MODERNSYSTEMS

WHITEPAPER

WHEN CA GEN ISN'T COOL ANYMORE

THE BUSINESS CASE, CHALLENGES AND SOLUTION FOR MOVING CA GEN APPLICATIONS TO A MODERN PLATFORM

Introduction

CA GEN applications are usually of considerable size and accumulate complex business logic. A lack of skilled resources, limited integration and high costs create challenges around maintaining CA GEN. This whitepaper covers the current state of CA GEN in the enterprise and the most efficient, low-risk way to modernize.

Special Offer

Modern Solutions has chosen to extend a special offer to audiences of this whitepaper. If you have CA GEN systems in your environment, you are eligible for a complimentary one hour consultation to determine if your system is an ideal fit for our translation technology. This solution:

- Reduces costs by reducing or eliminating software licensing costs, deploying to lower cost environments, and eliminating duplicate development tools and processes
- Automates the conversion of legacy code and data to a modern language, development tool set, and database -- low risk, highly maintainable, flexible
- Increases responsiveness and time to market by reducing development time
- Enables reuse of proven, tested applications and enables deployment to new platforms and environments

If you are interested in this offer, email us at

CAGEN@MODERNSYSTEMS.COM.

Contents

> **“Modern Systems worked closely with our teams from the inception of the project through the sizing and planning to the delivery. The project has been completed within timescale and budget and this enables us to move onto the next stage of our IT strategy.”**
>
> **David Loughenbury, CIO**
> **Police Mutual Assurance Society, UK**

Business Case: Why Move?

Total Cost Of System Ownership

The ability to move to a lower cost platform can greatly impact IT's bottom line, generating flexibility to add resources to other critical areas. A recent study by HP, Micro Focus, Microsoft, and Unisys Corporation estimates that the equivalent of 1,715 MIPS could run on one 8-CPU Intel Xeon enterprise-class system. The difference in cost between the two platforms is an example of how customers who have migrated mainframe systems report significant cost savings.

While CA GEN applications generate significant tangible costs like hardware and licensing, the impact on internal resources can have a higher strategic cost. Most organizations with mainframe-based systems are spending about 75 percent of their development resources simply to maintain existing applications, leaving only 25 percent of their development resources with time to innovate on ways to deliver value to the business.

Effecting a 25-percent reduction in maintenance costs means that investment in business-driven IT advancements can be almost doubled.

Feature Freeze

Many companies are running mission-critical applications on legacy systems that do not have a roadmap for signifcant feature improvement. In the case of CA GEN, no significant features have been added in over 10 years. Unlike modern platforms, CA GEN's development roadmap doesn't accurately reflect the value of the data stored on the platform.

> **"One of the greatest growing risks for these (legacy) systems is not the applications themselves, but rather the skills necessary to continue to develop, maintain, and operate them."**
>
> Gartner®
>
> *Dale Vecchio. "Impact of Generational IT Skills Shift on Legacy Applications."*

Diminishing Skills and Resources

For legacy languages such as CA GEN, programmers are becoming increasingly difficult to find. The people who know mainframe technology are steadily retiring. Most universities no longer offer mainframe instruction. As the shortage of experienced programmers becomes worse, the cost of these resources will continue to rise.

Difficulty Responding To Competitive Pressure

Lack of frameworks, productive and advanced IDEs, debugging tools and test automation add significant time to development cycles on the mainframe. Organizations relying on legacy systems have a high time-to-market for new business needs and respond slowly to challenges from competitors.

Even if new applications are developed using modern technologies, integrating these with core business functionality running on legacy systems is a time intensive and risky task.

Business Case: What Are Your Choices?

Maintain Status Quo

The first choice for any scenario is always to do nothing, to “let it play out”. As time passes, dealing with GEN legacy is more and more difficult – less available expertise, more application development backlog and more money thrown away for GEN licensing that could be spent on projects that improve IT efficiency.

Third-Party “Off The Shelf” Solutions

This approach focuses on replacing mainframe application functionality with packages and components available from third party vendors. The positives of this approach include reduced maintenance of source code, as the vendor is responsible for fixing production bugs and implementing new functional enhancements.

