Enterprise modernization
White paper
April 2008

Leveraging your Natural applications with IBM Rational software.

Edward J. Gondek, business unit executive, application modernization, Rational software, IBM Software Group

Contents

Unleashing the potential of legacy applications

In an age when speed and adaptability are crucial to staying competitive, today's businesses cannot afford to be held back by outdated technology. Many companies are looking to modernize their legacy systems—moving beyond the limits of the traditional application model to establish more open, flexible architectures. Enterprise modernization initiatives promise to unleash significant business potential, allowing companies to reuse application functionality for multiple purposes, to foster integration with partners and to create new channels for sales and customer service.

For companies that want to unleash the business potential of their Natural and ADABAS systems on the IBM® System z™ platform, IBM recommends a thorough, systematic approach to simplify the transformation. While this type of project can seem overwhelming, the cost-effective and highly automated solution described in this paper can help companies rejuvenate business-critical enterprise applications and establish a viable platform for the design and implementation of new applications.

The push to modernize

Nearly every enterprise has at least one cornerstone legacy application that runs core business functions, processes essential transactions and captures critical information. Legacy applications often represent decades of investment in software development and maintenance. Within their customized lines of code lie many of the company's best practices and knowledge capital. And their longevity is a testament to their value to the company.

Highlights

Legacy applications are increasingly difficult and expensive to maintain.

The legacy application, however, lives in a development environment that is not ideally suited to change. Because of this inflexibility, companies are finding it increasingly time consuming and expensive to modify and maintain their core business applications. To make matters worse, the number of programmers who are skilled in older technologies is dwindling. As a result, many companies find themselves limited in terms of new development. They may be unable to integrate with partner systems, for example, or to give employees central access to information. User interfaces may not live up to modern standards. And software development may be inefficient, with a significant amount of recoding required. Continuing to support these systems is simply unsustainable.

Enterprise modernization can help companies preserve valuable business functionality while taking advantage of more modern technologies.

For many companies, all of these factors produce a situation that has to change. They're facing an important challenge. On one hand, they need to modernize their development environments. On the other hand, they can't afford to lose the business functionality on which they still depend. The answer for a growing number of companies is enterprise modernization. It can help extend the viability of legacy applications, making them work in a modern IT environment—one that can support Web services, rich client interfaces and service-oriented architecture (SOA) models.

Enterprise modernization on the IBM System z platform

On the System z platform, it is common to find core business application systems built on Natural and ADABAS technologies. Many companies using these technologies are ready to move their applications to a more flexible, open foundation. To do this successfully, companies need a comprehensive approach that addresses the Natural applications, the ADABAS database management system and the data itself.

One such approach—and perhaps the most feasible one for System z customers—springs from the enterprise application transformation capabilities within the IBM Rational® Software Delivery Platform. The approach is called the Natural-to-EGL (Nat2EGL) solution, and it provides a structured, highly automated process for converting Natural code to open source EGL—the source code language for IBM Rational Business Developer tools. This process also involves migrating data from ADABAS to IBM DB2® information management software or to an Oracle database.

The Nat2EGL migration provides companies with a flexible development platform that can help fold existing applications into a more modern architecture while offering powerful tools for new application development. This modernization can put companies in a position to establish a Web services architecture, leveraging application assets in new ways. It is an ambitious objective, and, while

About EGL

Enterprise Generation Language (EGL) is a high-level programming language designed to allow developers to write enterprise-scale, full-function applications quickly and easily. Similar to Natural, EGL is easy for developers to learn, especially those who are well versed in fourth-generation language (4GL) functions. The preservation of these skills can be an invaluable asset for companies hoping to facilitate a smooth transition to more modern development technologies. Featured as part of the IBM Rational Business Developer platform, EGL is platform neutral, capable of generating run-time artifacts in COBOL or Java™ technology. It supports the latest standards in Java and Java Platform, Enterprise Edition (Java EE) technology as well as in Enterprise COBOL. This broad interoperability is critical when it comes to reusing legacy applications and extending their reach.

Highlights

The Nat2EGL solution follows a three-phase process to ensure careful planning and execution.

more than 95 percent of this transformation is automated, it is important to plan and execute the migration carefully. To understand how the Nat2EGL approach can help companies manage the complexity of enterprise modernization, consider the three main phases encompassed in Nat2EGL migration efforts:

1. *Discovery and analysis (known as DNA)*
2. *Code conversion and knowledge transfer*
3. *Implementation, testing and deployment*

