

Rapid Web Development for System i.

The natural way
by
Brian W Kelly

Introduction

In 2001, SystemObjects introduced JACi400, bringing the advanced capabilities of its years of user interface work to the more than 20 million users of the IBM System i™, iSeries, and AS/400® business computing systems. SystemObjects is proud to be the only modernization tool vendor to provide a complete suite of development software devices for the highly popular System i operating system, i5/OS®.

In addition to the wizard-oriented development tooling, which is the main subject matter for this white paper, the JACi400 suite also provides for the conversion of thousands of green screen programs to the browser interface. The converted programs and more importantly the new wizard-developed programs are deployed in the same fashion with any of the new-age IBM WebSphere servers as well as the WebSphere Portal and WebSphere Express. Unlike other WebSphere applications methods such as WebFacing and those from the many vendors who market modernization tools, once you have deployed JACi400, you can put your WebSphere knowledge to sleep. You just won't need it -- and for most System i shops, that is good!

Application Modernization

Application modernization in most IT shops does not mean scrapping everything green. Were this the case, we'd soon be witnessing the total deforestation of IT and no one would have air to breathe. Instead, it relates to answering external and internal pressures to look good. No well-reasoned sane person would consider surgically removing the bulging part of their stomach by having a lumberjack use a chain saw. More than likely, the inevitable result would be something nasty (for example, death). Likewise, nobody would suggest cutting off the lifeblood of business functionality provided by those green screen applications that simply don't look good anymore and do not project a positive image for the firm. To excise them all and move to a browser-based environment with cascading style sheets full of flowers and candy is a certain death sentence. Note that does not mean that there are certain individuals in the firm or people doing business with the firm who would prefer the firm die than project such an "ugly" image. Beauty certainly is in the ... well, you know!

The smart CIO and smart CEO both want to have a "modern" system but more importantly, they want systems that support the business needs. If green screen systems were all about to disappear, these "C-level" executives would be in panic mode to assure that the business was able to survive tomorrow. The good news is the best C-level executives are not panicking, yet they realize that to the extent that it is good for the business, they must modernize. They don't necessarily have to take everything that is green and make it mauve or chartreuse or teal, but they know that their IT people need to know how to do so if they actually need to go "pink" one day in the immediate future.

So, what is the best application plan for the future? Were you to take a trip to any legitimate company website that focuses on application modernization, you would notice a common theme. The reason the theme is common is because it is able to be implemented. CEOs and CIOs really like things that can actually be implemented and realized. Unlike the big bang theorists who cannot stand green so much that they want the whole application set to be pink tomorrow, if your mission is the survival and prosperity of the firm as is the CEOs, you want to be assured that you can reach your goal.

For the CIO whose professional mission is to please the CEO, her or his personal mission is to be around another day to fight the battle -- for both the firm and the family. Yet though the job is important, the CIO is not about to undergo chainsaw belly surgery to make anybody happy. The smart CEOs and the smart CIOs are implementing one small project at a time and then the next and then the next and in between they are evaluating how well they are doing.

The JACi400 development environment does not care whether you like pink or you like green. It cares about whether you want to move forward or not. As you will see in the sections about today's IT landscape and the JACi400 development environment, there is only one effective way to eat an elephant...and that is... one bite at a time.

Too Much of a Good Thing

Five big questions.

In addition to the dilemma that CIOs and CEOs have regarding the absolute inability to redo everything "big bang style" with an intuitive graphical user interface, like it or not, there is no stable environment on which to attach -- even if this were a desirable approach. Look around and you will see that there is a continuing Web tools and protocols explosion in place right now. Tomorrow the names will be different but the notion will be the same. This leaves the C-level executives again at a loss for a clear solution. One might characterize this mostly as an adverbial quandary since **what**, **when**, **where**, **how**, and **who** (they already know **why**) are the big questions of the day.

Adding to the tool choice dilemma, the state of the Web itself finds users and builders demanding more and more richness from sites. Returning to the five "adverbs," consider how much richness can be added to the "best of class" sites (the very rich large businesses)? Then consider that what's good for best of class isn't always right and isn't always affordable for SMBs. The "best of class" sites have bigger pockets and can have many parallel efforts under way -- even some using beta level tools and concepts. I have heard some Web analysts smugly call those companies not fully meeting the "best of class" challenge as "Web laggards." This clearly separates most SMBs, including many System i shops from the "best of class" privileged user class. Whereas the "best of class" are examining the finer points of their many choices, the "laggards" are struggling to conduct real business as effectively as possible on their sites.