However, commercial packages offer standard domain business processes that often differ from the homegrown mainframe application. Reuse of existing business logic isn’t possible, so some level of business process re-engineering or customization/rewriting of the third party solution is required. Both of these processes can be time consuming and expensive.

Full Re-engineering

Known as the “big bang approach”, re-engineering is the most expensive and risky solution for CA GEN modernization. It includes requirement capturing, coding, debugging, testing and refining. To recreate the legacy solution’s wealth of functionality, with newly written application stable and error-free to the level of GEN applications requires significant time and effort. Impact on end users and their adjustment cycle is also a major factor. The cost, time and risk involved with re-engineering are so high that not many organizations choose to embark on large-scale rewrite projects.

Application and Database Conversion

Modern Systems’ technology is the core of this approach, which guarantees like-for-like functionality with the legacy environment while empowering customers to leverage the advantages of newer platforms.

Our solution catalogs application code as stored in GEN Encyclopedia and automatically converts it to pure Java. The resulting Java code is completely functionally equivalent to the original GEN application. No single business rule is lost or distorted. The resulting Java code is as clean, readable, and maintainable as the original GEN application. Use it with mainstream development tools such as Eclipse or Visual Studio and achieve the same level of productivity enjoyed with GEN, while not depending on any third-party code or licensing, and with unlimited pool of Java or C# developers.

Technical Overview: The Modern Systems Solution

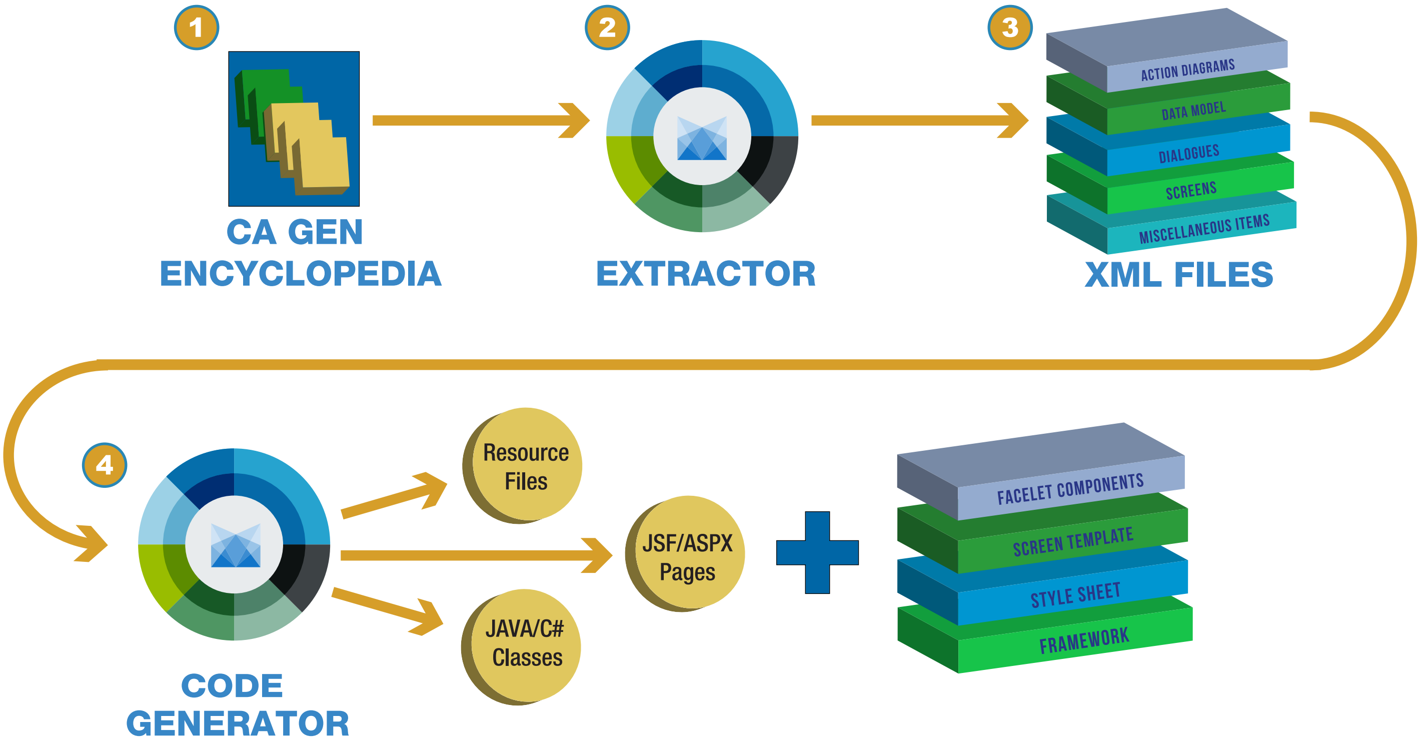
Modern Systems' conversion technology automatically converts existing CA GEN applications to fully maintainable and easily extendable Java or C# code.

