

COBOL TRENDS IN 2020



The COBOL language is one of the oldest programming languages in use today. While some engineers predicted that this classic computer language would quickly disappear or become obsolete within a few decades, many older systems still rely heavily on COBOL.

According to [Reuters](#), you can find 220 billion lines of code still in production. From many federal government agencies to your local bank, COBOL is still in use. An estimated 43% of banking systems and 95% of ATM swipes utilize COBOL code. Let's take a look at the state of COBOL in 2020.

What is COBOL?

Developed in the late 1950s, COBOL is a programming language used primarily in major corporations and government industries. It was created by the Committee on Data Systems Languages, also known as CODASYL, a group of computer scientists and mainframe manufacturers. The Department of Defense had tasked CODASYL with designing a common business language for programming.

This "Common Business Oriented Language" came to be referred to as COBOL. The new language was inspired by computer engineer [Grace Hopper's FLOW-MATIC](#), [Univac's AIMACO](#) and [IBM's COMTRAN](#).



COBOL Today



220 billion lines of code still in production

Mainframe COBOL:

\$10k salary increase

43% of banking systems use COBOL

95% of ATMs rely on COBOL code

Based on industry averages from 2019-2020

Industries using COBOL

COBOL is most often found in companies such as:

- Banks and other financial institutions
- Insurance companies
- Transportation systems
- Government agencies

The Department of Veterans Affairs, Department of Justice, Treasury Department, Department of Homeland Security, Social Security Administration all have systems that rely in some part on COBOL. The code is used in mainframe operations and for processes such as inventory, payroll, and accounts receivable.

2020 COBOL Trends

Today's computer engineers and scientists are unsure whether COBOL will remain integrated into established systems. Some businesses have already replaced the old code with newer, easier to navigate languages. Other businesses have tasked programmers with modernizing and adapting the COBOL that's already in use.

Meanwhile, certain government agencies are struggling with the limitations of these systems and the current overload on COBOL-reliant mainframes that are now experiencing technical difficulties. Simply finding the right programmers to handle these problems can be difficult.

Replacing old systems

Some companies are making efforts to replace COBOL code that is already in use. This is no simple migration though. It can take years to replace a program that relies on COBOL. Integrating a new language into one of these older systems also requires maintenance work.

It is a time-consuming and costly task that demands expert programmers with the ability to navigate

older mainframes and slowly update commands. A [2017 Reuters article](#) reports that “When Commonwealth Bank of Australia replaced its core COBOL platform in 2012, it took five years—and cost \$749.9 million.”

Many banks and government agencies are starting to use mobile platforms that require the use of newer languages. It will take time, but for certain companies the necessity to update older systems with languages such as [Java](#), C, and Python is undeniable.

Modernizing COBOL code

Rather than eliminating the COBOL code in use, some companies are working within the system to modernize COBOL programming. Programmers are working hard to make hybrid programs that use COBOL and Java or other popular languages.

“According to a survey by [Micro Focus](#), which follows data gathered in previous 2017 survey, 70 percent of enterprises favor modernization as an approach for implementing strategic change.”

Some programmers favor modernization because they see COBOL as a flexible and resilient computer language that can be adapted to meet today's needs. These programmers see COBOL as a language that will remain relevant due to the system's inherent [readability, portability and adaptability](#). Critics of modernization cite the time investment to update older code as well as risks in developing new code. Creating hybrid code that may have problems that put sensitive consumer data at risk of being accessed by the wrong parties or completely lost.

Current challenges with COBOL

While larger questions about the longevity of COBOL use exist, programmers are currently facing challenges with today's systems that depend on COBOL. Maintaining the billions of lines of code in use is problematic.

There are limited resources for programmers wanting to learn COBOL as well as a limited number of experts available to maintain systems using COBOL. Also, finding the talent to work on older programs can be difficult. Finally, the recent global health crisis has resulted in an overload on systems using COBOL, which further iterates the need to find programmers with the ability and availability to make updates.

Learning COBOL

While COBOL may not have the same appeal as some newer, popular languages, there are still resources available to learn it—just not as many as there used to be. As of 2017, there were reportedly [75 U.S. schools still teaching COBOL](#). This can, in part, be attributed to efforts of IBM. IBM even offers enhancements which make it possible to integrate COBOL and JAVA on the same mainframe.

Leon Kappelman, a professor of information systems at the University of North Texas, believes that [students immediately benefit from studying COBOL](#). Kappelman says, “Undergrads who take the school's two classes in mainframe COBOL 'tend to earn about \$10,000 per year more starting out than those that don't.”

However, despite the possible earning potential, many students and established programmers still see the language as ancient and outdated.

Finding developers

Even though schools offer COBOL training, the hiring pool for finding these developers is small. Many seasoned mainframe and COBOL experts are nearing retirement age. At the same time, the younger generation of COBOL programmers have an opportunity to work at thriving tech companies. Unfortunately, working in the financial sector or for government agencies may not have the same appeal as Facebook and Google.

Fixing overloaded systems

The recent surge in unemployment claims has put enormous demands on older systems that rely on COBOL. "The states of Kansas, New Jersey, and Connecticut all experienced technical meltdowns after a stunning 6.6 million Americans filed for unemployment benefits last week," according to [FastCompany.com](https://www.fastcompany.com). With situations like this, the demand for state government systems to upgrade their code increases.

Certain major corporations and businesses in the financial sector may continue to find that COBOL is a stable language for their systems to rely on. But, for now, the spotlight is on government agencies to quickly maintain, update or replace COBOL systems that are experiencing technical issues.

Additional resources

BMC Blogs offers a variety of roundups on trends and statistics on IT and technology:

- [The State of Java in the Enterprise](#)
- [ITSM Trends](#)
- [The State of DevOps Report Roundup](#)
- [SaaS Growth Trends](#)
- [Cloud Revenue & Market Share Trends](#)
- [IT Spending Trends](#)