They are struggling because it is hard. It just is not easy. OK, it is plain old difficult. The Web is anything but a piece of cake no matter how many development dollars you have at your disposal. This is true especially for those companies who are accustomed to running small IT shops while producing high quality service and information outputs. These organizations are finding the nuances of the Web a very difficult chore to understand and manage. Unlike the early days of the interactive revolution when IBM's System/34, System/38, and System/36 ruled the day, there is no one way to the Web that capitalizes on the skills of today's System i community.

IBM's Developer Roadmap is Helpful.

IBM's developer roadmap (<http://www-03.ibm.com/systems/i/roadmap>), as helpful as it is, actually demonstrates the difficult choices that CIOs and CEOs must make regarding Web Development. The roadmap has multiple starting places and multiple destinations and thus, it is not at all like MapQuest in which you merely specify your originating address and your destination address and then just follow the directions. In the Web world, even to use IBM's map, you must first know where you "really" are; inevitably, this sounds lots easier than it actually is.

In order to engage IBM's Web development roadmap, you must first take the time to analyze where you are in a Web continuum of choices that are not necessarily intuitive. Yet, in using the roadmap, where you are is clearly the simplest thing to determine.

However, once you know where you are by finding your description in the roadmap, the task of determining your destination is even more difficult to accomplish. Of course JACi400 qualifies as a destination point in IBM's Roadmap, and it is a very good place to start your Web experience. But there are enough other destinations with little guidance to help make your map reading effort fruitful. In fact, there is no question that the Roadmap shows you just about every possible choice that IBM approves. But which choice is the best choice for your shop? Now, that is a different song, and there is no bouncing ball.

The list of possibilities is seemingly infinite because it actually is endless. Adding to the difficulty, a new product or Web protocol *du jour* is in one of various forms of industry evaluation / acceptance at all times. The list of possible tools that are not mentioned in IBM's roadmap is also substantial. You may not believe the list is so big until you see the partial list of the industry standard Web methodologies and protocols that are presented below. Consider that IBM has no space in its roadmap for other viable technologies that are used by System i shops. For example there is no space in the roadmap for CGI, the original path to the Web that is still used on many Web sites. Likewise, IBM's own CGIDEV2, a derivative of CGI with better ease-of-use characteristics, is nowhere to be found. Additionally, PHP, which IBM is clearly favoring today, is nowhere to be found in its Developer's Roadmap. One would conclude that IBM has concluded that to have a map that is reasonably understandable, not all destinations can fit in the space allotted. Yet, unfortunately for many, such a map does not take one to their destination.

Consider the recent harmonious relationship of IBM with ZEND, the inventors of PHP, and the ZEND PHP offerings. Though mentioned in IBM's Introductory material, PHP (PHP Hypertext Preprocessor), which exists within or functions as the principal engine for an estimated 33% of worldwide websites is not a destination of the IBM System i Developers Roadmap. The point is not that PHP is bad and thus excluded and it is not that you should run out and adopt PHP because it is so prevalent. PHP is in fact exceptionally popular with newer developers and college students for many reasons, not the least of which is that it is free and the coding is similar to the C/C++ coding found in many computer science curricula. The point is that the CEO / CIO tool decision is ginormous. How can a simple human being, overwhelmed with such a myriad of possibility permutations, make an objectively right choice?

A reasonable person would not be far from the full truth after concluding that there is actually way too much to know to make a fully informed Web development decision. Unless you pay for your own consultant, there is no proverbial spoon-feeder ready to guide you. However, you may find success by trying to find a fully independent body suggesting that you adopt a particular approach, a particular vendor, and a particular set of tools. Nobody can provide this advice at an independent level. First of all such ombudsman global bodies do not exist and if they did, for the most part, they would not

know all the tools themselves. Moreover, without a real effort, they would not know what was specifically best for you.

Ironically, IBM in the System i Web race, is somewhat of an independent body, though IBM's own solutions do come first in its roadmap -- even though IBM does not begin with A. All other solutions (and admittedly, they are not all there) are listed in alphabetical sequence by vendor. And, yes, in multiple destination categories in IBM's System i Developer Roadmap, you will find JACi400 along with other solutions provided by SystemObjects.

Internal Bias is Another Big Factor

The goals of a CIO and CEO choosing and investing in the right Web methodology are further complicated by internal biases. In many System i SMB shops for instance, a secondary IT department emerged over the years. This evolved primarily to deal with the desktop PC explosion and later the servers that came in to handle mail, static Web pages, and other one-of-a-kind applications. Therefore, a large number of System i shops that develop, maintain and run the core business activities also have server farms. The server farms address aspects of the business that long ago, the IBM AS/400 IT shop wanted nothing to do with. Though more and more companies have reigned in the two shops and combined them with common leadership, the cultures are very different and between the two it is often difficult to come up with a clear meeting of the minds to set a direction that can actually work for the organization. Bias finds its way into the process from the outset.