We start by using CA GEN encyclopedia as input, not the CA GEN generated code. This ensures the converted code is compact, clean and easy to maintain. We offer flexibility around the deployment configuration- whatever it is, we can automatically convert it. Batch, 3270 blockmode, CICS, IMS DC, client/server, GUI, proxies – each of these approaches will work with our solution. We handle COBOL External Action Blocks (EABs) automatically, converting them to the same target (Java or C#) together with CA GEN. We can handle batch job JCL by automatically converting job decks to scripts on the target platform.

Customers who prefer to keep their database in the same place after the transition can do so, or we can move the database along with the converted application to the new platform. Our goal is to empower the customer to make the choice that's right for them- and provide a flexible environment for doing so.

The converted application is architected conforming to object-oriented Web-centric requirements, running on any Java application server (WebSphere, Weblogic, Apache Tomcat) or on Microsoft IIS if C# is the target language. The converted code follows the same component and naming structure as the original application, adapted to the Java/C# requirements. Even comments are preserved. The goal is for the customer's development team to look at the converted code and immediately recognize familiar business logic. Only basic Java/C# training is required to easily navigate the newly produced code.

Overview of Translation Process

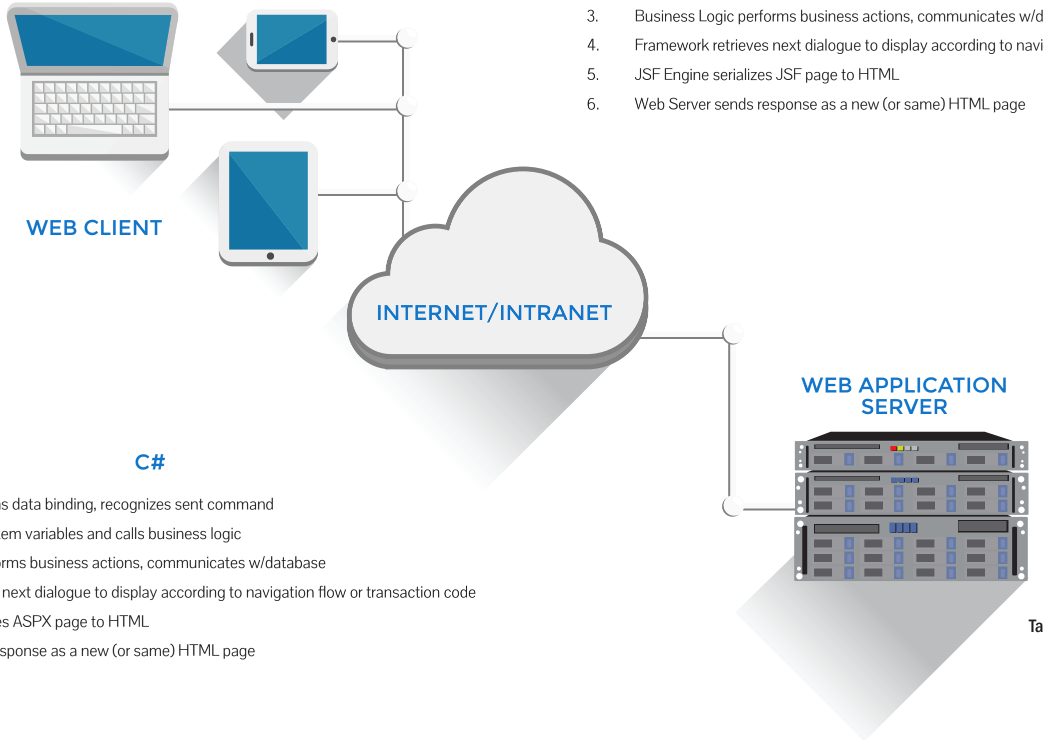


Target Architecture: Pure Java or Pure .NET

Modern Systems' automated conversion technology empowers the customer to leverage the target state that's best for them. This example depicts the highlights of a pure Java target state as well as a pure .NET target.

JAVA

1. JSF Engine performs data binding, recognizes sent command
2. Framework sets system variables and calls business logic
3. Business Logic performs business actions, communicates w/database
4. Framework retrieves next dialogue to display according to navigation flow or transaction code
5. JSF Engine serializes JSF page to HTML
6. Web Server sends response as a new (or same) HTML page



Target State Architecture

C#

1. .NET Engine performs data binding, recognizes sent command
2. Framework sets system variables and calls business logic
3. Business Logic performs business actions, communicates w/database
4. Framework retrieves next dialogue to display according to navigation flow or transaction code
5. .NET Engine serializes ASPX page to HTML
6. Web Server sends response as a new (or same) HTML page

Real World Examples

Chicago Mercantile Exchange (CME)

Chicago Mercantile Exchange initiated a department-wide legacy modernization effort. CME chose Modern Systems to help upgrade their settlement, banking and asset systems. Modern Systems' automated conversion technology enabled CEM to move to a new Linux platform, significantly reducing operational costs and improving application development agility.

