Enterprise COBOL Version 5 User Experience

Brian Peterson

Optum Technology, UnitedHealth Group

brian_d_peterson@optum.com

Session 16710

Insert Custom Session QR if Desired.

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides **education**, professional **networking** and industry **influence**.

Trademarks

Agenda

- Goals of initiative
- A Brief History of COBOL
- COBOL Version 5 – What Changed?
- Optimization value proposition
- Optimization and debugging
- Our project
- Our timeline
- Recommendations for success
- Managing the COBOL Version 5 product
- Where are we today
- Other sessions this week on COBOL Version 5

Goals of initiative

- IBM Enterprise COBOL Version 5 is the first COBOL compiler to exploit zArchitecture enhancements

- Lots of new features, but the most compelling to us:

  In exchange for a more CPU and memory-intensive compile process, there is an opportunity to save CPU cycles at runtime

A Brief History of COBOL

- Reference:

`http://ibm.com/systems/z/os/zos/features/lang_environment/history/cobmvs.html`

- IBM has offered many COBOL products over the past 40 years, starting with …

- OS/VS COBOL

A Brief History of COBOL conversions

- OS/VS COBOL to VS COBOL II
  - Mid 1980s - most programs required source code changes

- Large conversion effort (call it **100** on a “1-100 point” scale)

A Brief History of COBOL conversions

- VS COBOL II to successor products
  - COBOL/370
  - COBOL for MVS & VM
  - COBOL for OS/390 & VM

- Incremental changes, small conversion effort
  - Between **2** and **5** on a “1-100 point” scale

A Brief History of COBOL conversions

- Enterprise COBOL Version 3

- Medium conversion effort
  - Call this one a **10** on a “1-100 point” scale

A Brief History of COBOL conversions

- Enterprise COBOL Version 4

- Small conversion effort
  - Call this one a **2** on a “1-100 point” scale

A Brief History of COBOL conversions

- Enterprise COBOL Version 5

- Significant conversion effort
  - Few COBOL source code changes required, ***but***
  - Significant attention and support from our “core team”
  - Call this one a **20** on a “1-100 point” scale
  - The biggest COBOL conversion since Enterprise COBOL Version 3
    - *In other words, this is the biggest COBOL conversion in more than a decade*

COBOL Version 5 – What Changed?

- Two parts to the COBOL compiler
- “Front End”
  - New syntax supported, tightened language rules, COBOL standards changes, enhancements
  - “Business as usual” for a new compiler release
- “Back End”
  - Completely new technology generates and optimizes machine code created by the compiler
  - First COBOL compiler to take advantage of more than two decades of IBM zArchitecture machine enhancements (ARCH compiler option)

COBOL Version 5 – What Changed?

- Compiler generates Program Objects, requires PDSE
- New runtime shipped with z/OS Language Environment
  - IGZXnnnn modules in CEE.SCEERUN
  - Provides streamlined interaction between programs generated by this compiler and the runtime
- Backward compatibility and interoperability is provided all the way back to VS COBOL II (if compiled with RES)
- Not interoperable with OS/VS COBOL – ***it will not work***
  - Out of support since 1994

COBOL Version 5 – What Changed?

- ARCH
  - Compiler generates code targeted to a machine's capabilities
  - Newer architectures generate more efficient code
  - Programs compiled at higher ARCH level will not run on downlevel machines – ABEND

  - You must set the ARCH level to match the OLDEST machine in your environment where your program might run
    - *Don't forget to consider machines in your Disaster Recovery environment*
  - ARCH is one of only two COBOL options we prohibit programmers from overriding

Optimization value proposition

- New compiler uses significantly more CPU and virtual storage during the compile process
  BUT
- Generates programs that in many cases run more efficiently than any prior COBOL compiler

- Optimization is a tradeoff
  - More expensive compile process, in order to achieve savings at runtime

Optimization and debugging

- COBOL optimization has always caused at least some difficulty for debugging tools
- IBM and other debugging tool vendors have always recommended recompiling without optimization for the best debugging experience
- In COBOL V5 this is still true, but to a **much** more significant degree than ever before
- Debugging tooling is evolving to improve support for COBOL V5 and we've seen considerable improvements over the past year with our company's tooling product set