One group knows System i, i5/OS, RPG programming DB2 database (mostly DDS & not SQL), and of course interactive 24 X 80 column designed text-only applications that run the business (many home written), etc. The other group knows Windows/Linux/Unix, Visual BASIC or a "C" variety language, SQL Databases (no DDS) - MYSQL, Microsoft SQL etc, and of course the new darling, Microsoft Dot NET. Add to that GUI, & limited Web based applications that touch on ancillary needs of the business and you have the right picture. We would not be far off the mark to characterize this as a battle of form (Windows & GUI) vs. function (air-tight proven core business applications).

No Matter How You Cut it, It's a Tough Job

The dilemma of course today is that Web sites need not only great form and navigability, they need to deal with real business processes at the B to B and B to C level. Now, add to this pot of technical differences and perspectives, the amazing array of Internet tools available and you have a list that neither side can fully articulate or evaluate. Try finding someone from either side or even a consultant who understands this reasonably comprehensive starter set of the acronyms representing the tools of the trade for Web development

ADO, AJAX, Apache, Applet, AppML, ASP, CORBA, CSS, DCOM, DB2, Delphi, DHTML, DJANGO, DOM, DOT NET, DreamWeaver, DTD, Drupal, E4X, Eclipse, EJB, Flash, FrontPage, HATS, Hibernate, HTTP, HTML, IAWEB, IIS, J2EE, J2SE, Java, JAVAB, JavaBean, JavaScript, JBoss, JDBC, JMS, JOnAS, Joomla, JServ, JSF, JSP, LAMP, MamboMySQL, ODBC, OMG, PLONE, Portlet, PERL, Python, RDF, Roxen, RSS, RUBY, Schema, Servlet, SMIL, SOA, SOAP, Spring, Struts, SVG, Tomcat, Typo3, VBScript, WAMP, WAP, WDSC, WSDL, WebFace, Web-Service, WebSphere, WMLScript,

XAMPP, XForms, XHTML, XLink, XML, XPath, XPointer, XQuery, XSL, XML-DOM, XSL-FO, XSLT, Zend_Studio, ZOPE, etc.

And, no, you can't get by with just one of these technologies. You will always need to use them in combination. The approach for existing core applications surely will be different than the approach for the new edge applications which you would like to build to extend the core.

There is no denying that the Internet and especially the World Wide Web and the infamous Web browser have changed the very shape of what it means to create comprehensive business applications with centralized data repositories. Just like the brick and mortar counterparts, for most users, the Web is a store, a brokerage firm, a country auction, or simply a data-gathering tool. For more and more companies and many more to come, the Web has become their primary means of interacting with their full function business applications.

Both types of users find the experience incomplete in one way or another and that's why companies continually change the look and feel as well as the content and function of their Web sites. It's also why companies who just aren't there yet get a sense that their very businesses won't continue existing if they do not fix this problem and fix it fast. Whereas brick and mortar firms benefit from their proximity to their customers and rarely have to change the shape of their buildings to compete, Internet firms need a steady stream of architects, designers, re-designers, carpenters, electricians, masons, and of course plumbers. On the Web however, their dress is not blue collar and their functions are quite cerebral. Their tools are often very complex.

Underneath all the tools and functions lies a concept that in two words hits the nail on the head for those companies and organizations that have been enjoying the benefits of IT quietly behind the firewall. Those two words are "application modernization." Let's expand these two words to a few more words to better explain the notion.

Application modernization is an obligatory transformation for any company hoping to assure a competitive edge in its respective market through the long-term vitality of its information systems.

No organization can escape the word "obligatory." Though this modernization cannot occur with a big bang, it won't wait either. The notion predicates on a willingness to invest to gain the ability to adapt the company's core applications to new technologies, and open them up to new market standards, new opportunities, and ultimately new successes.

This is not entirely novel. It is no secret that behind the scenes, companies have always had to perpetually advance their applications to stay ahead of the opposition, without getting rid of their existing system (in most cases). The system already in place is the product of many evolutions and perhaps even a few minor revolutions in order to enforce the company's business rules. Though some may smear this ability to handle the company's integral core applications as "legacy," companies and organizations would quickly stop functioning if it were not for their in-place IT systems.

Even without the Web, no off-the-shelf standard cookie cutter package, irrespective of its quality, can replace the years of knowledge wrapped inside those internal systems. Moving to a browser-based system just because it is browser-based is a big net loss. Rewriting an entire system is always far too expensive and time consuming. Yet the status quo cannot be maintained indefinitely.

