

FOUR HUNDRED *Stuff*

Hardware, Software & Services

REPRINTED FROM VOLUME 8, NUMBER 37 – October 14, 2008

PKS Provides the Missing Link from RPG to EGL

By Alex Woodie

It's been a little more than a year since IBM started promoting its previously mainframe-focused Enterprise Generation Language (EGL) as a way for System i shops to move to Java. Instead of writing RPG, the thinking goes, System i shops would develop in EGL, and then generate platform-neutral Java code. However, that plan did not address the problem of what to do with all that RPG code, and how to get started in EGL. Now, thanks to the work of PKS Software, IBM has a tool for converting RPG into EGL.

Since it was founded in 1988, PKS has specialized in developing application migration and code converter tools for IBM platforms, including AS/400, RS/6000, and S/390 servers, as well as Linux, Unix, Windows, and Java. Today, the company, which is based in Ravensburg, Germany, is very heavily focused on big IBM iron, and has more than 1,450 customers in 22 countries.

When IBM announced its EGL plans for System i customers last year, the folks at PKS recognized another opportunity to utilize its rules-based conversion technology as a way to move RPG code to EGL. PKS invested its time and money into supporting EGL with its conversion technology, and eventually succeeded in creating a conversion tool.

IBM took notice of the work, and made an offer to buy it under an original equipment manufacturer (OEM) license. PKS accepted, and IBM launched the tool as Rational Migration Extension for IBM i, or RMEi, which became available on June 24. See this IBM software announcement for more on IBM's RMEi offering.

Sparking Interest

Since it shipped four months ago, interest in the RMEi tool has been intense, according to Heidi Schmidt, sales director for PKS. "We're starting the first customers into prototyping phases, mainly in the U.S., but also Europe," she says. "We are already on the way in some accounts where Microsoft tried to migrate a customer away from System i and RPG to .NET. The customer really liked it, and was wondering, and also angry, why IBM was not able earlier to present such a story."

That story is a complex one that traverses people, applications, and technology. It starts with the RPG developers, who are getting older and harder to replace. Organizations are loathe to replace their RPG developers for fear that their RPG applications could be left stranded without a qualified caretaker, according to Schmidt. But at the same time, the organizations recognize the need to move their applications forward, and that means bringing in some other technology that is more suited to Web development than RPG.

Today, that other technology means Java. IBM, of course, has been pushing Java onto the System i crowd for more than a decade. For years, that push has been an abysmal failure, mainly due to the difficulty in moving RPG applications to Java, and converting RPG *people* into Java developers. But that resistance to Java is beginning to thaw.

In short, RPG and Java do not play well together. But owing to their heritage as procedural languages, RPG and EGL do work together. It's relatively easy for an RPG developer to move to EGL, the thinking goes--certainly it's much easier than retraining an RPG developer in the object-oriented world of Java. And once the developers are on EGL, they can move to Java--or whatever comes next--at their own pace.

The trick is getting from RPG to EGL, and that's where RMEi comes in.

"RMEi is architected as a file migration utility," Schmidt says. "So you do this migration from your RPG code one time, together with us or with the IBM Migration Factory." The Migration Factory is a services program designed to help organizations choose, plan, and implement migrations. PKS has paid lots of attention into insuring that RMEi generates EGL code that is tidy and maintainable, thereby providing EGL developers with a clean slate going forward.

Once the RPG code has been converted into EGL, it can be thrown away, Schmidt says. That's not to say that the organization ceases to develop in RPG. Organizations can take a piecemeal approach to migrating their RPG applications slowly over time, and can continue to write RPG that can be integrated into EGL through integration hooks that IBM has built into EGL. But once the decision has been made to move from RPG to EGL, the organization should be firmly committed to it.

Building Bridges

System i shops that subscribe to the EGL path must be ready to make considerable investments in EGL training for their developers. IBM offers a one-week introductory EGL course. Following that, students are encouraged to take another four-to-six week self-study course at home, according to Schmidt.

The ideal candidate for EGL--and, therefore, RMEi--is an organization that has a group of older RPG developers and a group of younger Java developers, Schmidt says. The older RPG crew is well-versed in the organization's business processes, but not so good at graphical interfaces or Web development. The Java crew is the opposite of that--they

know the latest Web 2.0 tricks, but don't have that body of business logic knowledge that takes years, if not decades, to develop. The least ideal candidate is an organization that has already made substantial investments in Java, and has stopped RPG development, or scaled it way back.

The combination of EGL and RMEi can bridge the gap between Java and RPG, Schmidt says. "With EGL you can bring them both together. The Java people can develop further on in Java, and in EGL the programs can be called directly. But the RPG people can move forward to EGL and do the development in EGL, and also--due to the code generation in EGL--generate Java applications. But they don't need necessarily to learn Java."

Without tools like EGL and RMEi, it could be very difficult to move from RPG to Java. "What IBM has recognized is it will not work at the end of the day, if you move directly from RPG to Java. It's impossible," Schmidt says. "The gap is too big, to move an application and also the people, from a monolithic, structural approach of software development to an object-oriented Java world. Therefore the Rational management decided that EGL is the perfect step to learn between RPG and Java."

(Interestingly, System i shops may also want to consider COBOL, which is the only other third-generation language generated by EGL. Not all of the performance issues of Java on the System i have been worked out, Schmidt says, and batch programs may run faster on the System i in COBOL than in Java.)

With tools like EGL and RMEi in place, the biggest challenges facing IBM don't concern technology, but marketing, according to Schmidt. "The main issue today is people are afraid if EGL is a strategic product," she says. "This is also the critical point, if a customer really believes this will be the strategic focus. We believe it; otherwise we would not have decided to invest such money in the migration technology."

However, IBM could do more to educate its own employees and the System i market at large that EGL will be around for the long haul, she says. "There's a statement from [IBM's software chief] Steve Mills that EGL is a strategic business language for System i and System z customers," Schmidt says. "But I think IBM must do more to convince customers that it really is a strategic business language."