

Bringing Your iSeries Legacy Applications to the Web

by Heather Gately

Executive Summary

Explore the pros and cons of the four main options for taking your iSeries (AS/400) legacy applications to the Web: Screen-scrapers, In-house development, Consulting, Development Tools.

You are:

- A decision-maker at a company with an IT department of 15 or less.
- Relying on an AS/400 or iSeries midrange server to house data for one or more of the following departments: Accounting, Customer Service, or Production.
- Currently functioning in a green-screen environment for some or all of your business applications
- Looking at a very tight budget
- Searching for a way to move some or all of your AS/400 legacy applications to the Web and investigating your options.

If two or more of the above apply, this article was written for you. Read on...

This article will take you through the migration of iSeries (AS/400) legacy applications to a Web environment, the benefits of choosing this direction, and the pros and cons of the four most popular methods of bringing iSeries legacy applications to the Web.

If it ain't broke...

Let's say you have a working model T in your garage at home. It's a family heirloom that you've put a ton of money into, and spent many years tuning up and polishing. However, it's probably not what you use on your commute to work every day. It might get you there 9 out of 10 times, but not comfortably, quickly, or efficiently. Moreover, how many mechanics know their way around her engine? For daily wear and tear on this classic, can you imagine the expense of the maintenance?

Regrettably, when it comes to business, that's the current state of affairs in IT departments across the globe. Much time and money has been invested in expensive business machines bought years ago, and often the software running on them is completely unsupported, either written by defunct companies, or long-lost employees. Big businesses are being run on the backs of ancient machines, and there is a very noticeable shortage of mechanics.

Why not start anew?

According to the Gartner Group, between 60 and 80 percent of an average company's IT budget is spent on maintaining existing systems and the applications that run on them. So, why not just use that money toward brand new systems?

Darcy Fowkes, Research Director of Internet Business Practice, Aberdeen Group: "The fact that there is two trillion dollars' worth of mainframe applications in corporations today housing approximately 70 percent of all critical business logic and data make them valuable sources to leverage."

So, even if the IT world could make a decision on what all future technologies are going to be, and conform everyone to one path, it would take billions of dollars to recreate all of the applications. It's also taken years to get all the bugs out of the current applications. Starting over from scratch in a new environment would run the risk of major losses due to business discontinuity and customer frustrations.

If you haven't figured out the perfect solution yet, you're in good company. No matter how "ahead of the game" your company may appear to the outside world, technology is approaching such a pace that it is almost impossible to keep up. Somewhere, in some system, you are already out of date.

Get your priorities straight.

Since you can't win every battle, the only way to win the war is to do a proper evaluation and address what areas would best benefit your business to modernize. Here are a few common areas of scrutiny:

Production is big business' bread and butter...but inventory might be kept on various systems in different plants running a decade-old MRP software package that demands constant maintenance. As an example, let's say you have an application that has the ability to set a separate and unique price for each of your products for every single one of your clients. How much time would it take to fine-tune that application? How would you go about updating it efficiently while going on with your business without interruption? How much of your day-to-day applications are homegrown? What if the vendor that brought you your MRP a decade ago went out of business, unable to support its products? Where does that leave you? Is that your situation? Will it be tomorrow, or next year?

Or, maybe your main focus is not production, but instead you are working toward streamlining your accounting department. Perhaps comparing reports between different plant and office locations is completely apples to oranges. Reports in such situations are often written for disparate systems, by individual IT departments. To compound

the problem, the information might be two weeks old or more because of an IT application backlog. How stale are the numbers your decision-makers are basing their decisions on? Do all the necessary reports relate to one another? Are your company captains being forced to navigate by guesswork and old coordinates?

Then there's CRM. CRM, or Customer Relationship Management, may be the most-talked about buzz-term of the day, but the gist is age-old: it all comes down to customer service. If your customer service representatives are not knowledgeable, helpful, and efficient, your customers will go elsewhere, simple as that. Do you find that good Customer Service Representatives (CSRs) are hard to find, and even harder to train? CSRs often have to be trained in a text-based or green-screen environment that is non-intuitive and can take months to understand. Is that the case at your company? Even with a client/server solution, management often asks CSRs to navigate the ever growing, twisting and turning, forest of directory trees, extensive business rules, and query limitations. In addition, at the same time, try to relay information to the customer with a smile.

