

***Organizations can reuse and leverage their existing IT assets to realize more flexibility, improve application development productivity and support new business models.***

***High maintenance costs, reduced application agility or the development platform losing its edge all point to the need to migrate an application to a new platform.***

Yet organizations have already invested heavily in legacy systems, applications and staff. And for many, their systems and RPG code are assets that continue to hold great value. So in order to make this generational change economically feasible, organizations need to transfer a large proportion of their existing applications and skills to the new platform, so they can be reused, while serving as a launching pad for step-by-step modernization. Hence, the migration of RPG applications to the IBM System i platform is one option in achieving this transformation.

This paper discusses how enterprises are using IBM Rational® Business Developer software to cost-effectively achieve greater flexibility, increase application development productivity and support newer business models such as a service-oriented architecture (SOA) – all while reusing and leveraging their existing IT assets.

### **When does it make sense to modernize RPG applications?**

Businesses typically spend huge sums of money developing custom applications that set them apart from competitors and optimally support their specific business processes. Yet when it comes to functionality and technology, these custom applications can quickly grow old and lose their value.

Through maintenance, functional aging can be circumvented or at least significantly slowed. Most of the time, however, technical aging goes unnoticed until unreasonably high maintenance costs arise later on in the lifecycle, application agility declines, or the development platform begins dying a slow death. These are all indications that it's time to migrate an application to a new technological platform before its costs outweigh its benefits.

---

## Highlights

---

**Several factors make transformation necessary, including extending the application lifecycle, needing an application on another platform, transitioning to an SOA, using GUIs more efficiently, improving agility, increasing competitiveness, losing RPG developer skills and consolidating development teams.**

When they reach this point, companies often find themselves in a bind. They realize that it's not cost-effective to simply replace their legacy applications with modern standard software. And few can afford to continually rewrite all of their custom software to keep it up to date. Even if they can afford rewrites, companies realize that the money could be better spent on a more cost-effective approach to modernizing and migrating to the latest technical standards. Moreover, several factors make these transformations increasingly necessary.

- **Extending the application lifecycle.** *Maintaining older, business-critical applications becomes increasingly difficult to justify as investments in RPG training continue to decline.*
- **Needing an application on another platform.** *Within the scope of platform strategies or consolidations, the application is required on a non-System i platform such as the Linux<sup>®</sup>, UNIX<sup>®</sup>, Microsoft Windows<sup>®</sup> or IBM System z<sup>™</sup> platforms—or an IBM DB2<sup>®</sup> Universal Database<sup>™</sup>, Oracle or structured query language (SQL) server is needed in place of a DB2 server on the IBM AS/400<sup>®</sup> platform.*
- **Transitioning to an SOA.** *When existing monolithic architectures are broken down by business processes or an SOA, supporting the singular assets through the language and development environment is vital. Additionally, optimal support of Web services and languages for business process modeling, such as Business Process Execution Language (BPEL), is warranted.*
- **Using GUIs more efficiently.** *Users who work primarily with Windows or Web-based applications find the transition to green-screen applications to be a nuisance in the flow of operation. This is because there is no reasonable integration between the interfaces, and usage is typically tied to intensive training efforts. In addition, management may misinterpret green-screen applications as an indication that the IT department is not as innovative as it could be.*

---

---

Highlights

---

---

- **Improving agility.** *It is difficult to make changes to applications that develop over time. Consequently, introducing new business processes can become expensive and time consuming, seriously hindering an organization's ability to compete.*
- **Increasing competitiveness.** *Now that Java and Microsoft .NET applications are the new platform standards, RPG standard software vendors are having major problems attracting new customers. Companies must now look outside of RPG if they hope to remain competitive.*
- **Vanishing RPG developer skills.** *Apart from weakened innovative power, the maintenance and development of existing applications are endangered as RPG developer skills become increasingly scarce.*
- **Consolidating development teams.** *Many of today's enterprises have mixed developer teams that deal with RPG, Java and even older fourth-generation language (4GL) and COBOL applications. This makes development less flexible and more costly. By merging different development teams, companies can leverage a standard modern language and development environment to increase flexibility and cut costs.*

***While transformation represents the source of the solution to these issues, applications need to have a minimum standard of quality in order for the transformation to be economically feasible.***

Together with reengineering and modularization, transformation represents the basis of the solution to these issues. According to Phil Murphy of Forrester Research, "As an industry, IT tends to dismiss rather than reuse previous technology genres and, as a result, has spent vast sums to rip and replace technology for technology's sake, often at the expense of business value. With value-to-the-business being the rallying cry of the 21st century IT, CIOs are struggling to reconcile the ways of the past with value-to-the-business, creating a resurgence in interest in legacy modernization."<sup>2</sup> However, for transformation to make good economic sense, the application in question must demonstrate a minimum standard of quality.

