Software development and EGL
White paper
March 2008

Rational. software

IBM

Simplifying innovation with IBM Rational Business Developer—an executive overview.

Michelle A. Cordes, System z ecosystem team, Rational software, IBM Software Group

Stefano Sergi, product line manager, Rational software, IBM Software Group

Contents

Introduction

Businesses' efficiency and competitiveness continue to depend on their information systems. This trend is increasingly apparent in today's global economy, which is characterized by the constant search for cost optimization and aggressive penetration into new marketplaces via mergers, acquisitions, outsourcing and right-sourcing. The explosion of computer networks and the global reach of the Internet are transforming the way businesses operate and use information, and it has become apparent that traditional islands of application developers must be unified to help IT departments successfully support the business.

According to *CIO* magazine, integrating and enhancing existing systems and processes are at the top of many CIOs' 2008 agendas.[1] Therefore, to help transition away from traditional, siloed development models, many CIOs are embracing service orientation as the architectural underpinning for their enterprise modernization efforts.

Today's IT systems require software development capabilities that support new middleware and emerging application architectures. At the same time, development teams have grown accustomed to high levels of productivity and simplicity. To address these needs, IBM® has updated the capabilities of the IBM Rational® Software Delivery Platform with an integrated set of tools, methodologies and best practices that enable effective governance of the software development process, help accelerate modernization efforts and ease the transition to service-oriented architecture (SOA).

Highlights

The IBM Rational Software Delivery Platform features a set of design and construction capabilities delivered on top of the Eclipse platform.

IBM recognizes that each computing technology revolution challenges developers to keep up with the latest technologies.

Skills challenges for today's application developers

At the heart of the IBM Rational Software Delivery Platform is a set of design and construction capabilities delivered on top of the Eclipse workbench as powerful tools to help organize and speed the work of software architects and designers, as well as Web, Java™ and traditional developers. IBM recognizes that while Java is a powerful and popular technology, it requires developers to write code that works with its application programming interfaces (APIs), middleware and productivity-boosting frameworks, which can pose significant technical challenges that require a long and continuous learning process.

These skills issues are not new in the IT industry. With each computing technology revolution come new technical programming challenges that require developers to learn and code intricate systems services interfaces, instead of working on business functionality. For example, with the advent of online transaction processing (OLTP) in the early 1970s, application developers had to learn and master the services interfaces (typically COBOL or Assembler APIs) of OLTP middleware such as IBM CICS® or IBM IMS™ technology. And when relational data, graphical interfaces and client/server computing were introduced, developers had to learn and code the intricate interfaces of database middleware and GUI services middleware. Finally, with the explosion of application integration, application developers had to master a whole new set of skills relative to using the Application Integration Middleware (AIM) layers, from message-oriented middleware to adapter or broker APIs.

While in some cases it is essential to have complete access to the full systems interface capability, a significant portion of requirements can be addressed through typical usage patterns that can be abstracted to provide a simpler interface and shield the programmer from low-level coding. To continue the

Highlights

historical perspective, think of how capturing usage patterns of GUI widgets or relational data tables into visual tools, and language integration with such tools, brought significant levels of productivity into the creation of many desktop and client/server applications. While accomplishing more sophisticated or specialized requirements may have required developers to write C or C++ programs, the vast majority of the applications were written using these simpler and more abstract programming approaches, which were quicker and easier to learn, and more productive to use.

Why did IBM create EGL?

The EGL programming language is designed to be easy to learn and to enable developers to quickly become highly productive.

To address the skills challenges and other issues caused by the shift to SOA, IBM introduced the EGL programming language as an integral part of the Eclipse-based programming workbench of the IBM Rational Software Delivery Platform. EGL is designed to be easier to learn and more productive to use than other languages, complementing the breadth and depth of Java and COBOL technologies with a simplified and more abstract development paradigm that allows developers with a variety of skill sets to quickly deliver cross-platform, transactional, data-centric services and applications.

