



IBM i Modernization

The IBM i Challenge

White Paper

IBM i Modernization
The IBM i Challenge
White Paper

Copyright © 1982–2011 ASNA
All rights reserved.

Internet: <http://www.asna.com>

ASNA is a registered trademark of ASNA, Inc. All other product and company names mentioned herein are for identification purposes only and are the property of, and may be trademarks of, their respective owners.

Certain statements made in this document may constitute “forward looking statements.” Forwardlooking statements provide current expectations of future events based on certain assumptions and include any statement that does not directly relate to any historical or current fact. Words such as “anticipates,” “believes,” “expects,” “estimates,” “intends,” “plans,” “projects,” and similar expressions, may identify such forward looking statements.

Future plans for products may be discussed in this document. ASNA does not guarantee that any work discussed herein will be initiated or completed. Nothing in this document should be taken as an absolute direction of the company, but rather as plans that the company may or may not pursue in the future. ASNA has made every effort to ensure that the information contained in this document is accurate; however, there are no representations or warranties regarding this information, including warranties of merchantability or fitness for a particular purpose. ASNA assumes no responsibility for errors or omissions that may occur in this document. The information in this document is subject to change without prior notice and does not represent a commitment by ASNA or its representatives.

Contents

- The IBM I Golden Days1**
- The IBM i Challenge2**
 - ISVs Facing a New Reality2
 - Enterprise Concerns3
 - Call for Action3
- What Are the Alternatives?4**
 - Rewrite and Replace the Existing Application4
 - Scrape the Display to Provide a Web Interface5
 - Modernize Legacy Code to a Modern Language5
- Choosing the Right Alternative7**
- ASNA iBRIDGE—An Automated Modernization Solution to .NET8**
 - Why Modernize to .NET9
 - The .NET Initiative9
- Conclusion—ASNA iBRIDGE Technology Ties IT Together10**

The IBM i Golden Days

IBM i (formerly known as System I, i5, iSeries, AS/400, and Application System/400) is a midrange platform produced by IBM.

In the '90s the AS/400 platform became one of IBM's greatest success stories. A symbiotic relationship between platforms, vendors, and independent software vendors (ISVs) proved to be a critical success factor for both the platform and the application providers. The value propositions of the IBM i platform and the advantage over alternatives, led many enterprises and ISVs to develop vertical applications based on IBM i. Thousands of enterprises and ISVs put their trust and future in this environment. This was the golden age of the platform, and new vertical solutions were created one after the other.

Like other technology-based products, over time there was a decline in purchases of new IBM i boxes, as can be seen in the following diagram:

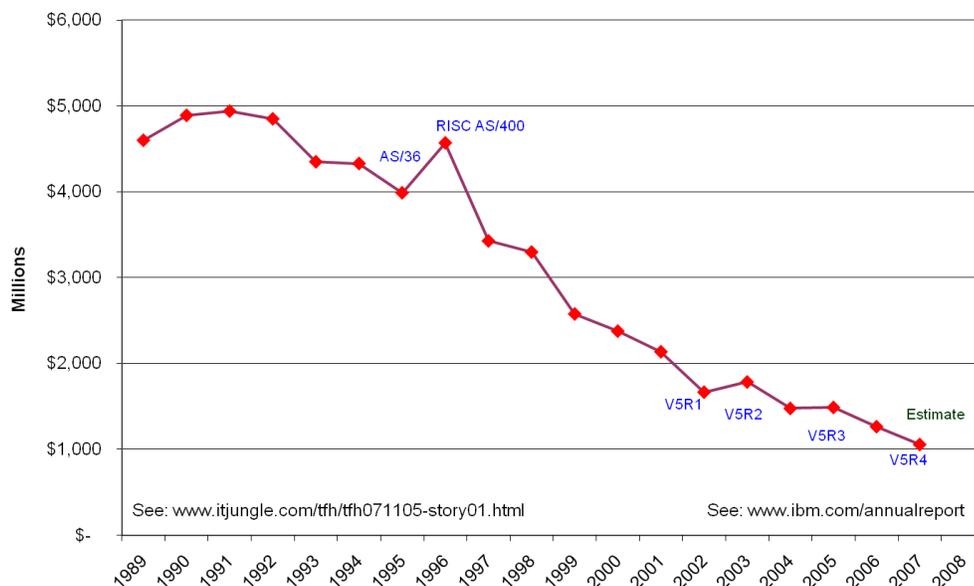


Figure 1. IBM i Purchase Decline

The decline in revenue reflects the difficulty IBM had trying to sell the IBM i line, which in turn caused IBM to change their product strategy regarding this platform. The IBM i division was eliminated, and the activity was split between two other divisions (the Power Systems unit and the Business Systems unit that focus on small- and medium-sized businesses).

http://search400.techtarget.com/news/article/0,289142,sid3_gci1264994,00.html

The IBM i Challenge

During the peak years of IBM i, the interdependency between the platform and the application vendors brought prosperity to both IBM and ISVs. Many enterprises relied on the benefits of the platform as well. Today, however, as the platform seems to be ending its life cycle, both ISVs and enterprises that rely on the platform are facing a real problem.