The converted application matches the functionality of the original exactly, is maintainable by the existing CME developers, and dramatically improves the performance of the application it replaced.

Source System:

CA GEN z/OS CICS/DB2

Target System:

Linux platform with Oracle Database, applications migrated to Java, J2EE/JSF WebLogic

United States Department of Agriculture (USDA)

The USDA was leveraging a CA GEN application to run key financial data and processes. This application was the last in the USDA portfolio running on a legacy technology and skilled development and maintenance resources were difficult to secure.

Unisys Federal Systems chose Modern Systems as the best and least risky option for an upgrade. The converted application matches the functionality of the original exactly, is maintainable by the existing USDA developers. Along with functional parity, the USDA has dramatically improved the performance of the application it replaced.

Source System:

CA GEN z/OS, CICS/DB2

Target System:

Linux platform with Tomcat, DB2, applications migrated to Java, J2EE/JSF

> **“Modern Systems' proven technology and methodology significantly reduced risk, cost and timeline for the USDA. Modern Systems' automated process eliminated manual conversion errors and allowed us to quickly provide our customer with an open scalable web services based application.”**
>
> **Mike Boyd**
> **Capability Lead, Legacy Modernization**
> **Unisys Federal Systems**

Conclusion

Companies of all sizes are grappling with aging, complex systems that are costly to maintain and too inflexible to support new business initiatives.

Most vendors are more concerned with selling new systems than helping to retire old ones. It can take a lot of effort to move data off the old system, archive the application's data and decommission the supporting infrastructure. It can be a challenging project because of lack of documentation or data — or both — but the more legacy applications that are retired, the greater the cost savings in ongoing maintenance. This helps companies meaningfully shift their IT spending to more business-focused initiatives. Further, companies can manage operations more effectively with common technologies and processes.

Modern Systems' automated conversion technology is the safest, lowest-risk and most cost-effective solution for modernizing CA GEN systems.

About Us

Modern Systems is the leading provider of legacy language and database conversion. The Modern Systems portfolio includes a comprehensive suite of tools and services for database and application modernization.

Leveraging over 30 years of best-practice domain expertise, Modern Systems works closely with its customers to minimize risk and provide a clear path from legacy platforms like COBOL, Natural/Adabas and others to modern solutions like SQL, DB2, Java and more.

Modern Systems customers come from diverse industries and vertical markets such as automotive, banking and financial services, insurance, manufacturing, and retail.

Modern Systems has offices in the USA, UK, Italy, Romania, and Israel.