How to take advantage of the new Enterprise COBOL V5/V6 compilers – Migration!

Tom Ross 'Captain COBOL'

May17, 2016

#SHAREorg

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

Copyright © 2016 by SHARE Inc. Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license. http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration

SHARE
EDUCATE • NETWORK • INFLUENCE

- A little history of COBOL Migrations
- What is different about COBOL V5/V6
- What is different about migrating to COBOL V5
- Best practices recommendations

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration History

- A little history of COBOL Migrations
- First, some terminology
- What is a compiler ‘front end’ ?
  - Parser/Syntax checker
  - Builds dictionary of data items
  - Creates internal representation of COBOL statements
- What is a compiler ‘back end’ ?
  - Machine code generator
  - Register allocator/manager
  - Optimizer

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration History

Compiler	Front end	Back End
OS/VS COBOL	74 Std	1st generation
VS COBOL II	85 Std (new)	2nd generation (new)
COBOL/370	85 Std (same)	2nd generation(same)
COBOL for OS/390 V2	85 Std (same)	2nd generation(same)
COBOL for z/OS V3	85 Std (same)	2nd generation(same)
COBOL for z/OS V4	85 Std (same)	2nd generation(same)
COBOL for z/OS V5	**85 Std (same)**	**3rd generation(new!)**
COBOL for z/OS V6	**85 Std (same)**	**3rd generation(same!)**

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration History

- OS/VS COBOL to newer (any)
  - Most difficult of all
    - Source incompatibilities between 1974 COBOL Standard and 1985 COBOL Standard
    - New code generator with ‘more accurate’ numeric results

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration History

- VS COBOL II thru Enterprise COBOL V3 to COBOL/370 thru Enterprise COBOL V4
  - Super easy!
    - Source is compatible
    - Generated code is the same between versions
    - Even ‘invalid COBOL data’ behaves the same

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration History

- VS COBOL II thru Enterprise COBOL V4 to Enterprise COBOL V5/V6
  - Medium difficult
    - Source is compatible
      - A few new reserved words and some removed (rarely used) language
      - Most programs compile cleanly with COBOL V5/V6 with no source changes
    - New code generator when used with ‘invalid COBOL data’ can give different results from previous compilers (More on this later)

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5/V6 Migration

- What is different about COBOL V5/V6
  - More memory required at compile time, about 20x
  - More time is required at compile time, 5x to 15x
    - Depending on optimization level
  - More compiler work datasets (SYSUTx) required (Use new compile PROCs)
  - Executables must be in PDSE load libraries
  - Old IGZEBSTs can cause problems for VS COBOL II programs mixed with COBOL V5/V6
    - Link edit/bind time correction

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5/V6 Migration

- What's different about COBOL V5/V6 migration?
  - Differences from previous compilers can mostly only be found by testing
- Why doesn't the compiler give error diagnostics for 'invalid COBOL data'?
  - We will describe several cases, but in general it is data values at run time or inter-program dependencies, neither of which can be found by a compiler

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration

- Why didn't IBM enforce rules against invalid COBOL data for the past 30 years?
  - IBM does not test invalid data in general
    - And we had no idea there was this much 'misuse' of COBOL by customers!
  - The COBOL Standard provided solutions
    - EG: IF NUMERIC
  - IBM provided solutions
    - EG: SSRANGE compiler option

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration

- Best practices recommendations
  - But first, why we have new ones!

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

SHARE
EDUCATE • NETWORK • INFLUENCE

- Some users had problems with defects in the COBOL V5 compiler or runtime library, and because of this we recommend aggressive installation of service for COBOL V5
- Most migration problems, and the ones that cause the most customer pain, are problems with invalid COBOL data in programs
  - These are problems which are solved by the customer changing their COBOL source code or the source of their COBOL data
  - They mostly can only be found by testing, which also causes grief

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration Notes

- The main invalid COBOL issues seem to be invalid data and parameter passing, but here are the top 4:
  - Invalid data in numeric USAGE DISPLAY data items
  - Parameter/argument size mismatch
  - SSRANGE violations:
    - ❑ Modifying data outside the bounds of a table
      - ✓ Especially a table with INDEXED BY indexes
    - ❑ Using tables when the ODO object value is not in the legal range
  - Overpopulated data items, with values that have more digits than are defined in the data definitions

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration Notes

- To find these cases of invalid COBOL, IBM has recommendations for migrating to COBOL V5:
  - We recommend always using RULES(NOEVENPACK)
    - To reduce risk of overpopulated numeric items
  - We recommend always using DIAGTRUNC
    - To find any cases of ‘hidden’ loss of data when statements truncate numeric data items
  - Use **Scanning COBOL programs for compatibility** feature of RDz to check parameters
    - To find parameter mismatches

