The Future of COBOL

Mike Murach

The usual title for an article like this is, "Is COBOL dead?" But that was a silly question when I first heard it 20 years ago. And today, with tens of billions of lines of COBOL code still in use, that question is even sillier. As this report shows, COBOL will continue to play a dominant role in application development for many years to come, although the nature of that role may be changing.

The current status of COBOL on IBM mainframes

Today, the typical enterprise application in a large organization consists of dozens of COBOL programs running on an IBM mainframe computer. To drive the workstations or PCs that use those applications, the programs run under a world-class transaction processor known as CICS (Customer Information Control System). These COBOL/CICS applications process billions of online transactions every day in businesses like banks, airlines, insurance companies, and hospitals, and they provide the business logic and database processing for most large e-business sites.

In fact, one estimate says that COBOL/CICS applications account for 60% of all the applications that are currently in operation, and another estimate says that these applications process 85% of all the transactions that are processed. So when you withdraw money from an ATM, place an airline reservation, or order a product over the Internet, chances are that a COBOL/CICS application has been used to process your transaction.

In the late 90s, the large companies and organizations that run these COBOL/CICS applications spent many billions of dollars fixing their Y2K problems. As they fixed these problems, they also updated the code so these programs could continue to run indefinitely. As we see it, that spoke volumes. In short, these programs are not going to be replaced by other types of programs or systems any time soon because there isn't enough money, time, or programmers to do that. Besides that, these programs do what they're supposed to do.

What this means is that these programs will have to be maintained and enhanced indefinitely. That in turn means that there will be a continuing demand for COBOL programmers with perhaps 90% of the work as maintenance and enhancement of old programs and 10% new program development. This provides a continuing job market for COBOL programmers as older programmers retire and new programmers are hired to replace them.

Today, one common type of enhancement is to convert existing COBOL/CICS applications to web applications by converting the CICS user interface to a browser interface. For instance, the department of motor vehicles in one state converted many...
of its applications to web applications so the car owners and drivers could do much of their own processing online. This type of program enhancement is relatively easy because CICS provides tools that convert a CICS user interface to a browser interface. In this case, the COBOL and CICS for business, file, and database processing remain the same so the need for program maintenance and enhancement continues.

The current status of COBOL on smaller systems

In the 1980s, many COBOL applications were developed for small businesses and operational departments within large companies as minicomputers from companies like Digital Equipment Corporation became prominent. Unix also became popular during this period, and some COBOL applications were developed for that operating system. Eventually, though, most of these COBOL applications were replaced by applications written in other languages. So this type of application represents just a small portion of the total amount of COBOL code in operation today.

In the 1980s, there was also a push to migrate some mainframe COBOL/CICS applications to midrange systems and PC networks by converting the COBOL/CICS code to standard COBOL that would work with compilers from companies like AccuCOBOL and Micro Focus. Although this worked for some small mainframe applications, the conversion process was never easy, and this was never practical for large applications. Today, some of these migrated applications are still running, but most have been replaced by applications written in other languages.

So at least for the time being, when you think of COBOL, you should think of structured COBOL/CICS programs on an IBM mainframe because that's where 90% or more of all of the COBOL code is running. And that's where the jobs for COBOL programmers are.

Current efforts to migrate COBOL/CICS programs from mainframes to smaller systems

As I write this, there's another push to migrate COBOL/CICS applications from mainframes to Unix, Linux, or Windows networks. Three of the primary reasons for doing this are (1) to reduce hardware costs, (2) to reduce software costs (or to improve programmer productivity), and (3) to make it easier to enhance existing systems so they can provide competitive business advantages. These are legitimate reasons for considering migrations because mainframe hardware and software are indeed expensive, and COBOL/CICS applications have always been relatively hard to maintain and enhance.

To make migrations like this possible, companies like Micro Focus and Fujitsu provide COBOL compilers, conversion tools, and CICS transaction servers. To keep the migration risk as low as possible, the general approach is to make as few changes as possible to the existing COBOL/CICS code. When you're done, the CICS user interface has been converted to either a browser interface that can be enhanced with Java or ASP.NET or a Windows interface that can be enhanced with a .NET language like C# or Visual Basic. Please note, however, that the business logic, file processing, and database processing are still done by COBOL and CICS. This, of course, means that the need for COBOL maintenance and enhancement continues.