JACi400 is no reason to go anyplace else.

In this climate of sweeping change, change that is happening in nearly every area of business seemingly overnight, CEOs and CIOs need to know how to work with the new technologies while staying true to the spirit of existing "legacy" applications. Nothing in life worth having is easy. This is where SystemObjects and the JACi400 development environment come in. System Objects has made it about as easy as it can be.

Modernizing your newest System i applications means making technical choices that will still be around years down the road. Whatever strategy you need to implement, your tool choice needs to deliver your vibrant new applications quickly and bug free. They must provide the true look and feel of a modern application while using as much existing and well understood technology as possible. Familiar and time tested business notions such as the RPG language and the DB2/400 database are key elements in the System i modernization strategy so that your AD team does not have to start from scratch to rapidly move ahead. JACi400 is about as good as it gets to get your modernization efforts moving.

What is JACi400 Web Application Development?

JACi400 is a suite of application modernization tools designed to enable RPG and COBOL developers to replace the green screen interface with HTML/GUI. New applications are built for the Web using the best of the System i's innate facilities. No programming is required to develop JACi400 applications. The JACi400 Generator wizard grabs your hand and leads you through the process of selecting the database, setting up the application for inquiry, data entry, or update, finalizing your specifications, and deploying your application. And then, it actually works!

After the Wizard assures your responses, the generator does the rest. Besides getting an application deployed in minutes vs. days, the resultant HTML/AJAX and RPG code can be tailored using any industry standard tools including IBM's Rational family of which the System i WebSphere Development Studio Client is a member. Yes, you can even use SEU. Even better, COBOL shops get COBOL logic generated and RPG shops get RPG logic generated for the application. HTML/AJAX is a standard and is as leading edge as you can get today. Nobody needs to tell a System i developer that RPG and COBOL are the lifeblood of the System i shop.

But, what about if you want to change the application after it is generated? Do you need to regenerate it? You may but you certainly need not! In fact, unlike WebFaced applications, you don't even have to redeploy the application to the WebSphere Server. That's because the JACi400 deployment module, once installed under WebSphere, acts like it is part of the operating system. The bottom line is that you do not have to work with WebSphere tools to get your new applications or your modified applications up and running under WebSphere.

You make the HTML and RPG/COBOL changes if you wish after the fact, and JACi400 does the rest. Moreover, your WebSphere deployment choices include all of the players in the servlet server arena including the standard WebSphere Express, the full WebSphere, WebSphere Portal Server, or even the Apache Jakarta Tomcat Web applications server. In fact, the WebSphere you use does not even need to be the one on your System i.

For those System i developers who have been wanting to engage in the SQL revolution that is going on outside System i development, the generator produces SQL that works without bugs and thus can not only help in rapid application deployment but can also help teach you how to use SQL access within HLL programs.

One of the sharp criticisms of System i modernization tools is that the client is often left with little more than a dressed up 24 x 80 screen template regardless of how powerful the browser itself may be in extending the horizontal and vertical limits (and other limits) of the application look and feel. The JACi400 Generation Wizard has no such limitation. Moreover, the code that it creates has no such limitation. You get with this tool, all the Web has to offer.

The standard controls for the developer to use to specify data features are fashioned after SDA so there is no need to learn 40 tools in order to tailor the behavior of your generated Web page. You simply choose numeric, alpha, edit code, values, comparisons... etc. Without making this a technical paper for example, the generation wizard can even produce multi-tab displays and entry pages. Try doing that with a 5250.

Since the application "HTML" is really created with AJAX technology, there is a lot more that can be done in the browser to ensure the validity of the data as well as the user experience. For example, when users begin to enter information in the field of a Web application created with the generation wizard, they are prompted with suggestions for what they are looking for based on what they have typed, such as a certain customer name or a customer number. In this way, users can search for and locate specific

customer records in DB2/400 very quickly. The AJAX technology also is very handy for creating F4 "pop-up" screens. Note that this refers not to the annoying pop up that your browser does not permit anymore but pop ups that enhance the overall entry experience.

Anybody who has tried to modernize applications using IBM's WebFacing and even the HATS facility knows that enhanced 5250 display emulation in the browser is exactly that, display emulation. Sooner or later a sharp developer will ask what happens when a user wants to print something that internally prints very naturally on the local intranet printers. How is this done on the Web with WebFacing or HATS? The answer is that it is not done. These are display notions only. Yet admittedly many applications that you have in house or that you hope to deploy need secure printer output.

SystemObjects JACi400 Developer solves this problem by having anticipated it. The deployment module that is installed to WebSphere as a natural part of your JACi400 installation process also handles printing.

