A LegacyJ Whitepaper

Future of COBOL

There is Life in Cobol

I see a long fruitful life ahead for COBOL. COBOL remains the best implementation language for business applications and will retain that distinction for the foreseeable future.

COBOL is the dominant language for business applications. Over 80 percent of the world's business runs on COBOL (Gartner Group, '97). Almost every major industry from financial applications to manufacturing real-time system relies on COBOL.

The investment in COBOL technologies, staff and hardware, is estimated to be greater than 5 trillion dollars. Over 180 billion lines of COBOL code are in use today, with an estimated 5 billion new lines added per year. COBOL applications include:

- *INVENTORY MANAGEMENT*
- *ORDER PROCESSING*
- *ACCOUNTS RECEIVABLE AND PAYABLE*
- *RETAIL*
- *SHIPPING*
- *MANUFACTURING*
- *FINANCIAL*
- *AND THE LIST CONTINUES...*

CONTENTS

Language Choices

There is no single language that meets all programming requirements on all platforms. The question comes down to *"function and fit"*. A good solution must provide great performance and fit well with hosting infrastructures. COBOL, CICS and DB2 are good examples. Some solutions while not as good a fit will have a following because of functionality. For example, Visual Basic from Microsoft provides an extensive toolset. The fit, however, is limited to select platforms.

The solution selection process boils down to "risk" then "functionality." A review of most businesses depict a complex intermingling of applications and implementation vehicles with varying mixes of assembler, COBOL, C, C++, PL/1, Visual Basic and Java, among others.

In the 80s the SHARE Language Futures Task Force recommended to IBM that they provide more support for multi-language programming. The requirement was based on the typically extensive mainframe inventory of languages. The recommendation addressed issues of interoperability - applications written in different languages should work together seamlessly. The requirement that programming languages work together, and do so efficiently, is truer today than ever. A major difference today is that the host has changed from a single platform to a multi-tier environment.

310 billion lines of software are in use today – Gartner Group

200 billion lines of COBOL are in use today (65% of the total software). This represents a $2 trillion dollar total investment. – Gartner Group

5 billion lines of new COBOL are developed every year. – Gartner Group

COBOL applications process over 80% of all daily business transactions and mainframe platforms store 70% of the all the data

Pervasive COBOL Connectivity

To succeed in a rapidly changing and pervasively connected world, the business software applications must meet these requirements: flexibility, integration, reliability and rapid return on investment. Many traditional COBOL applications fall short in meeting these requirements.

Requirement	Challenge	COBOL
Flexibility	Rapid adaptability to unpredictable marketing, financial and technological developments without restriction to a proprietary platform.	Rigid, monolithic codes developed for closed platforms.
Integration	All software applications must work together seamlessly both within the enterprise with those of customers, partners and vendors.	Designed to work and to remain in the back office, isolated from other applications.
Reliability	Stability, robustness, business continuity, security and dependability are minimum requirements of software applications.	Developed by disparate groups of programmers over an extended period of time. These individuals are long gone. Documentation rarely exists. Constant repairs and patches have made the codes fragile and unstable. Mergers and acquisitions compound this problem even more by poor integration of dissimilar COBOL systems,
ROI	Since cost is a key consideration in all business expenditures, software applications must effectively demonstrate a significant return on investment.	COBOL has traditionally been available from a few hardware vendors (Burroughs, Sperry-Rand, Hewlett-Packard, Wang, Data General, IBM and others). COBOL from a single vendor may not provide options and can restrict portability and price. In addition, independent compiler vendors have historically charged prices based on old execution paradigms that penalize client/server or web-enabled COBOL applications. Therefore a per-user charge may not make sense in the new execution model.

Technology

The IT community wants answers to, *"how do I maximize what I have?"* and *"how do I get from here to there"?* They are, in many respects, different sides of the same coin.

Once the millennium problem is solved the next major issue will be the Internet. The driving force is profit. Business seeks to decrease costs while increasing performance. Unfortunately, hidden costs particularly software maintenance costs can be life changing (and not in a good way).

Business will approach the Internet in a variety of ways. Some companies will choose to embrace seemingly better functionality over fit by taking the "baby/bathwater" approach, attempting to rewrite legacy code in more "Web-friendly" languages such as C++. Others will prepare by moving to Web-enabled turnkey solutions, at great cost and greater risk. Still others will get from "here" to "there" by leveraging existing code bases and programmer skills.