EGL is not intended as a substitute for the Java or COBOL language, but rather as a tool to help IT organizations:

EGL helps IT organizations take advantage of their business developers' existing skills and the power of their mainframe.

- *Leverage the knowledge of their business developers.*
- *Harness the power of their mainframe.*
- *Embrace and leverage the strengths of Java Platform, Enterprise Edition (Java EE) technology without forcing their entire team of developers to learn and master the lower-level intricacies and complexities of Java, or for that matter, COBOL and mainframe programming.*

Highlights

EGL is ideal for development teams that need high levels of productivity while delivering modern applications and services.

EGL helps developers leverage the benefits of mainframe and Java platforms without having to learn all the details of those environments.

While large percentages—and even 100 percent—of an application can be delivered using only EGL, there will be cases where additional code may be required or even appropriate for the requirements at hand. However, EGL is ideal for business-oriented development teams that:

- *Value ease of learning.*
- *Need high levels of productivity.*
- *Must deliver modern applications and services but cannot afford the cost, time or risk of transforming each developer into an expert in mainframe or Java development.*

A closer look at EGL

EGL is a modern programming language specifically designed to help business-oriented developers leverage the benefits of mainframe and Java platforms without having to learn all the details. Enabling your developers to focus on the business problem, EGL features high-level specifications that let developers quickly write fully functional applications and services. IBM Rational Business Developer software enables developers to write business logic in EGL source code and then generate Java or COBOL code along with the run-time artifacts required to deploy the application to the desired execution platform.

EGL hides the details of the deployment platform and associated middleware programming interfaces, which frees your developers to focus on the business problem rather than on the underlying implementation technologies. EGL is designed to help developers who have little or no experience with COBOL, mainframe, Java or Web technologies create enterprise-class services and applications quickly and easily.

Highlights

Developers can use Rational Business Developer software to build a number of applications to address your business requirements, including Web applications and portlets.

Because EGL does not provide low-level APIs to operating systems and subsystems, it is best suited for transactional data business services and applications development, rather than systems development.

What applications can developers build with Rational Business Developer?

The software is designed to address the full spectrum of business application requirements, including:

- ***Business services.*** *The language includes the built-in notion of "service," allowing developers to create and consume services in a simple and straightforward way, and to permeate systems architecture with service orientation.*
- ***Web applications.*** *EGL is tightly integrated with the JavaServer Faces (JSF) framework and JSF tooling, enabling programmers to easily create Web applications without needing to know Java or the details of the JSF framework.*
- ***Portlets.*** *Developers can deploy EGL JSF Web applications to the IBM WebSphere® Portal Server platform. EGL provides built-in functions to interface with the IBM WebSphere Portal framework, including support for interportlet communications.*
- ***Reports.*** *EGL integrates with Business Intelligence and Reporting Tools (BIRT), an open source reporting engine, allowing your team to create professional reports.*
- ***Batch systems.*** *The language includes the built-in notion of "batch program," which can be generated to run without end-user interaction, allowing your developers to produce reports or perform batch database loads or updates.*

Highlights

- ***Text user-interface applications.*** *To facilitate migration of legacy systems to a modern development environment, EGL enables your staff to create traditional character-based or green-screen user interfaces, such as 5250 and 3270, and relative programming constructs. This capability can also be used for new development, if needed.*
- ***Rich Internet applications.*** *EGL is also being extended to provide a simplified programming model to deliver applications with rich user interfaces that exploit the Ajax framework and popular Web 2.0 widgets.*

IBM Rational Business Developer enables you to transform EGL source into Java or COBOL sources that can be deployed to a number of different environments.

EGL applications target execution

IBM Rational Business Developer, Version 7.1 provides a state-of-the-art integrated development environment (IDE) for EGL. The application also provides a code generation engine that transforms EGL source for applications and services into either Java or COBOL source.