---

### Highlights

---

*There are three established approaches to instituting a change in application development: replacing the application with standard software, rewriting code and migrating to a new environment.*

#### Protecting IT investments

Traditionally, there have been three approaches to initiating a generational change in application development:

- **Replace the application with standard software.** *Substituting an existing application with standard software from a corresponding vendor takes a lot of effort, since the move requires organizational changes and creates new dependencies on a vendor. It may make sense for standard processes, but not for competitive-critical processes in which the differentiation potential is highly limited.*
- **Rewrite code.** *Rewriting code goes hand-in-hand with high costs and scheduling risks. Additionally, existing RPG applications are often not documented in enough detail; therefore, their functional features aren't given adequate consideration during redevelopment. When introducing the rewritten functionality, this can lead to substantial interference in the flow of operations.*
- **Migrate to a new environment.** *In an automated procedure, the functionality of existing software is completely or incrementally transferred to a modern development environment. Risks and costs can be significantly reduced. Over the course of transformation, quality improvements can be made through reengineering, or the structure can be modularized.*

#### Reasons to consider migrating to IBM Rational Business Developer software

As a modern target environment for critical business applications, Rational Business Developer software, which features Enterprise Generation Language (EGL), is particularly valuable. Its cost-effective EGL code generation automates the creation of repetitive and commonly needed application infrastructure

---

**Highlights**

---

***EGL helps facilitate the transformation of RPG applications by enabling developers to easily transfer their business knowledge in less time than it takes to transfer to Java technology.***

coding to help significantly reduce manually injected errors while helping to shorten the transformation process. EGL also combines the strengths of Java technology with high-performance native code engineering for System i and System z environments. Thus, the applications are designed to run optimally on nearly every platform, from Windows, Linux or UNIX to System i and System z.

Embedded in Eclipse, EGL offers a modern, service-oriented language for the efficient development of different application types, including Web programs, database applications, Web services, and batch and high-performance servers for fast transaction processing.

Thanks to the tightly knit integration in the IBM Rational tool chain, EGL development can be embedded in a professional development process or lifecycle – from requirements management, to modeling and coding, to testing and deployment.

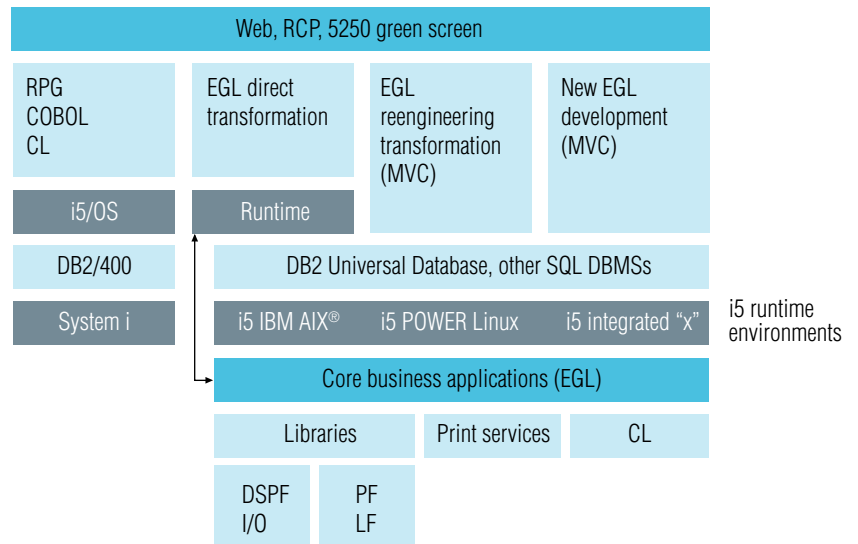
EGL's ease of adoption helps facilitate the transformation of RPG applications. Because EGL is easy to learn, it enables RPG developers to transfer their valuable business know-how in just a fraction of the time it would take for those same developers to transfer to Java technology.

**Highlights**

**Organizations can modernize and improve their software development practices and tools on nearly any platform by migrating RPG applications to EGL.**

**Potential application transformation scenarios**

Migrating RPG applications to EGL is a beneficial way for organizations to modernize and improve their current software development practices and tools on practically any platform. Here are a few of the most common scenarios.



System i application transformation scenarios

The application stays on the System i platform

If the migrated application is needed only on the System i platform, the existing RPG programs can be migrated to EGL in an easy, step-by-step process. Specific i5/OS functionalities like control language, commands, application programming interfaces (APIs) and system commands are often used in applications. These functionalities are still available posttransformation, making replacement unnecessary.