ISVs Facing a New Reality

Many ISVs with vertical solutions based on IBM i still produce an ongoing revenue stream from existing customers for maintenance and support, but there is hardly any new business. Moreover, there is a constant decrease in maintenance and support fees, because customers are moving to other solutions or demanding payment reductions. Some customers have concluded that the platform and applications are at the end of their life cycle, and simply refuse to pay.

Typical challenges that ISVs face are:

- **Selling to new customers is becoming more difficult.** As soon as a prospect understands that a IBM i box is required, the sales process becomes much more difficult. In addition, today's customers are demanding Web-based applications, and are unwilling to use green screen applications.
- **Lack of available team resources.** According to customers, there are very few green-screen RPG courses conducted by IBM. These courses are expensive and the number of people attending them is quite low. Veteran green-screen RPG programmers are retiring, and it is very hard to replace them or enlarge the development team.
- **Training new users on IBM i applications takes too long,** because users find it hard to cope with unfamiliar function buttons (rather than a mouse).
- **Adapting applications to meet customer demands takes too long.** Customers are accustomed to getting their requests met very quickly (with modern integrated development environments (IDEs) such as Visual Studio). Although RPG is considered a 3rd or even 4th generation language, compared to today's modern languages—not to mention their powerful IDEs—green-screen RPG is less productive and lacks a rich third-party source for additional tools and code libraries.
- **Existing customers are moving to new applications** on more modern platforms such as .NET, where there is a growing availability of packaged applications.

Enterprise Concerns

Enterprises that built their IT solutions around IBM i are facing difficulties in-house that are similar to those facing ISVs (especially the lack of available, trained resources); in addition, there are some specific concerns for those enterprises such as:

- **Hard to justify internally the high TCO of the legacy application.** As enterprises are trying to reduce operational costs, there is an ongoing effort to lower operational cost such as the IT total cost of ownership.
- **End users find the application “green screens” difficult to work with,** compared to other Web-based applications they are accustomed to using.
- **Existing IT staff,** especially the younger developers, want to get real experience in newer development environments.
- **Lack of agility,** when new functionality is needed to enhance the application.
- **Creation of sporadic solutions,** like screen scraping, increase the cost while leaving the legacy pains untouched.
- **Integration of business processes and data** is complex and expensive.

Call for Action

In recent years, IBM has tried various solutions to mitigate some of these challenges, such as pushing the ISV and enterprise community towards Java, EGL, screen scraping (with HATS), or even PHP. But these solutions did not provide the desired outcome, and it is now clear that while these solutions might buy some time, they do not solve the core problem.

The current situation calls for immediate action. ISVs and enterprises must recognize that it is time to move on, though it is in their interest to find ways to leverage their existing knowledge and investments. A proactive approach is needed to preserve these investments while creating a viable path to the future.

The competitive advantage of enterprises and application vendors is their expertise in a specific vertical market (manufacturing, energy, insurance, and so on). Over the years a great deal of knowledge was gathered into the software and keeping this knowledge is a significant need. This knowledge is the real added value that an ISV can offer to their customers and an IT department can contribute to the profitability of the business.

What Are the Alternatives?

There are three main alternatives:

- Rewrite and replace the existing application
- Scrape the display to provide a Web interface
- Modernize legacy code to a modern language

Each alternative has advantages and disadvantages. This section explores each alternative in order to help determine the best option.

Rewrite and Replace the Existing Application

This alternative involves creating detailed specifications based on the existing knowledge and the old application, and writing a new application from scratch using a modern technology.

Strengths:

- The application utilizes a modern platform and language with no legacy holdovers.
- A new user experience can be created, with improved workflow and interface.
- New required features can be planned and added to the application more easily.

Weaknesses:

- **Project time:** Creating a detailed specification, writing the code, and testing and debugging is a time-consuming process.
- **High risk for human error.**
- **Code and knowledge loss:** In a totally new development, all the proven code and the knowledge encapsulated in the application are lost.
- **Loss of team and accumulated experience:** Need to replace the development team with new programmers that have knowledge in the new technology. Losing the veteran developers is accompanied by the loss of the vertical knowledge they gathered over the years.
- **Retraining:** Existing customers who move to the new application will need to retrain their end users.