Generated COBOL code can be deployed to the following environments (for specific supported product versions, please consult the product literature):

- *IBM z/OS®: batch, CICS, IBM IMS/Transaction Manager (TM) or IBM WebSphere MQ Series support in CICS and batch applications*
- *IBM i5/OS®*

Generated Java code can be deployed to the following environments:

- *z/OS*
  - *IBM Software Development Kit for z/OS Java 2 Technology Edition for execution of Java programs under UNIX® System Services (USS)*
  - *IBM WebSphere Application Server for z/OS for execution of Web applications*

Highlights

Generated Java code can be deployed to the IBM z/OS, IBM i5/OS, IBM AIX, Linux, HP-UX and Solaris environments.

- *i5/OS*
  - *Java Development Kit (JDK) for execution of non-Java EE programs*
  - *WebSphere Application Server for execution of Web applications*
  - *i5/OS Integrated Application Server*

- *IBM AIX®, Microsoft® Windows® 2000, Microsoft Windows 2003, Microsoft Windows XP, Microsoft Windows Vista, Linux® (Red Hat, SUSE 9), HP-UX, and Sun Microsystems Solaris*
  - *JDK for execution of non-Java EE programs*
  - *WebSphere Application Server for execution of Web applications*
  - *Apache Tomcat for execution of Web applications*

Deploying EGL Java applications requires the installation of a set of run-time libraries (Java archive [JAR] files that are included as part of the Rational Business Developer, Version 7.1 product) in the target environment.

Deploying EGL COBOL applications for the IBM System i™ platform requires the installation of a set of run-time libraries (included as part of the Rational Business Developer, Version 7.1 application) in the target environment.

Deploying EGL COBOL applications for the IBM System z™ platform requires the installation of a set of run-time libraries (available in the IBM Rational COBOL Runtime for zSeries®, Version 6.0.1 product) in the target environment.

Highlights

EGL enables developers to gain access to a variety of databases without having to learn access programming technologies.

Supported databases

EGL provides simple and powerful language elements that can accelerate access to data stored in a variety of databases. Programmers no longer need to be proficient in structured query language (SQL) Data Manipulation Language (DML), Java Database Connectivity (JDBC) or other SQL access programming technologies; nor do they need to learn file or hierarchical database access methods. Instead, using a small set of polymorphic verbs, they can write full-function services accessing the following data stores:

- *IBM DB2®*
- *IBM Informix® Dynamic Server*
- *IBM IMS/DB (DL/I)*
- *Oracle*
- *SQL Server 2000*
- *Derby 10*
- *Virtual Storage Access Method (VSAM)*

What makes EGL powerful?

EGL empowers business-oriented developers to be extremely productive in a very short time via the following capabilities:

Abstraction capabilities in EGL eliminate tight couplings and reduce the amount of coding to interface systems and middleware, while declarative programming capabilities help reduce repetitive and error-prone coding.

- ***Abstraction.*** *EGL provides concise and powerful notations that eliminate tight coupling and reduce the amount of coding required to interface systems and middleware, simplifying and speeding developers' work.*
- ***Declarative programming.*** *EGL includes, where possible and appropriate, a certain number of declarative specifications to help reduce repetitive and error-prone coding. For example, by associating a validation rule to a data item, the validation is automatically applied and enforced practically every time the item is used in a certain context.*

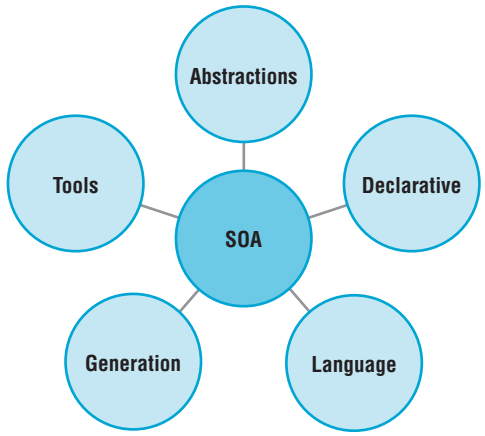
Highlights

The modern, modular language notation in EGL can help boost productivity for commonly required operations.