Inevitably with a generator product, there will be those who like using the wizard more than changing the code after the fact. Again, JACi400 anticipated that both types of developers exist. The typical System i RPG/COBOL developer wants control of their applications. Therefore to solve both types of development scenarios, SystemObjects devised a highly innovative technique in that it does not matter whether the code is developed from the generator or from the generator with subsequent modifications. The only price a shop pays for this facility is to use the unique "YOURCODE" tags whenever the code is YOURS. When you choose to regenerate an application for whatever reason, you do not have to reapply your changes. Your code is saved in case of a new generation via the YOURCODE tags. No fuss, no muss.

How many times have you been out to a Web site, booking a flight or a rent-a-car, and you experienced the handy dandy calendar object? Instead of having to get a paper calendar or use the calendar on your PC, the Generator has its own calendar object so that when you define a field as a date field, your Web users get to use the calendar object rather than having to type the date. That is a big hole in a converted 5250 application and the hole is patched with JACi400 so much so that the new road actually looks better than the original paving.

There are lots more items that SystemObjects anticipated you would need for your Web Development and the company has already included them in JACi400. If I told you them all, you would have less reason to go visit them at www.systemobjects.com. You will enjoy the trip to Paris.

Is this the natural browser based interface to RPG?

No, this is not the exact natural browser interface for RPG that I have been badgering IBM's System i and Rational executives to bring to the System i, but it just may be the best approach in the industry.

Unfortunately now that Rational has taken over responsibilities in Toronto, IBM has been very clear recently that this often requested, natural RPG Web interface is not coming. The latest from IBM appears that the company will ask RPG shops to write in yet another "new" language - EGL. At least it is not Java. But it is not RPG either. Historically, for the record EGL is a warmed-over 1980s mainframe code generator that had once been known as CSP.

Since mainframer's like COBOL, the language will produce COBOL and it will also produce Java code for those in love with Java. It does not apparently matter to IBM any more what System i programmers like. Of course IBM may argue that the Java generation output is in place because Big Blue has not gotten the message about how much System i shops like Java. One thing is for sure, RPG programmers have been forgotten again by IBM for there will be no EGL-generated RPG I, II, III, IV, or V.

In the latest pre-announcement rumblings, the search for a RPG Web solution is over. EGL is to be used instead of RPG on the Web and it will produce COBOL and Java code only -- not RPG and never RPG. IBM's coding evangelists say that shouldn't matter since Rational does not want anybody changing their generated code anyway. If that is really the case, couldn't they have saved even more development dollars by skipping COBOL and Java and going right to pure machine language?

For the embattled, long disenfranchised RPG shop looking for a simple way to the Web, the IBM Rational message is immediately understandable. Don't look to RPG, RPGIV, RPGV or any RPG language package for a lift to the Web.

RPG will never be capable of producing modern Web-based applications. This IBM decision assures that SystemObjects' decision to focus on the predominant System i language along with its unique invention of the "YOURCODE" tags, encouraging post generation code modifications, will continue to be unique.

Since most System i shops use the RPG language, SystemObjects solution is built for RPG developers. It is not a retrofit from the mainframe era or a novel computer science notion from Sun. It is designed specifically for the needs and wants of the System i (AS/400) community... the community that writes programs for business.

After all these years of promises and hints from Big Blue that it was finally going to do the right thing by enhancing RPG with a leading edge GUI facility, the deciders have decided. It is not going to happen. Thus, to move forward, developers must look outside of IBM for solutions. The SystemObjects team has the solution today. There is no longer any reason to wait for IBM to announce anything. IBM has spoken. The SystemObjects approach to getting your new applications to the Web with the JACi400 Generation Wizard would very well make an IBM natural RPG interface, if it ever was to arrive, the next best thing in Web development...

Best wishes in your Web implementation efforts.

Brian W. Kelly

About the author of this white paper:

Brian W. Kelly was an IBM Senior Systems Engineer (SE) for 30 years, and has spent over a decade as a System i consultant based in Wilkes-Barre / Scranton, Pennsylvania. He is also an author of dozens of AS/400, iSeries, and System i books and numerous articles. He serves as an assistant professor at Marywood University, which uses the OS/400 and i5/OS platform and teaches courses using the platform as well. Kelly is also one of the contributing technical authors to IT Jungle's "The Four Hundred" and "Four Hundred Guru" newsletters. Kelly has written extensively about the specifics of a natural GUI interface for System i and its development tools and compilers. He also provides counsel regarding the future shape of System i GUI facilities to IBM's Rational Development Lab team as well as to IBM's Systems Executives.