---

**Highlights**

---

***Through a two-phase approach, to transformation platform-specific functionalities and database access from RPG programs are included in EGL libraries.***

Transformation is typically completed in two phases. In the first phase, a server builder tool decouples the existing RPG application from the 5250 user interface. The RPG application then works with the API functions to directly access a Web-based or Windows user interface. The digital data storage source automatically creates the GUI using a GUI style sheet with more than 60 new functions. With very little effort, the application transforms into a server application featuring a GUI, and it no longer requires interactive performance.

In the second phase, the RPG programs successively migrate to EGL. The resulting EGL programs can use the same API for the user interface as the RPG programs. Thanks to its simple integration with RPG and the GUI, EGL can easily add new functionality, and new EGL programs can be developed directly within an SOA. Over time, programs requiring upgrades are regenerated in EGL, or the existing architecture is transformed into an SOA through refactoring.

The end result is that platform-specific functionalities and database access from RPG programs are encapsulated in EGL libraries. By separating the business logic and the platform-specific layer, application portability significantly increases.

The application is needed on a Windows, UNIX or Linux platform

Because EGL is platform independent, migrated RPG programs can potentially be used on other platforms. However, due to the detailed functionality of the i5/OS operating system, some specific features merit further consideration.

In Windows, UNIX and Linux environments, there is no standard database like there is in the i5/OS environment, which is why a database such as IBM DB2 software is needed.

---

**Highlights**

---

***If an application is needed on the System z platform, there are two approaches to migrating System i RPG applications to System z using EGL, each with varying levels of complexity.***

RPG applications using functionalities from the i5/OS operating system that are not available in the Windows, UNIX or Linux environments are extracted from the application and then made available in a special service library. This is done manually or semiautomatically through rule-based reengineering. Given that an RPG compiler is not available on these platforms, the application has to be completely migrated to EGL before it is executable. With the corresponding tools, existing data from the DB2 database in the AS/400 environment can automatically be transferred to the DB2 Universal Database environment.

The application is needed on the System z platform

With EGL, existing System i RPG applications are migrated to System z using one of two methods, each with significantly different levels of complexity.

The first approach consists of migrating the DB2 database in the AS/400 environment to the DB2 database in the System z environment. All RPG batch programs are also migrated to EGL. The job control language (JCL) from the System i platform is either manually or semiautomatically migrated to JCL via rule-based systems. Interactive RPG programs are then migrated via EGL to the Linux for System z platform, while also working with DB2 software on the IBM z/OS® platform. A Web interface is generated as the user interface, enabling the interactive EGL programs to run in the Java and Linux for System z environments.

The second approach anticipates the transformation of interactive and batch programs via EGL to the System z environment. To make the interactive programs capable of working with IBM CICS® software, it is necessary to reengineer the RPG/EGL programs so that a switch can be made from procedural architecture to a transaction-oriented architecture. This can be accomplished manually or semiautomatically through rule-based systems, resulting in a virtually pure System z application.

---

## Highlights

---

***By following a series of six steps using IBM Rational Business Developer software, businesses can help ensure a more efficient transformation of their RPG applications while realizing a variety of long-term advantages.***

### Conclusion

For companies that turn to IBM to mitigate the risks of modernizing their IBM System i RPG applications, a number of long-term benefits can be realized, including:

- *Automated code generation to help reduce manually driven errors.*
- *Reduced time and expense in maintaining applications.*
- *Lowered risk, based on the reuse and extension of proven functionalities and existing IT assets.*
- *Simplified SOA support.*
- *Improved flexibility and responsiveness from a unified team of developers who can be freely shifted across projects based on business demands.*

Modernizing an RPG application is an important strategic step toward maintaining a successful business outlook. As such, it requires careful preparation. Organizations can expect that the long-term efforts of numerous programmers will create a number of unanticipated glitches. To help ensure a smooth transformation, organizations can use IBM Rational Business Developer software to follow a series of six steps for success:

- *Build EGL skills and gain practical experience with EGL development.*
- *Convert an existing RPG program to EGL for closer examination.*
- *Conduct a comprehensive, detailed analysis of the existing application(s).*
- *Conduct a proof-of-concept study on a small part of an application.*
- *Transform the application with the help of experienced migration specialists from IBM and IBM Business Partners.*
- *Take advantage of the new EGL functionality.*



**For more information**

To learn more about IBM Rational Business Developer software, contact your IBM sales representative or IBM Business Partner, or visit:

[ibm.com/software/awdtools/developer/business](http://ibm.com/software/awdtools/developer/business)

© Copyright IBM Corporation 2008

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

Produced in the United States of America  
05-08  
All Rights Reserved.

AIX, AS/400, CICS, DB2, Universal Database, IBM, the IBM logo, i5/OS, Rational, System i, System z and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

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

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.