EVOLVING SOLUTIONS

KEEP YOUR COBOL UP TO DATE:

Modernizing to Enterprise COBOL V6

Tips on upgrading from Enterprise COBOL V4 to V6, and the benefits of migrating

By Jim Fyffe

Mainframe clients continue to move forward with modernization initiatives, recognizing that their platform is well-equipped to power today's cloud-connected business world. However, a critical aspect of modernization that sometimes gets overlooked is the need to upgrade COBOL code.

While Enterprise COBOL is the programming language that runs the world economy, older versions of COBOL remain in use in many mainframe environments. With IBM ending support for the popular Enterprise COBOL V4.2, a version that was announced in 2009, it is critical that enterprise clients move to V6, and then prepare themselves so they can fully take advantage of its capabilities.

And the new capabilities are quite impressive. IBM developers did a tremendous job creating and designing Enterprise COBOL Version 6. This is completely new technology. There is new firmware, a new code generator, and a significantly re-architected compiler. V6 is built for speed and precision—including the newest version, Enterprise COBOL V6.4, announced in April. It provides greater interoperability with more future-ready technologies. It is also designed to reduce system overhead, so you should see performance benefits and associated cost-savings.

The COBOL Migration Process

Of course, a COBOL migration is more involved than simply downloading the latest version. However, with preparation and planning, upgrading from V4 to V6 can certainly be managed. IBM's documentation (the migration guide), in particular, is an absolutely critical component of the new compiler technology.

The primary challenge with COBOL migration comes with the decisions that must be made. Which programs should you upgrade? How do you approach testing? The challenge here is not necessarily technical; it is one of project management. A COBOL migration project comes down to planning the migration by assessing the code inventory, and then testing the results. Let us take a closer look:

1. PLANNING

Even a small mainframe organization could easily have hundreds to thousands of COBOL programs supporting V4.2 and/or even earlier compiled releases, so pinpointing the code you want and need to convert is extremely important. In general, converting 100% of your code inventory is impractical, if not impossible. So, to help make the necessary choices, most organizations should at least explore the possibility of undergoing an assessment. Choose a provider with experience and expertise in delivering modernization services. But for those select few organizations that already thoroughly understand the contents of their mainframe COBOL portfolio, a formal assessment may be unnecessary.

2. TESTING

Wherever possible, the testing process should be automated. Automation allows you to manage your large volume of programs and is essential in making the largely repeatable syntactical code changes that will need to be implemented.

Generally speaking, there is not a blanket approach to testing. This part of the process will be determined by the particulars of your organization: the size of the IT staff, the constraints posed by your hardware, and software. When it comes to executing your migration plan, flexibility is essential. It may be necessary to deviate from your organization's standard internal processes.

During the testing process, you will likely have to adjust some of your code to comply with V6 COBOL. Certain programs—primarily those with numeric data elements—will run differently on V6. In delivering the performance improvements and other enhancements in the latest release, IBM made assumptions about organizations' individualized use of COBOL code. In some cases, developers were unaware of the volume and variety of programming loopholes that, over time, have become commonplace across client mainframe installations. Uncovering these errors can only be done by running real data through the programs, particularly unformatted data like web input or data feeds that may not be as tightly edited as your organization requires them to be.

Once testing is complete, a small pilot project should be conducted. This step is standard for modernization initiatives of all sizes and types. Simply run a select few programs on a test system and put them through the full upgrade process. The results should tell you what you need to know.

A New, Modern COBOL

Running unsupported, decades-old programs and compilers poses significant security risks. From a risk-avoidance perspective, a persuasive case can be made for migrating from V4 COBOL code. That said, an even stronger case can be made by focusing on the positives surrounding COBOL V6 in terms of providing capability and performance. You will be happy you made the effort to put it in place.

Evolving Solutions is poised to help with the COBOL migration process. Contact us to get started.

Let's get to work.

EVOLVING SOLUTIONS

 ESI-051722-x110