Even if you are only trying to answer one of the previous hypotheticals, it remains mission-critical to leverage your existing systems. Truly, your best bet is to modernize your legacy applications by bringing them to the Web. Browser-based applications present you with the most flexibility in distribution, data accuracy, training, and transitioning toward the future. Going to the Web can help your business grow, expand your client base, and improve customer relations.

Here are just some of the pluses to Web-enabling applications on your iSeries:

- You can serve your Web site directly from the iSeries or AS/400 that you've already invested in, so there is no added hardware cost.

- Distribution costs of mailing, faxing, and printing reports, invoices, bills of lading, and the like, will go down dramatically, and you'll experience a quick return on your investment.

- Business decisions will be based on accurate numbers; made from reports and queries that reflect live data, while remote users (e.g., salespersons in the field) can securely log on to access live data (inventory, pricing) from wherever they are.

- Your data and applications will remain secure because the iSeries was built with security as its #1 priority.

- Users and IT alike can focus on tasks that are more mission-critical by giving authorized users access to online reports and data instead of requiring IT developers to build individual applications.

- You will realize a rapid ROI by going to a "paperless" system, saving money across the board including: IT man hours, training for end-users and developers, lost orders, "temporary" software solutions, paper, ink, pre-printed forms, etc.

All right, already. What is the best way to take legacy apps to the Web?

Well, there is no silver bullet, but you have four basic choices if you are trying to bring your current iSeries 5250 apps into a browser-based environment. You can:

- elect to use a "Screen-scraping" software
- re-write all of the applications in-house in Java (or the like)
- hire outside consultants to re-write applications or customize package software for the Web
- purchase a software tool to modify the applications yourself.

Web > "e-business"

Evolution after the post-dot-com implosion

The term "Web" has evolved over the past few years. Not long ago, when the "dot-com" boom was busy booming, and venture capitalist millions were paying for SuperBowl ads for pet stores, all that anyone could talk about was "e-business."

Unfortunately, during the dot-com implosion, "e-business" became too closely tied to overnight millionaires, and man-child CEOs, and it got a bad rap along the way. The term became synonymous with crushed IPO expectations and fiduciary losses as even successful brick and mortar businesses fell for all the hype and threw a lot of money at e-business plans that could not succeed. Many companies are still licking their wounds from their own failed ventures. Many more have learned from their mistakes, understand the true business potential, and are now climbing slowly back up toward those e-business heights. The difference this time is they aren't hurtling themselves toward them in a rocket with no flight plan.

With the Web's maturity has come three clear Web venues, and as a corporate decision-maker, it's your job to determine which, or what combination of the three, would best suit your business plan. Let's start with the most famous of the three.

The Internet is defined as a worldwide interconnection of individual networks operated by government, industry, academia, and private parties. Its potential for your business is defined entirely by the nature of your company. It can be used for something as small as an online billboard, a way for your business to be found in a search engine search and contacted, or a simple, inexpensive way to extend your brand.

You can choose to include corporate information about your company; what you do, vendors you use, partners you have, and clients who recommend you. You can incorporate an online catalog to display your wares, or develop the traditional "e-business" features including an online shopping cart that allows users to charge their purchases to an authorized a credit card with unprecedented ease.

Additionally, you can include ways to mine for potential client data including forms for customer surveys, membership sign-ups, or contests. You can also add to your revenue by selling niche space to advertisers! Whatever you decide though, be sure to let your business plan be your guide. Know where you are going when you pull onto the Information superhighway. This isn't a road trip.

Another venue is the intranet. The intranet is a private network that uses some or all of the protocols of The Internet. The data being exchanged is controlled and displayed using a browser, but the emphasis here is on PRIVATE. The intranet may be connected to The Internet via firewalls, or it may be totally separate. It's entirely up to you.

"Web > e-business", Continued next page >

There are pros and cons to each, of course. Here are the basics...