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration Notes

- To find these cases of invalid COBOL, IBM has recommendations for migrating to COBOL V5:

1) Compile with SSRANGE, ZONECHECK and OPT(0) for initial code changes and unit test
   - To find table misuse and invalid data
   - OPT(0) programs are easiest to debug
2) Recompile with NOSSRANGE, NOZONECHECK and OPT(2) for quality assurance test and production
   - NOSSRANGE and NOZONECHECK are required for good performance
   - OPT(2) is preferred for good performance in production

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

- Invalid data in numeric USAGE DISPLAY data items

```
77 VALUE0 PIC X(4) VALUE ’00 0’. *> x’F0F040F0’, third
                                 *> byte x’4’ for zone
                                 *> bits, OK in PIC X,
77 VALUE1 REDEFINES VALUE0 PIC 9(4). *> not valid in
                                 *> PIC 9 USAGE DISPLAY
PROCEDURE DIVISION.
   IF VALUE1 = ZERO
      DISPLAY ’ZERO'
   ELSE
      DISPLAY ’NOT ZERO'
   END-IF
```

Whether the program displays 'ZERO' or 'NOT ZERO' depends on the compiler options you use in COBOL v4 (and earlier) and in COBOL V5.

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

SHARE
EDUCATE • NETWORK • INFLUENCE

- VALID data in numeric USAGE DISPLAY data items
  - If you are certain that there is no invalid USAGE DISPLAY data used by your program, you can use ZONEDATA(PFD) in COBOL V5.
- INvalid data in numeric USAGE DISPLAY data items
  - Otherwise, to continue getting the same results as you did with COBOL V4 or earlier,
    - If you used NUMPROC(MIG) in COBOL V4 or earlier, use ZONEDATA(MIG) and NUMPROC(NOPFD) in COBOL V5.
    - If you used NUMPROC(NOPFD) in COBOL V4 or earlier, use ZONEDATA(NOPFD) and NUMPROC(NOPFD) in COBOL V5.
    - If you used NUMPROC(PFD) in COBOL V4 or earlier, use ZONEDATA(NOPFD) and NUMPROC(PFD) in COBOL V5.

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

SHARE
EDUCATE • NETWORK • INFLUENCE

- Invalid data in numeric USAGE DISPLAY data items
  - From the Enterprise COBOL V5.2 Programming Guide: Each digit of a valid zoned decimal number is represented by a single byte from X'F0' through X'F9'. The 4 high-order bits of each byte are zone bits, and the 4 low-order bits of each byte contain the value of the digit. The 4 high-order bits of the low-order byte for SIGN TRAILING represent the sign of the item. The sign is in the high-order byte with SIGN LEADING, or in a separate byte for SIGN IS SEPARATE.
  - COBOL V5 assumes that the zone bits are always correct. COBOL V5 generates instructions that are different from previous COBOL compilers, and could get different results in cases when the zone bits are not valid.

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

- Invalid data in numeric USAGE DISPLAY data items
  - Invalid zone bits can be detected with the ZONECHECK compiler option:
    - COBOL V6.1 in base
    - COBOL V5.2 May 2015 PTFs
    - COBOL V5.1 June 2015 PTFs and
    - COBOL V4.2 PTF UI32232 (Oct 2015).

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

- Parameter/argument size mismatch:

```
   77   GRP1 PIC X(100).
Procedure Division.
. . .
   Call ‘SUBP’ Using GRP1.

Program-Id. SUBP.
Linkage Section.
   01  GRP2 PIC X(500).
Procedure Division Using GRP2.
   MOVE ‘stuff’ To GRP2(300:20) *> Illegal!
```

Results:

–For V2, V3, V4: Illegal program did not fail

–For V5R1: File-status in CALLER changed, flow changed

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

- Parameter/argument size mismatch 2:

```
   77   GRP1 PIC X(100).
Procedure Division.
. . .
   Call ‘SUBP’ Using GRP1.

Program-Id. SUBP.
Linkage Section.
   01  GRP2 PIC X(1).
Procedure Division Using GRP2.
   MOVE ‘stuff’ To GRP2(1:100)
```

Results:

- For V2, V3, V4: Moved 100 bytes, did not follow COBOL rules
- For V5: Moved 1 byte
  - Fixed in APAR PI57812 !