Rational Business Developer software includes tools built on the Eclipse IDE framework to help further developer productivity.

- ***Language.*** *EGL provides a comprehensive but easy-to-learn language notation that's modern and modular. And it includes a rich library of built-in functions to help boost your developers' productivity for commonly required operations, such as date and time, math and string manipulation. Additionally, the language is extensible and offers full interoperability with other languages, including EGL interfaces to native Java code.*
- ***Generation.*** *Although simplified, the EGL development technology can help ensure optimal deployment to the run-time platforms to take advantage of their qualities of service and to allow native management and monitoring of the systems in operation. A code generation engine included within Rational Business Developer transforms the EGL specification into native Java or COBOL source, and it also creates other required deployment artifacts.*
- ***Tools.*** *To further boost developers' productivity, Rational Business Developer includes a rich set of tools built on the Eclipse IDE framework. The set includes EGL source animation for debugging; powerful smart editing; visual construction; graphical navigation; tight integration of EGL notation with graphical Web development tooling; and automatic transformation of Unified Modeling Language (UML) models or database schemas into completely functional EGL services and applications.*
- ***SOA.*** *EGL has been designed from the ground up to facilitate services development and deployment. A simplified and abstracted SOA development paradigm has been built into the language itself, and it's complemented by tools and generation technology that are consistent with the basic tenets of the power of EGL. Rational Business Developer is designed to enable developers of different skill levels to easily create services without having to understand Web service protocols and standards.*

Highlights



EGL and Rational Business Developer software helps a broader class of developers deliver modern solutions and SOA, and simplifies legacy integration.

The value of EGL and Rational Business Developer software

Thanks to the power of EGL, organizations can deliver modern, powerful Web and SOA solutions and reap the following benefits:

- Increased programmer productivity
- A broader class of developers to deliver modern solutions and SOA
- A reduction in skills silos, and gains in flexibility and responsiveness
- Reduced training costs
- Reduced errors and increased product quality
- Simplified legacy integration
- Reduced development and maintenance costs
- Choices in deployment options
- Cross-platform solutions

Highlights

EGL can help organizations that run the IBM System i and IBM System z platforms and have siloed development teams.

EGL is designed to be used by any developer, which helps to unify your siloed teams and create a single pool of developers that works together.

Who can benefit from using EGL and Rational Business Developer software?

EGL is an optimal solution for any development organization that is starting new projects requiring Web development, integrating Web and mainframe applications, working on SOA projects, facing skills challenges and tight deadlines, or planning to modernize and improve its current software development practices and tools.

IBM System i and IBM System z users with siloed development teams

Organizations with in-house development teams often have developers who fall into two fairly distinct groups: traditional developers, such as RPG, COBOL and fourth-generation language (4GL) programmers, and new developers, such as GUI, Microsoft Windows, Java and Web language programmers. If your organization does a large amount of legacy development and there is mounting pressure to deliver new solutions to the business, EGL offers a number of benefits.

EGL helps unify siloed development teams, enabling you to make optimal use of your programming resources. Legacy developers have difficulty working with Web and SOA technology, and new developers are not able to deal with earlier-generation technology, so managing cross-technology teams can be challenging and inefficient. But all developers can easily learn EGL, creating a unified pool of specialists who can work on end-to-end projects in a single technology. EGL can also help motivate uninspired developers, retain existing talent and attract new talent, because it offers a modern, powerful programming environment that targets legacy environments.

Highlights

Organizations that use 4GL can leverage EGL to help extend the life of existing investments and optimize the value of 4GL developers.

EGL can be used to repurpose 4GL code via powerful migration and transformation tools.

Organizations with obsolete 4GL code (IBM and non-IBM)

Organizations currently using 4GL as well as those that have used it in the past may be interested in EGL for a number of reasons. First, such organizations typically value high-productivity programming or they would not have invested in 4GL in the first place. And if the 4GL is no longer capable of delivering value in the transformation to Web and SOA applications, either because the vendor has stabilized the products or because the requirements of the new systems exceed the 4GL capabilities, then EGL can be a good vehicle for extending the life of existing investments by leveraging the skills of 4GL developers.