Screen-Scrapers

"Screen-scrapers" are software products that allow PCs to intercept character-based data from a midrange server and present it in a graphical user interface. In basic terms, it automatically turns what looks like green text on a black screen into what looks similar to a Web page. Screen scrapers can often seem like the easiest way to convert legacy applications into a Graphic User Interface (GUI), and now, by extension, a Web application. The popular claim for screen-scrapers is that with the push of a button, the application is translated to an Internet Language, usually Java, and ready for the Web.

If it seems too good to be true, it usually is. Once purchased, developers often realize that it is necessary to go in and hand-code many features of the application in order for the screen-scrapers to work. But, by then, it's too late. Also, because they are traditionally written in Java applets, they require Java to be downloaded onto the PCs of those users that want to access the application. They are traditionally very Java-heavy which means that they tend to chug along at a snails' pace.

Common problems with screen scrapers include: server time-outs, browser-compatibility problems, software plug-in warnings, and complications with firewalls. They also don't look much like they belong on a Web page, and are not very intuitive in navigation because they are based entirely on the green-screen world.

The direct analogy to our earlier Model T example would be if you purchased the body of a Ford Explorer, and somehow secured it on the body of your Model T. It looks modern, but with the added weight of the new body, it's difficult to maneuver, and extremely slow.

Screen-scrapers are a good solution if you have an extremely simple application, are looking for a short-term solution until you are ready to scrap your legacy applications entirely, and

<Continued from previous page, "Web > e-business">

When the Web began, intranets began to be built because there was an emphasis on internal corporate information exchanges. The speed bumps that many early intranets hit were that there were too many chefs in the kitchen, and a lot of spoiled soup. In many cases, so many different departments were involved that the intranet was often just used as a corporate bulletin board with little organization or usefulness. They were either overloaded with outdated and poorly organized information or they were so underused that information was only sporadically added, and no one knew when to look for it...so they didn't.

Today's intranet is all grown up. Many companies are choosing to create employee Web portals with Web-enabled, interactive applications such as enterprise-critical reports, sales number graphs, data retrievals, inventory catalogs with pictures, pricing calculators, shipping labels, and other daily needs. Often reviled as an HR dumping ground in the past, intranets are now flexing their muscles as sleek profit centers.

Speaking of HR, intranets can also be properly stocked with employee applications to update health insurance, sign up for 401(k) s, register for training classes, or apply for tuition reimbursement. Saving time for employees and improving human resources productivity alike can save your business money. It all adds up.

Lastly, we come to the extranet... This is an intranet that has been extended to include access to or from selected external organizations such as customers or suppliers, but not the general public. Extranets are definitely proof of the evolution of the Web. These systems were just a twinkle in the eye when some of the original dot-com behemoths were cutting a swath of riches through Wall Street.

An extranet is a very useful tool, particularly if your bread and butter is in the B2B world. Some companies have created online delivery schedules to communicate to vendors when materials are scheduled to arrive in various plants and warehouses. Some have given their dealers and distributors access to their current open orders, maintenance of their account information, and even the ability to pay current invoices online. This ease-of-use has been proven to strengthen business relationships, and with the set-up of a few key applications, you can become the "go-to" company.

It's not just that "Web" got a bad rap, but that "e-business" also became too pigeonholed. e-business means more than choosing a book title, typing in a credit card, and having it shipped to you. It also means more than "click-throughs," page-views, hits, or traffic reports. A true e-business initiative should streamline all aspects of information exchange, and prioritize Web information centers as your business plan allows.

However you decide to use the Web: Internet, intranet, extranet, or all of the above; the flow of information and the pace of business is truly running at the speed of light. No matter what the size of your business is, harnessing that ever-flowing knowledge and power (even with something as small as a logo and a phone number on the Internet) is truly necessary to remain competitive in today's marketplace. How you choose to face the future and the next business revolution is up to you.

are prepared to spend in the neighborhood of \$100,000.