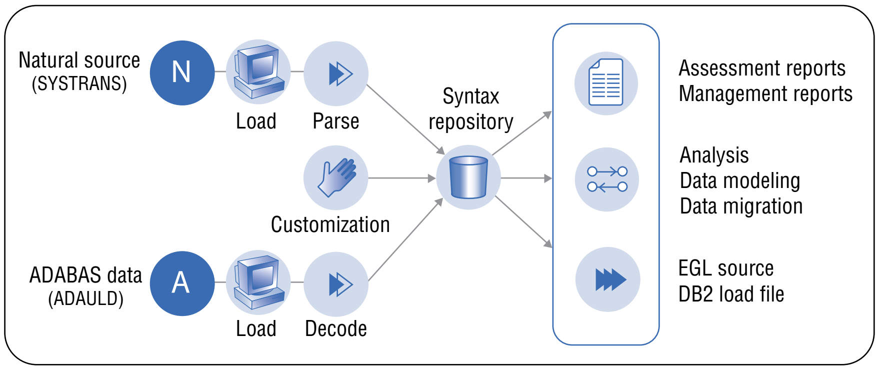

Figure 1: The Nat2EGL process

Understanding the Nat2EGL process

Phase 1 begins with a comprehensive discovery and analysis of the Natural environment.

IBM can help guide companies through the Nat2EGL process, from initial data gathering to execution and knowledge transfer.

Phase 1: discovery and analysis

The Nat2EGL process begins with a comprehensive discovery and analysis of existing source code and database files. Performing this analysis generates a significant amount of information about the Natural environment and helps identify areas in the source code that need to be cleansed. Once cleansed, the

Highlights

code runs through a final discovery and analysis process, thus setting the stage for the transformation process. It is at this point that IBM determines the scope of the conversion project, based on number of modules, lines of code and other factors. Phase 1 includes the following steps:

Phase 1 of the Nat2EGL process includes code preparation, partitioning, analysis, normalization and data conversion.

- ***Preparing the code and the data.*** *Document the entire Natural and ADABAS environment; identify missing or duplicate modules; examine the usage of data areas, maps, help routines, database access and more; and highlight areas that need manual remediation.*
- ***Partitioning.*** *Segment the application into logical units of work that will be tested and delivered incrementally based on priority and resource availability.*
- ***Analysis.*** *Parse and load Natural source code objects into the syntax repository; determine the feasibility of conversion; identify problematic code; and ensure the completeness and integrity of the code.*
- ***Normalization.*** *Generate the data definition language for the DB2 or Oracle database, and build relational tables and relational mapping files for the data extraction process.*
- ***Data conversion.*** *Map data into tables and fields that were defined during the normalization process.*

Phase 2: code conversion and knowledge transfer

Phase 2 includes code conversion, which is typically 90 to 100 percent automated.

Performed on an offsite, security-rich server, code conversion requires little contribution from the customer. Between 90 and 100 percent of the conversion can be automated, depending on the style and structure of the existing code.

During this phase, the Nat2EGL solution reads the Natural source code that has been loaded into the syntax repository and converts it to EGL, while migrating ADABAS files into a DB2 or Oracle normalized data structure. The solution

Highlights

automatically tags sections of code to highlight areas that may need remediation or review, so problems can be resolved before delivery. Upon delivery of code for installation, IBM can provide a package of EGL source objects designed to imitate the functionality of Natural application systems.

IBM provides education and knowledge transfer services to familiarize developers with EGL and the IBM Rational Software Delivery Platform.

Meanwhile, IBM can provide education and knowledge transfer, preparing developers to use EGL, the Rational Software Delivery Platform and IBM WebSphere® software—and, if necessary, educating database administrators on the use of IBM DB2 software.

Phase 3: implementation, testing and deployment

By far the longest phase of the Nat2EGL process, the final phase involves implementation, testing and deployment, which take place onsite at the client's location. The more resources the client dedicates to this phase of the process, the faster and more cost-effectively it can be completed.

Implementation, testing and deployment occur in the third and final phase.

Phase 3 begins with the implementation of the newly converted source code, database and EGL source objects. IBM can then provide documentation of syntactical problems with the source code that needs to be remediated before the application can compile and function properly. Once the company makes appropriate changes, it becomes possible to generate the resulting COBOL or Java run-time artifacts. The enterprise modernization process ends with functional testing, system testing, user testing, performance testing and, finally, deployment.

Moving forward with confidence

An enterprise modernization project can seem overwhelming. After all, most legacy applications are extremely large—some with hundreds of thousands or even millions of lines of code. Ensuring that each bit of code gets translated accurately can be an intimidating task. And rewriting all those lines of code is almost certainly out of the question. Simply replacing the applications with off-the-shelf software is not always an easy or cost-effective option either. Off-the-shelf packages typically require extensive customization to match the functionality of existing applications. But for many companies, enterprise modernization is quickly becoming a necessity, and those organizations need the best tools for the job. The key to doing it right is to use an automated approach that helps remove much of the potential for human error and helps support a rapid, cost-effective transformation.

The Nat2EGL process described in this paper provides just such an approach, allowing companies to continue to leverage their existing assets—including code, data structure and developers—as they modernize their IT environments.

For more information

To find out more about IBM Rational solutions for enterprise modernization on the IBM System z platform, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/rational/modernization

RAW14014-USEN-00