Fourth-generation language developers can quickly learn EGL, using powerful migration and transformation tools and services to repurpose the old 4GL into EGL, breathing new life into valuable old code. Using tools included with the EGL development environment, the following 4GLs from IBM are easy to migrate to EGL:

- *IBM Informix 4GL (including migration of ACE reports)*
- *IBM VisualAge® Generator (including support for CICS, IMS, DL/I, 3270 applications, etc.)*
- *IBM Cross System Product*
- *IBM VisualGen®*

The following non-IBM 4GLs can be transformed into EGL using third-party tools and services:

- *Natural*
- *COOL:Gen, COOL:Enterprise and Ideal from Computer Associates*
- *HPSeer*

Highlights

If your developers have limited experience with Web and SOA application development, EGL can help them begin working with those technologies more easily.

Independent software vendors can leverage EGL to develop general-purpose business solutions to resell along with customization services to customers that prefer an off-the-shelf solution.

Organizations with developers inexperienced in Web applications or SOA

Organizations that are planning to develop Web and/or SOA applications but have limited or no experience in these areas should consider EGL—especially for Java, the Java EE platform and Ajax. Also, developers with limited or no object-oriented development expertise or those who cannot afford the time and costs of retraining—and are averse to contracting out the development to third parties because they want to avoid long-term dependency and costs—should take a closer look at EGL.

Regional and global systems integrators

EGL can be particularly valuable to systems integrators building custom business solutions for other companies. EGL can help increase their competitiveness by helping them reduce development costs, train new developers more quickly and establish a pool of developers who are flexible and not locked in to a single skill set.

Moreover, systems integrators are often involved in large rewrite or modernization projects, and EGL can become a significant advantage in their bids, especially when used in combination with automated transformation tools. Automated tools provide the ability to transform aging COBOL applications into EGL—a capability that's gaining the attention of several major systems integrators.

Independent software vendors

EGL can also help create a competitive advantage and a source of new marketplace opportunities for independent software vendors (ISVs). Unlike systems integrators, ISVs create general-purpose business solutions that they resell along with customization services to companies that prefer off-the-shelf solutions to custom solutions.

Highlights

EGL enables ISVs to introduce a great deal of platform flexibility into their solutions, which can help reduce development costs, speed new developer training and create a more flexible workforce.

Common questions about EGL and Rational Business Developer software

Q: Why haven't I heard of EGL?

IBM introduced EGL in the latest evolution of rapid development technologies currently used by thousands of companies worldwide.

A: IBM recently introduced EGL as the latest evolution of popular rapid development technologies currently in use by thousands of companies worldwide. IBM expects EGL to not only be the natural path for these users, but also grow significantly, given the interest in the marketplace.

Q: Why would I want to adopt a proprietary IBM language? Is EGL a strategic technology?

IBM plans to position the EGL core as an open language for program generation, allowing vendors, partners and customers to leverage the language's extensibility to meet their own objectives.

A: IBM is closely aligned with the dynamics and trends of the openness marketplace, and has been a powerful force behind open industry use of many of its core intellectual properties, such as Eclipse and the IBM Jazz™ platform. IBM designed EGL to be extensible and plans to position the EGL core as an open language for program generation, allowing third parties such as vendors, partners and customers to use the EGL extensibility model for their specific objectives. IBM has already defined the EGL specification and is in the process of socializing it with the Object Management Group (OMG). IBM is also exploring the possibility of providing an open source reference implementation of the EGL core specification.[2]

Highlights

There's a vast network of IBM Business Partners and consultants that can help you begin working with EGL.

Q: Where can I find skilled EGL developers?

A: There is an extensive network of consultants and IBM Business Partners that have the expertise you need to help you get your projects off the ground quickly. And don't forget that your developers can obtain the required skills in a matter of weeks, eliminating the need to compete for scarce resources.