Scrape the Display to Provide a Web Interface

Wrapping an existing green-screen application in a Windows- or Web-based GUI provides a simplified experience for the end user, but leaves the application code and platform unchanged.

Strengths:

- Projects can typically be completed quickly.
- Continue to maintain and develop in the old and familiar environment.

Weaknesses:

- Solves only the interface problem, without addressing other challenges like high cost and lack of agility and available resources.
- This alternative does not solve the most important ISV challenge—selling to new customers. A new customer would need to buy a IBM i box, which is frequently a showstopper.
- This alternative doesn't solve the enterprise need to reduce the IT TCO.

Modernize Legacy Code to a Modern Language

Migrating source code to a modern object-oriented language such as Java, C#, or RPG for .NET is now a viable option for modernizing legacy applications. Tools and services are in place to automate much of this process. The resulting application can then be further enhanced by utilizing technologies inherent in the target platform, such as workflow, Web services, and modern user interfaces.

Strengths:

- This is a strategic approach to modernization that better positions the organization for the future.
- Automated tools remove much of the risk and virtually eliminate the unknowns from a project. The human error factor is removed from the risk equation.
- Preserve your past investments and the accumulated knowledge.
- Add new functionality in any programming language that is supported by the .NET platform.
- The existing team can continue maintaining the application, leveraging their vertical knowledge alongside the application structure know-how.

- Add new programmers to the team with knowledge in new programming languages. This enables an organic shift of knowledge from veteran programmers to new team members, with a built-in transition period. This method ensures the continuity of the software house.
- Easier to recruit new personnel.
- Easy path to GUI or Web-based interface.
- Open a simple path to the SOA world.
- Ease the sales process and end-user training.

Weaknesses:

- Source code must be available for the migration process.
- It might be necessary to modify the architecture to best utilize the targeted platform.
- At the first stage after the migration, the application is not object oriented. With time and modifications this can be improved.

Choosing the Right Alternative

Each alternative encapsulates a certain amount of risk. The risk rank (high to low) is as follows:

- Rewrite project
- Modernization project
- Screen scraping

Each of the alternatives has advantages over the other and disadvantages as shown in the previous chapter. As each enterprise and ISV is facing a different reality, all of the options should be considered and the best alternative for a specific case should be chosen.

ASNA has been part of the IBM Midrange community for the last 20 years, from the early days of System 36. We provide our customers all three options and we have experience with a wide variety of projects. Over the years we have discovered that in most cases the best solution for both organizations and ISVs is to adopt a modern environment like Microsoft .NET. This path provides the best value for current and future operations while minimizing risk to a reasonable level.

A successful modernization project is built on a clear and proven methodology. We have conducted many modernization projects and rely on our field-proven methodology.

This experience enabled us to offer a comprehensive solution called iBRIDGE.

ASNA iBRIDGE—An Automated Modernization Solution to .NET

ASNA offers an end-to-end service, called iBRIDGE, that extends and transforms your IBM i applications (originally written in ILE RPG or RPG/400) into native Microsoft .NET applications and your IBM i database to a Microsoft SQL Server database.

iBRIDGE comprises three elements:

- **ASNA Monarch®**—transforms iSeries applications originally written in ILE RPG or RPG/400 into native .NET applications. For a complete platform migration, Monarch migrates your iSeries database to SQL Server.
- **ASNA Visual RPG .NET®**—an add-on to the Microsoft Visual Studio development environment that enables your team to maintain and enhance the application after the modernization in the .NET environment, while continuing to use the familiar and powerful RPG syntax. It not only embraces the original data processing foundation of the RPG language, it is a fully object-oriented language.
- **ASNA DataGate®**—a middleware product that enables the .NET application to natively communicate with the IBM i database and alternatively with SQL Server (without any code changes), and therefore enables staged modernization.

With the above components, the IT team has all the needed tools and resources to execute a successful modernization initiative.