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

- Parameter/argument size mismatch:
- New feature of
  IBM Rational Developer for z Systems V9.5:

  - **Scanning COBOL programs for compatibility**
    - Use the Scan for Compatibility action to scan a set of COBOL programs to determine whether the parameters passed between calling and called programs are compatible.
  - **About this task**
    - The Scan for Compatibility action can detect when a calling program is passing incompatible arguments to a called program.
      NOTE: This can only work for CALL ‘literal’ statements

Complete your session evaluations online at SHARE.org/SanAntonio-Eval
Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration Notes

- Modifying data outside the bounds of a table
  - Including data following a table with INDEXED BY indexes
- You may see different results with statements that modify data beyond the end of a table in COBOL V5 compared to previous compilers as memory is laid out differently in a few ways in COBOL V5.
  - In V5, indices are stored immediately after the table rather than being stored elsewhere in memory.
- These types of invalid programs can be detected with the SSRANGE compiler option.

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

- Using tables when ODO object value is not in legal range

```
01 OBJ PIC 9(5) BINARY.
01 MY-TABLE.
  02 T OCCURS 0 TO 1 TIMES DEPENDING ON OBJ.
    05 MY-FIELD PIC X(1).
01 OFLOW PIC X(500).

...
MOVE 300 TO OBJ.            *> Legal if table not referenced
MOVE ALL ‘M' TO MY-TABLE. *> Illegal, ODO object not in range
DISPLAY MY-TABLE
DISPLAY OFLOW
```

- Different results in different versions of COBOL
  - V2, V3, V4: Moved 300 bytes of ‘M’
  - V5: Moved 1 ‘M’ and 299 bytes of ‘other’
- You can use SSRANGE to detect this problem

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

- Overpopulated data items, with values that have more digits than are defined in the data definitions:

```
01 X PIC X(2).
01 Y REDEFINES X PIC 9(2) PACKED-DECIMAL.
01 Z REDEFINES X PIC 9(3) COMP.

MOVE x'123F' TO X   MOVE Y TO …
MOVE x'FFFF' TO X   MOVE Z TO …
```

- There's no way to detect this at compile time or by scanning source code, but problems are restricted to MOVEs between such overpopulated data items
- The RULES(NOEVENPACK) option in COBOL V5.2 helps you find cases where your programs define an even number of digits for PACKED-DECIMAL data items

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

COBOL V5 Migration: Recommendations

- Before you buy COBOL V5/V6, (or even if you have already started) we recommend the following:
  - Install the latest maintenance on LE, DB2, Binder and other products that is required for COBOL V5/V6
    - (Use the COBOL V5 FIXCAT feature as documented here: http://www-01.ibm.com/support/docview.wss?uid=swg21648871.)

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

Before you buy COBOL V5/V6

- Install latest maintenance required for COBOL V5/V6:
  - Run the SMP/E MISSINGFIX command to find required PTFs (LE,DB2,CICS,Binder, etc) for the new compilers:

```
SET BDY(GLOBAL)
REPORT MISSINGFIX ZONES(ZOS13T,ZOS13P)
FIXCAT(IBM.TargetSystem-RequiredService.Enterprise-COBOL.V5R1,
       IBM.TargetSystem-RequiredService.Enterprise-COBOL.V5R2,
       IBM.TargetSystem-RequiredService.Enterprise-COBOL.V6R1)
```

    This command will look for all PTFs needed for COBOL V5/V6

  - Install indicated PTFs on all systems before using the new compiler

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

Before you buy COBOL V5/V6

- Convert PDS COBOL load libraries to PDSE datasets
- Locate all OS/VS COBOL programs and either target them for early migration to V5/V6 or migrate to V4
  - (Get rid of the “OS/VS COBOL” problem early)
- Change build processes in the BIND/LINK step to help with the "old VS COBOL II bootstrap" problem
  - REPLACE IGZEBST, -IMMED
  - (This will not fix all, but is a no risk change to process and could have good reward)

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

Before you buy COBOL V5/V6

- Start using the new migration process with COBOL V4.2 to see if you have 'invalid COBOL data'

1) Compile with SSRANGE, ZONECHECK and NOOPT
   - Run tests with 'normal' data
   - If you get SSRANGE or ZONECHECK errors, you may have problems migrating to COBOL V5/V6
     a) Fix the programs and/or the data
     b) Plan to tolerate the data
   - If you get no SSRANGE or ZONECHECK errors, then you will have an easy migration to COBOL V5/V6!!
     - It also means you have 'clean code' that will be safe for any newer compilers!

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/

QUESTIONS?

Complete your session evaluations online at SHARE.org/SanAntonio-Eval

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license.
http://creativecommons.org/licenses/by-nc-nd/3.0/