Rewrite applications in-house in a universal Web language like Java

Use your highly skilled IT staff to your advantage. After all, isn't that what you are paying them for? Why pay out extra money for contracting or hiring a team of Java developers, when you can train your current IT staff on a popular open source language, like Java, and get them working for you!

Upon first glance, this seems like perhaps the most suitable solution. After all, these people know these applications inside and out, many being built by their capable hands in RPG. And, learning a new skill set like this would be an absolute up-side to working all those late hours doing double-duty maintaining the current backlog of applications and reports while building all new applications and reports for the Web.

This is the model T equivalent to completely re-building the car yourself, creating certain parts by hand; from reconstructing the engine down to polishing the wheel hubs. It is a huge task to undertake, but it's entirely custom to your company's needs.

The downside falls in the category of time. According to the Butler Group², it is estimated that to retrain a developer in Java takes a minimum of nine months and has a significant failure rate. This is compounded by significant turnover when recently trained staff take their new skills elsewhere. The cost of this training needs to be taken into account, in addition to the expansion of the application backlog, the learning curve, development time, experience in taking a Java project from the early planning stages to a successful close within a reasonable timeframe, and the cost of staff turnover.

Also, keep in mind that the learning curve could be company-wide. According to another study reported by the Butler Group: "Further evidence is provided by the forecast that for over 65 percent of companies, their first generation e-business development work will be

scrapped and re-built within two years, often at a huge cost."

In-house Java coding is your best bet if you can say without a doubt that your IT people won't take the training and run, and they have extensive C++ or Java project management experience. Most important is if you can also say, without a doubt, that this intricately designed, expensive, "custom-built car" that you're aiming for is not on its way to obsolescence within two years.

Hire a consultant.

Now getting a consultant is not a bad idea at all. Yes, consultants will charge you more than a salaried employee for the privilege of contracting with you, but they ARE the experts specific to what you need. Would you want your general practitioner performing your brain surgery?

You just need to make sure you're hiring the right consultant. Consultants come in all shapes and sizes, and like surgeons; they sometimes have a bit of a God complex.

Just remember the following: if you go to the doctor to cure your cold, and he wants to cut out your brain, you might want a second opinion. Same with consultants.

There are several things to watch out for in working with consultants. The first is of course, references and expertise. It is your responsibility to know that they know what they are doing because you're letting them loose on the nerve center of your business. It's a dirty little secret in the consulting business that \$150/hr will often get you a fresh-faced college graduate who is "learning as he goes" on your dime.

Second, make sure that no matter whom you select, they stick to their schedule. In the old days, consultants would sit down with clients and charge high hourly rates to cover time and materials, but only for a limited period of time. In today's massive IT marketplace, "time-and-materials billing" can be a financial albatross when numerous consultants are working at a customer site on an "open-ended" project. These consultants are paid for every hour they put into a project, regardless of whether the project is

successful or on time. Make sure you cover yourself in your contract.

Finally, some corporate consultants are rewarded for selling follow-on work to customers so that once the company gets its foot in the door, it can stay there as long as possible. KPMG calls this its client-for-life strategy, according to Mark Lee, senior vice president of product solutions for KPMG in McLean, Virginia. (*Take Control of Your Consultants*, CIO Magazine, July 15, 2002).

Hiring IT consultants is the model T equivalent to hiring a team of expert mechanics to build that custom car just for you, and asking them to pick out the very best parts to use. It will likely cost a bundle, but that solution will get you to where you want to go, and you don't have to pay for internal training, worry about turnover, or take time away from your busy schedule. Of course, you have limited control over how long it will take, and since you only know the insides of your model T, do you feel comfortable questioning their methods? Or maintaining the final solution? Once it's completed, if you ever have a problem with it, who else will know how to fix it?

In that sense, some other vendor-based consultants, once they get their product's foot in the door, also try to "remain a part of their client's lives forever." They attach themselves to the wallet of a perfectly healthy corporation, and over time purposely create dependencies on certain products. Whenever a product needs to be updated, or a solution changed, that company's consultants are called in yet again.

If the product your consultant is recommending has the potential to create a real dependency on them, either through the amount of money spent (as in an ERP) or in the knowledge of a language, make sure you are making the right call before you sign on the dotted line.

