Marble
Enterprise Software

June 29, 2018
Excessive Time in COBOL compiles

Excessive Compile Time

Compile Time is longer when using COBOL 5 or 6. Not just longer but way longer! The storage used during compile time is also significantly greater. The benefit of COBOL 5 or 6 is significantly faster execution time of the program in production! This is good!

1 Some improvement in compile time may be had:

The following has some clear thoughts on what may be in a user's overall interest for maintaining COBOL programs and shortening Compile Time as well. The bottom line presented here is that how far a user goes in better handling excessive compile time, can be a very distinct bonus in long term maintenance of the COBOL programs affected.

2 Why improving compile time is important:

Minimum time to compile a program takes 5 to 10 times longer using COBOL 5 or 6, versus the older COBOL compilers before them.

A bad program (defined as a program that takes more time with COBOL 5 or 6) exceeds the estimate of 5 to 10 times and can go to 15 times (or greater). The last part of this statement comes from a conversation with the IBM compiler team in Toronto, Canada.

When a program takes more time to compile, this should be a red flag to the user that something needs to be done to clean up the program for the following reason:

- Almost always, when compile time takes significantly longer, the PERFORM structure of the program is convoluted in a way that makes maintenance harder on this program.
- Also, there may be hidden PERFORM Errors in the program that are causing much longer compile time. (The danger in PERFORM Errors is simply that your logic is probably incorrect and may be creating hidden errors in the output and may be costing your company extra dollars depending on the error.) To not pay attention to Perform Errors is like running empty on a car's gas tank and not paying attention to that.

As just stated, cleaning up extra compile time, also saves extra money by making the maintenance of the COBOL program clearer and easier to follow. Or in reverse, ignoring the red flag, leads to status quo and continual longer-term maintenance and extra costs beyond longer compile time.

Cleaning up the COBOL program should ideally be done by a technician with good, or better than average experience with COBOL programs.

3 Why COMPILE Time may be significantly longer for some programs

This information, again, comes from a conversation with the Toronto IBM people who are overall responsible for the newer design and delivery of COBOL 5 and 6. The structure of how PERFORMs are used in a COBOL program affects compile time!

While the entire algorithm IBM uses to create faster execution time is proprietary to IBM and not fully discussed, it was made clear that PERFORM structure and multiple PERFORMs to the same routine is the start of longer compile times. If there is more than one exit to a PERFORM or PERFORMS with overlapping ranges, these should be addressed. If there are PERFORM Errors in the program, these will create major havoc. See **PERFORM Errors** below.

It is assumed that the COBOL program is already structured. There are very few programs today that are not. Multiple PERFORMs to the same routine is common and should not be overly avoided yet can be addressed in some cases for better maintenance by breaking a large program into smaller ones.

4 Basic Steps Involved

Breaking large programs into smaller ones, is great for long term maintenance. It takes time and the time used, eliminates much long-term maintenance by making following program logic much easier.

The next step is to look for overlapping PERFORMs or PERFORMs with more than one exit. Removing these will do wonders both for maintenance and for Compile Time.

Any existing PERFORM Errors should be looked for and if found, eliminate them. To find any PERFORM Errors, see paper on 'Finding COBOL Perform Errors' from Marble Computer for quickly identifying these.

5 Tools for cleaning up a program

Using available tools if they are efficient and easy to use, help the overall process.

If the COBOL program has unused code in it (Dead Code), eliminating this first-hand and creating a smaller program without this unused code should be a first step. Control/DCD has a new Digital Documentation Manual in PDF format for any reader that can show both DATA DIVISION and PROCEDURE DIVISION dead or unused code. See Control/DCD – Quick Start Guide

Understanding PERFORM structure helps immensely in looking making changes. See Forward Tracing
The next feature is finding not always obvious PERFORM Errors. Again, the Marble Computer 'Digital Documentation Manual' in PDF format shows all PERFORM errors in a COBOL program. No other software currently on the market is able to show PERFORM Errors.

6 Another Time Saving Measure

One simple easy to use step is save time is to make less compiles and only do as needed. Using Marble Computer's Control/DCD software, you can do a pre-compile to see most of the errors a compiler would find and also get out meaningful analysis and documentation at the same time. See Less compile time by using newer Enterprise COBOL

Want to know more? Contact us today. Call 1-800-252-1400 ext. #4 or send an email to sales@marblecomputer.com