Our project

- "Core team" comprised of
  - Systems programmers (compiler, z/OS and developer tools)
  - Performance analysts
  - Source code lifecycle management
  - Applications representatives

- Strategy
  - We recommended a "Convert at Change" conversion strategy
  - Many applications used this strategy, while a few are taking this opportunity to recompile and retest their entire application

Our timeline

- Beta program for COBOL V5 during 2012-2013
  - Gave us a sense of what was coming
- COBOL 5.1 generally available June 2013
- Ordered COBOL 5 Developer Trial product Fall 2013 (5655-TRY) – three month free trial product
- Ordered GA COBOL 5.1 product in early 2014
- Began using COBOL 5.1 for our applications in spring 2014

“Core team”

- Weekly meetings (since April 2014)
  - Agenda includes an update on each open issue with IBM and COBOL-related product/tool vendors
  - Allows for rapid identification of cases where issues are not progressing to resolution, and facilitates issue escalation if necessary
- Email group mailbox/distribution list
  - Any application developer can ask a question and get a response from the core team
- Self help
  - Developers can “fall back” to COBOL 4.2 if needed through an option in our source code management tooling

Recommendations for success

- Create a “core team”
- Develop expertise with the new COBOL dump and storage/memory layouts
- Identify key application reps to include in the project
- Identify key tool vendors and build a relationship with their support organization
- Manage your vendors – IBM as well as third party tool vendors
  - We have had a very positive and supportive relationship with IBM as well as our major developer tool vendor Compuware

Managing the COBOL Version 5 product

- IBM maintains a web page listing fixes for COBOL V5

  `http://www.ibm.com/support/docview.wss?uid=swg27041164`

  - Or, search for “Fix List for Enterprise COBOL”

- Service for the COBOL compiler and Language Environment (LE) runtime seems to be on an “every 8 weeks” cadence

Managing the COBOL Version 5 product

- We have treated every COBOL V5 / LE PTF so far as “HIPER” and have install them ASAP
  - So far, every COBOL / LE PTF release package has fixed one or more problems we’ve actually encountered

- APAR PI18087 describes a “incorrect results at runtime” defect
- Fixed back in **May 2014**, PTF UI18382

- **Flagged as HIPER in Jan/Feb 2015**

Managing the COBOL Version 5 product

- An “RSU-only” (or worse, “HIPERs-only”) service strategy is not sufficient for this product, in our experience
  - Delaying installation of PTFs is always a tradeoff
    - Risk of installation of service that goes PE, ***versus***
    - Risk of Rediscovery, on our system, of a problem that’s already fixed
- “RSU-only” causes increased rediscovery

- In our view, every “incorrect results at runtime” APAR should automatically be HIPER
  - Meanwhile, we treat them as HIPER ourselves

Where are we today

- Several thousand programs are COBOL 5 in production
- About half of our programs compiled since April 2014 are COBOL 5 vs COBOL 4.2, and the COBOL 5 percentage is increasing

- We have opened 39 PMRs since April 2014
  - Eight currently “open” status, with four of these to be resolved by “Feb 2015 PTF” fix release
- Over twenty APARs have been created for issues we reported, while other PMRs matched issues other customers reported and were already in the process of being fixed
  - Similar statistics for our development tool vendor’s product suite

What's next

- When can we safely deinstall COBOL 4.2?
  - Not yet!
    - We hope to be able to deinstall COBOL 4.2 later in 2015

What's next

- COBOL 5.2 is now available, with new features
  - **SIZE** compiler option for configuring memory during the compile process **is removed**
    - If too small, "front end" runs out of memory, if too large, "back end" runs out of memory
    - This option is GONE in COBOL 5.2 – Hurray!

What's next

- COBOL 5.2 is now available, with new features
  - RULES for highlighting problematic syntax situations
    - RULES(NOEVENPACK) issues message for COMP-3 Packed Decimal fields defined with an even number of digits, a situation where COBOL 5 will truncate the high order digit (per the COBOL Standard) more often than prior compilers
    - This new message gives our developers an opportunity to find this subtle coding error

Questions?

Other sessions this week on COBOL Version 5

- Monday
  - 16609: COBOL V5.2 Was Announced! What's New?
- Tuesday
  - 16615: How to Take Advantage of the New COBOL V5 Compiler - Migration!
- Friday
  - 17033: COBOL V5 Migration Strategies