Changing Environments

COBOL continues to change to meet user-demands supporting an ever more diverse application base. Traditional "Pseudo-Conversational" CICS applications continue to exist. Today, they are being complimented with client/server and network applications. COBOL has always played a major role in transactional subsystems and will continue to do so in the future.

COBOL investments are substantial and the decision about what should be done boils down to these options:

1. Throw the investment away in a staged or "cold turkey" approach.
2. Add functionality to the investment using a GUI builder.
3. Provide a bridge to increased Web-functionality and fit by adding Java capabilities.

Maintaining Investment

Business software has evolved over decades and is seldom considered "throw away". Software applications are more and more being regarded as corporate assets, critical to business success. Software tools must fit within the software development infrastructure and allow for evolutionary rather than revolutionary growth in development and maintenance. "Technology for technology's sake" has been, and continues to be, IT's Achilles' heel.

The Gartner Group estimates that there are over 3 million COBOL programmers, far exceeding any other programming language. It represents billions of dollars in code, training, and education. The business knowledge of an experienced programming staff represents an even greater asset. Business can access new platforms and leverage existing COBOL skills bringing fit and functionality goals within reach with minimal disruption to existing programming infrastructures.

COBOL

Tried and true applications may be line-mode, text oriented report-driven and work well for producing the current desired results. The application may have been in place for decades, it still works and it isn't broken! A need for visual environments and data requests from non-traditional users is placing new demands on existing systems. Business needs to maximize information gathering, delivery, and deployment by getting data to the people who need it, when they need it; in the form they need it.

Taking information to new heights means adding functionality to COBOL. Alternatives for adding functionality to existing systems can be separated into three categories: front-, middle- and back-end improvements.

Client Options

Build a new front-end – Front-end platform independent graphical user interfaces can provide access to legacy applications and reduce maintenance and support costs associated with porting legacy applications to new platforms.

1. Create a Windows-only solution with a tool designed to take advantage of functions and features available in that environment.
2. Create a hybrid solution with COBOL and HTML/CGI
3. Create a Java based front-end that relays messages to the terminal application.
4. Use PERCobol and write once and deploy anywhere.

Server Options

Create server applications that are capable of supporting multiple diverse clients. This middleware ideally should enable the deployment of applications on any platform.

1. Write Java applications that can be interfaced with legacy programs through special constructs. Existing legacy applications continue to reside on the mainframe and/or mid-range servers throughout the enterprise. Access to new platforms is achieved with new applications written in Java. Communication with existing server-based applications is done through special APIs.
2. Write COBOL applications using one of the multi-platform COBOL compilers. These COBOL applications can then be linked to existing COBOL programs. An API or callable interface between programs allows communications between older applications residing on the server and the new applications.

PERCobol

PERCobol™ enables an evolutionary extension of COBOL applications. PERCobol compiles COBOL applications for execution on the Java Execution Environment and permits the generation of J2EE compliant Enterprise JavaBeans directly from COBOL business logic.

Evolving COBOL applications through PERCobol allows access to new platforms easily and cost effectively, up and down the computing hierarchy. PERCobol can be used to further enhance applications through the use of additional functions or features that may not be available with the current COBOL compiler in use or permitted with existing computing environments.

Single platform applications can readily be moved to take advantage of storage, computing power, graphical user interfaces or Application Servers. LegacyJ PERCobol is a valuable tool to help reduce application complexity, boost productivity and deliver platform independent application deployment.

PERCobol extends the value of legacy applications. COBOL extensions in PERCobol exploit the features of both programming environments and provide opportunities for continued growth. Increasing the life of applications by adding Java technology creates a bridge for business to connect the best of both programming worlds.

Author *Chuck Townsend*

Charles F. Townsend (Chuck Townsend) is President of LegacyJ Corporation. LegacyJ is a software development company engaged in the creation of enterprise business software.

Prior to joining LegacyJ. Townsend spent almost 30 years with IBM. While at IBM he was the Debug Manager responsible for the COBOL, FORTRAN, PL/I and C debuggers, and was instrumental in the creation of IBM's multi-platform debug strategy and product offerings.