Purchase an iSeries Development Tool

An iSeries development tool offers the best of both worlds by capitalizing on your staff's expertise, and at the same time speeding up the development process. There are many different things you need to

consider when looking at development tools: cost, ease of use, flexibility, and extensibility. When purchasing any Web development tool, the latter two become increasingly important.

Flexibility and extensibility are both critical when looking to the future. Where will you go, and how will you grow? Flexibility comes into play in terms of application deployment. Where will you be able to use your solution? What if you want your solution to be deployable to wireless devices? Would that ever be a need? These questions are important because as the technology evolution moves forward, it is important not to place these applications on a dead-end path, or they will have to be rebuilt yet again. Does your solution offer platform flexibility?

A good iSeries Web development tool should also build applications with growth in mind. This is something to take a good hard look at if you are looking for a long-term solution, and want to allow for your company to be global.

Three-Tier Architecture? What am I building, a wedding cake?

One recommended method of allowing growth in your systems is called "three-tier architecture." The original concept with roots in the days of client/server, had two components, and was therefore "two-tiered." This was the client/server version. The server was one tier (the business logic tier), and the client was the other (the graphic user interface tier). "Three-tier architecture" incorporates the first two-tiers and adds a third-tier (the database tier).

This architecture, in a nutshell, breaks Web applications into three components: a database component, a presentation component (GUI), and a business logic component and the advantages are many. To give you a better idea of how it works, let's say you wanted to develop a price listing for hotel rates that had prices varying from state to state, city to city.

Well, the designer could make that report look like the rest of your site in the presentation component. The database programmer would know which files to link in the database component. The business logic expert would program the pricing structure calculations into the business logic component.

If the pricing method changed next year to double the state tax as a resort fee in California and Massachusetts the business logic component would be updated without ever touching the other two components, and the application would remain seamless. The changes would not have to be duplicated for each state's application. Instead, they would be made in the business logic component which would in turn affect just the applications using that particular component. Or, perhaps the designer decides to add a blue bar across a page of the Web site. He could change only his component, and the changes would be made across the board. The same applies to the database component.

This "three-tier" architecture is ideal for maintaining your applications going forward. The set up eliminates errors, simplifies change, and saves time because each component is chaired by persons with expertise in that tier. It also allows for future changes such as new languages, and new looks, to be added, and can save you money on hardware.

In the past, a program was self-contained, and would need to be duplicated for each application, taking up space, and required to be on each machine where the application was running. Now, the components of an application could all reside on one iSeries, as in the past, or one component could be in a box in Arizona, one could be in Paris, and one could be in Seoul, and still work together. If the hotel headquarters in Seoul wanted its own presentation component (GUI) to allow for the Korean language version, and different database component with Korean tariffs, it could still use the same business logic component as its Arizona division, if it wanted. This method allows global possibilities.

Not all tools use three-tier architecture in their design. Make sure you look for one that does.

What kind of tool should I use?

There are many different iSeries Web development tools to choose from, and keep in mind that there is no perfect solution. But, using a development tool that will deploy to the platform of your choice puts the power and control back into the hands of your IT department without expecting them to create entirely new systems from scratch.

In the category of Web development tool products for the IBM iSeries, there seem to be two main software camps: products using vendor-specific languages, and products that are menu-driven that can be used without learning another software language. So, what are the pros and cons to each?

Pro-proprietary Tools

Those vendors who offer products developed in their own proprietary language, offer developers the bonus of putting another language on their resume. These languages are often challenging to learn, and in some circles can be a source of pride among programmers and a bartering chip in salary negotiations. For the vendor, it provides a viral marketing of sorts. If you, as a developer, spend 6-9 months in training to learn another language, and you move companies, you will likely encourage your new company to purchase the software tool that uses that language, and shows off your skill set. It's only human!

The downside to tools built in proprietary software languages, however, is that they again force the user to have a certain dependence on their vendor. Their time in training is invested, the amount of money they have in the software is invested, and the additional "specialized" modules they need to purchase for later projects or for later upgrades must come from the same company because the language is incompatible with other software pieces. These modules are also priced individually so any future needs that crop up that aren't covered may require additional system purchases.