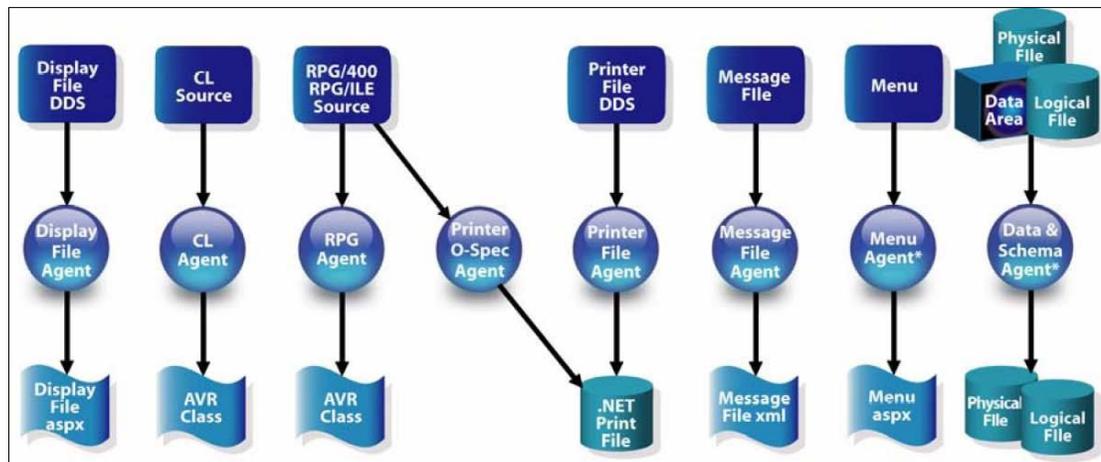


Figure 2. Migration Agents

Why Modernize to .NET

Simply put, leveraging your organization's investment in legacy systems and augmenting them with modern solutions increases the productivity and responsiveness of your IT organization while providing important options for shifting your expenditures from maintenance to advancement.

Modernization begins with the extension of your legacy applications using modern platforms. The rewards of this straightforward goal are many:

- Is a comprehensive, cost-effective, and safe way to preserve critical elements of the investment in your legacy assets
- Mitigates the costs of maintaining systems in their current state
- Provides the right tools for the right tasks
- Capitalizes on your human resources and extends their capability, productivity, and accomplishments
- Avoids the business impact of switching to an entirely new environment
- Can be implemented one step at a time, even starting simply and growing in depth and breadth, thus returning the management of evolution to the organization

Modernization may be implemented in a variety of ways, and frequently each implementation sets the stage for the next growth step. Modernization is not limited only to business applications, benefit, competitiveness, and bottom line. Your developers become modernized as well. They can forge ahead into all the new technologies with greater ability, capability, and productivity. Moreover, they hit the ground running. They are a known asset to your business, and, above all, they know your business.

The .NET Initiative

Through Microsoft's .NET initiative, companies are adding business value overall, and specifically with Web services—building, deploying, operating, integrating, aggregating, and consuming them.

Adoption of .NET as an architecture for “service oriented architectures” allows your organization to address the requirement to increase your business agility, evolving from a world where solutions are standalone nodes—whether applications, devices, systems, services, sites—to a world where solutions are constellations of multiple applications, devices, and services working together.

Microsoft is providing best-of-breed development tools with Visual Studio .NET and the .NET Framework. The .NET development model embraces many programming languages, so that developers can use whichever language is most appropriate for them and for each project.

Conclusion—ASNA iBRIDGE Technology Ties IT Together

Evolution or revolution, extension or migration: whichever strategy you choose, ASNA works with you to leverage your legacy portfolio into a technology infrastructure that adapts to the on-demand needs of your business and promises to strategically leverage your technology horsepower with new Internet standards to bring your business services online.

ASNA iBRIDGE combines IBM i, Microsoft .NET, and your legacy portfolio into a formidable engine to power your IT modernization, moving it forward rapidly and affordably.

The ASNA-Microsoft-IBM amalgam provides a virtually risk-free modernization architecture. ASNA's iBRIDGE offering is a *well-proven* solution for companies and ISVs with IBM i systems who must find a way forward to support the evolution of their people, processes, applications, and data to service oriented architectures.

With iBRIDGE, modernization can be conducted as a single project, modernizing the application and the database together; or, in a staged approach, beginning with the application, keeping the database on the IBM i, and completing the modernization of the database at a later time, according to the business needs.

ASNA, is the world leader in legacy modernization. Through hundreds of modernization projects, we have established beyond any doubt that modernizing IBM i applications into a modern programming language is the right modernization path for legacy applications.

For IBM i ISVs, modernizing your application to the .NET platform will enable you to open new markets, significantly increase your solution's prospect and customer base, and create a clear road for future prosperity.

For enterprises using IBM i applications, modernizing your application will insure the continuity of your IT system and will assist the organization to creation competitive advantages over competition.

About ASNA

ASNA provides a proven solution suite of flexible products and services allowing its clients to gradually or in totality modernize, integrate, migrate, and extend IBM i assets to the .NET and/or SQL Server standard at a reasonable cost and in orders of magnitude faster than rewriting or replacing with a package. These solutions result in a safe, risk limiting extension to the life and value of the IT assets while achieving an extraordinary ROA (Return on Assets). ASNA's experience since 1982 with singular focus on the midrange market, coupled with its automated tools and abundant successful modernizations puts ASNA in the undisputable position as market leader in the midrange modernization and migration space.



ASNA

ContactSales@asna.com

www.asna.com