Q: Code generators are great for productivity, but if something goes wrong in production, how will I be able to determine the problem if I don't understand the generated code?

By tracking back from generated code to the original EGL source, you can easily find problems and bugs.

A: The EGL generator allows developers to easily track back from generated code to the original EGL source. From there, developers can run debugging tools that step through the symbolic source while the generated code is executed, enabling them to easily find the problem.

Q: Code generators may be easier and quicker than hand coding in Java or COBOL language, but they always require a special run time. Can they achieve the scalability, availability and reliability offered by native servers?

A: Applications generated with EGL can deploy as native Java EE, CICS or IMS programs, exactly as you would deploy handwritten applications. The run-time libraries used by the generated code are simply packaged and commonly used auxiliary shared code. It wouldn't make sense to repeat these run-time libraries as generated code in each and every program, nor would doing so be in line with good software engineering practices that most companies already use in their own application architectures.

Highlights

It can be useful to look beyond IDE features when comparing different application development tools.

Examine each application's versatility, ability to deploy natively, legacy support, run time and development openness, and ease of learning before deciding which application development tool is right for your organization.

Summary

The application development tools marketplace is undergoing a profound transformation. Saturated with numerous programming languages and relative IDEs, the marketplace is brimming with technologies containing different tools, wizards, programmer shortcuts and aids such as smart editors, visual designers and project organizers. However, trying to draw a distinction between products based only on IDE features doesn't necessarily reveal a product's true worth. When using a comprehensive set of criteria, some of the most critical requirements weigh in favor of EGL when compared with the majority of the business-oriented high-level development technologies available today. Rational Business Developer not only has robust IDE power, it also takes full advantage of the mature and state-of-the-art Eclipse open source tooling framework.

Most organizations will agree that the following requirements are vital to supporting their application development objectives:

- ***Versatility.*** *Organizations need to be able to support different application requirements, such as desktop, Web, batch and SOA; different target platforms and operating systems; and different middleware, including databases, message-oriented middleware and transaction management servers.*
- ***Ability to deploy natively.*** *EGL enables your people to deploy executables without the need for special proprietary servers and leading scalable, robust enterprise transactional environments such as Java EE servers or CICS.*

Highlights

- ***Legacy support.*** *Organizations need to be able to easily integrate and use existing applications, or provide a way to transform and modernize older, valuable business systems.*
- ***Openness (run time).*** *EGL enables developers to create applications that interoperate with other systems through EGL's support for open standards, and they need to be able to participate in an SOA.*
- ***Openness (development).*** *EGL facilitates development success with its extensible language and development workbench.*
- ***Language.*** *EGL offers a high level of abstraction and productivity to developers, along with a robust and complete computational language.*
- ***Ease of learning.*** *Developers can quickly become fully productive with EGL.*
- ***Lifecycle integration.*** *EGL technology integrates easily within a robust lifecycle solution.*

EGL can deliver enterprise-class application and services development technology for business-oriented developers.

Easy to learn, easy to use, productive and robust, EGL offers enterprise-class application and services development technology for business-oriented developers. Because EGL has a great deal of intrinsic value, as reflected in the list of characteristics above, comparing it to tools based on low-level languages such as C, C++ or Java can be difficult. IBM is confident you will find that EGL will measure up against competing solutions.

Highlights

Visit IBM Web sites to find out more about IBM Rational Business Developer software, EGL and the complete set of IBM enterprise modernization solutions.

For more information

To learn more about IBM Rational Business Developer software or EGL, visit:

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

To learn more about the complete set of enterprise modernization offerings from IBM, visit:

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
03-08

---

1 *CIO,* “The State of the CIO 2008: The CIO’s Time to Shine,” December 10, 2007, http://www.cio.com/article/163700.

2 All statements regarding IBM future plans or intent are subject to change or withdrawal without notice and represent goals and objectives only.

RAW14096-USEN-00