To go back to our model T example, using a proprietary development software tool is deciding what you want in a new custom car, writing down how you want it

built in Swahili, which you will soon have to learn, and then having it built for you with car parts imported from Tanzania where Swahili is the primary language. This also leaves a question of maintenance. Whoever maintains your system will need to know Swahili as well.

If you want to handle the project internally, you enjoy the challenge of learning a new language, and you have the timeframe to account for staff training and learning curve, then this is the solution for you. However, you also have to be willing to accept that any expert-level work in that software will require an expert in that language, and if the software vendor goes out of business, you have no safety net, and will need to look for another solution. More and more companies using proprietary software are migrating toward open-source code solutions to prepare for the future.

Pro-Productivity Tools

The other tool option is to go with a specifications-based, menu-driven development tool. The downside to this option is that if you are a hard-core coder, you never really have a need to tinker with the code (although you often can). And, there is no proprietary language to learn, so no bartering chip in your salary negotiations, no gold star on your resume.

The plus side is that these software tools have an extremely short learning curve, with no languages to learn, so you can be up and running in a matter of days. Look for tools written in RPG/ILE, which is the native code of the iSeries, or open source code like Java servlets. Choosing a product like this means that if the software vendor ever goes out of business, any number of companies could support the software. It also shortens the project's time to market. Imagine being able to develop an application for your Web site in CGI/HTML, Java servlets, or XML simply by choosing something off a menu and hitting the enter button!

The Model T equivalent to this would be deciding what you want/need in your new car, choosing the features that would make up the car from a menu of options (in English)! And then, this custom car would automatically be assembled to your specifications and delivered directly to wherever you specify: to your driveway, your office, or your neighbor's

carport...wherever you would like it to go. (And you could also use it to build as many other cars as you want going forward!)

This is the best solution for you if you want to handle the project internally, avoid vendor dependencies or language-incompatibility in the future, and you want to get to the Web extremely quickly.

To sum up...

In summary, there are four basic methods to leverage your current legacy applications by bringing them to the Web. You can use a screen scraper method, you can build your solutions in-house, you can hire IT consultants to do the job for you, or you could use a tool to build your applications for you.

Whatever solution you choose, you should look for one that fits your budget, fits your schedule with training and learning-curves, is flexible in terms of

platforms, and allows for growth on a global scale (we recommend a three-tier architecture). You also need to consider whether or not you want to commit to a proprietary language tool, or if you'd feel more comfortable moving to a menu-driven tool written in either native code, or open source code.

To go back to the model T analogy, regardless, of which method you choose, ultimately, it is most important to know what direction you are headed toward, and why. Whether you are moving your legacy apps over to the Web because you would like to save money on maintenance, gain a competitive advantage over another company, improve internal communications, or increase employee productivity; make sure you have a clear goal in mind before entering into the Web world. The more you clarify your own needs, the more iSeries software vendors and consultants can present you with the clearest solutions for your business.

featured product

mrc-Productivity Series

mrc, maker of the mrc-Productivity Series and mrc-Lightning CE, has been helping developers create business solutions since 1981. The company specializes in accelerating the speed-to-market for Web and legacy applications alike.

The mrc-Productivity Series is a software development tool suite is among the most versatile, easily deploying to XML, CGI, HTML, and Java servlets. Simply by changing a selection from its menu, an application built with the mrc-Productivity Series, can be re-deployed to any of the above languages immediately. This is all without the user ever needing to learn a single line of code.

With its underlying three-tier architecture, the mrc-Productivity Series re-invents itself with every new technology that comes its way. It has a specifications-based, and menu-driven interface, giving it one of the shortest learning curves in the industry.

The mrc-Productivity Series is the most widely recognized force in bringing legacy applications to the Web. Staying ahead of the technology pace, and giving its users the ability to move forward with ease, is the mrc-Productivity Series' secret to longevity and the secret to mrc's clients' longevity as well.

For more information, visit www.mrc-productivity.com/products/ or call mrc at 630.916.0662.