Because midrange systems are more powerful and reliable than ever before and because the reasons for migrating can be compelling, some companies have already made this type of migration. In general, though, this type of migration is only practical
for relatively small mainframe applications that have limited performance requirements. For large applications that process millions of transactions each day, migration just isn't an option. Nevertheless, this is an attractive target market for companies like Micro Focus and Fujitsu who have a lot to gain by selling their products and services to companies who are interested in getting their applications off the mainframe.

Recent mainframe developments

Meanwhile, IBM is doing its best to keep COBOL/CICS applications on mainframes and to make it easier for customers to migrate applications from other platforms to mainframes. To keep applications on mainframes, IBM provides a couple of software products that make it relatively easy to web-enable COBOL/CICS programs without even changing the COBOL or CICS code. Beyond that, IBM now provides CICS mainframe support for both Java and Linux, as well as other software that lets you use Java, COBOL, and CICS in the same application (IBM calls this "Java interoperability").

For instance, the latest mainframe COBOL compiler (called "Enterprise COBOL for z/OS") provides several features for Java interoperability. In particular, it provides object-oriented COBOL syntax that lets you (1) create object instances of Java classes, (2) invoke methods on Java objects, (3) define Java classes with methods implemented in COBOL, and (4) access Enterprise Java Beans that run on a J2EE-compliant EJB server like Websphere. This compiler also provides XML capabilities, which will make it easier to work with data across platforms.

The upshot of these IBM developments is that there are now several different ways to enhance old applications and develop new ones on a mainframe. Although it's too early to tell which of these approaches will become commonly used, it's obvious that the mainframe is no longer the static development environment that it once was. So to prepare for the future, COBOL/CICS programmers should at least add object-oriented programming with Java to their skill sets.

Should COBOL still be taught in the college curriculum?

Because COBOL and CICS are so dominant in the real world, it seems fair to say that COBOL should still have a place in the modern computing curriculum in colleges. In short, if you want to prepare your students for working with enterprise applications in large companies, you should offer at least one course that shows how to develop COBOL programs for IBM mainframes and that course should probably include an introduction to CICS. This is especially useful for Information Science, MIS, or CIS majors, and it may even help some students get entry-level jobs that they wouldn't get otherwise.

Beyond that, there's a purely academic reason for offering COBOL courses. That is, that COBOL provides a striking contrast to Java, C#, and Visual Basic. Structured programming vs. object-oriented programming. Self-documenting code vs. cryptic code. One Read statement vs. no easy way to read a record in a file. A few dozen COBOL statements vs. thousands of classes, methods, and properties. By experiencing these differences, the students develop a better perspective on business applications, development methods, and programming languages.

Please keep in mind, though, that there is general agreement among trainers that the programmer with the best preparation for the real world is the one who understands all three of the major approaches to application development: (1) Java, (2) .NET with C# or Visual Basic, and (3) COBOL/CICS on a mainframe. So we're not suggesting that COBOL take priority over Java or .NET. On the contrary, we think that all three approaches should be taught in the modern computing curriculum. That's why we
offer books on all three that are uniquely designed to teach the skills that are needed on the job. To find out more, please visit us at www.murach.com

About the author

Mike Murach is the president of Mike Murach & Associates, Inc., a leading publisher of COBOL, mainframe, Java, and .NET books. He wrote this article in response to questions from college chairpersons and instructors who have to decide where COBOL fits in the modern computing curriculum. Mike can be reached at mike@murach.com, and the company web site address is www.murach.com

###

---

**Micro Focus COBOL**
Reduce costs, increase agility and minimize risk for business success
www.microfocus.com

**Cobol**
Post Your Resume Now on Dice.com Career Hub for Tech Professionals
www.Dice.com

**Cobol Documentation**
Technical & business documentation Cobol systems - mainframe & more
www.relativity.com

**Cobol Jobs**
Interested In Working From Home? Top 25 Work At Home Jobs Of 2007
Real-Home-Opportunities.com/

**Mainframe Training**
ProTech Training offers a wide variety of mainframe courses
protechtraining.com

---

First Place Learning

---