#SHAREorg

How to take advantage of the new COBOL V5 compiler: Migration

August 14,2013

Tom Ross

SHARE in Boston

Enterprise COBOL V5.1 Migration

Standard Legal Disclaimer

Enterprise COBOL V5.1 Migration

- Agenda
  - Introduction to COBOL V5.1
  - Are you ready to migrate to COBOL V5?
  - Pre-requisite software
  - Differences in using the new compiler
  - Binding COBOL V5 programs
  - Restrictions for running COBOL V5 programs
  - Differences in running the new programs
  - How to migrate!

Intro to Enterprise COBOL for V5.1

- Enterprise COBOL for z/OS, Version 5 Release 1 (V5.1)
  - Announced April 23, GA June 21
  - New features presentation here at GSE was Monday (session S502)
- Introduces advanced optimization technology
  - Designed to optimize applications for current and future System z hardware
  - Initiate delivery of performance improvements seen in C/C++ and Java compilers on System z
- Provides compatibility with previous COBOL releases
  - No need to recompile entire applications to take advantage of new V5 features
- Support modern development tools
  - IBM & ISVs
- Continue to deliver new features
  - to simplify programming and debugging to increase productivity
  - to modernize existing business critical applications

Intro to Enterprise COBOL for V5.1

- New compiler, new Web Sites!
- COBOL for z/OS Home page:
  - http://www-03.ibm.com/software/products/us/en/entecoboforzos
- Includes links to new publications page:
  - Enterprise COBOL for z/OS library
  - http://www-01.ibm.com/support/docview.wss?uid=swg27036733
- Links to new resources (news) page
  - Enterprise COBOL for z/OS resources
  - http://www-01.ibm.com/support/docview.wss?uid=swg21634215

Intro to Enterprise COBOL for V5.1

- New compiler, new Migration Guide!
- Run-time migration information no longer included
  - The Enterprise COBOL V4.2 Compiler and Runtime Migration Guide should be used
- Enterprise COBOL V5.1 Migration Guide
  - Streamlined! Compiler migration information only
  - Specific chapters for each compiler 'starting point'
  - Easy to find the small amount of information that you need

Intro to Enterprise COBOL for V5.1

- New compiler, new LE member number!

00 Reserved
01 Language Environment (CEL)
02 Reserved
03 OS/390 C/C++, C VM/ESA
04 **Reserved =>** Enterprise COBOL V5
05 COBOL for OS/390 & VM, COBOL for MVS & VM
06 Debug Tool
07 VS FORTRAN
08 DCE
09 Available
10 PL/I for MVS & VM
11 VisualAge PL/I for OS/390
12 Berkeley Sockets
13 Available
14 Reserved
15 ASSEMBLER
16 Reserved

Ready to migrate to COBOL V5?

- Complete your migration to LE
  - The Language Environment dataset SCEERUN is installed in LNKLST or LPALST
  - There are no instances of COBLIB, VSCLLIB or COB2LIB in LNKLST or LPALST
  - There are no instances of COBLIB, VSCLLIB or COB2LIB in JCL STEPLIB or JOBLIB statements in batch jobs or in CICS startup JCL
  - All statically bound runtime library routines (for programs compiled with NORES) and bootstrap modules (for programs compiled with RES) have been REPLACEd with routines from Language Environment.

Ready to migrate to COBOL V5?

- Identify any old COBOL in load libraries
  - OS/VS COBOL
    - z/OS V2 will put warning message when OS/VS COBOL programs are run or called
  - VS COBOL II compiled with NORES
- Also identify other AMODE 24 programs
  - Assembler
  - PL/I, Fortran, vendor code?
  - Mixed with or used by COBOL code?
- Use Debug Tool LMA, Edge Portfolio Analyzer, AMBLIST, or similar

Pre-requisite software

- z/OS V1R13 or later
- CICS Transaction Server for z/OS, V3 or later
- IBM DB2 V9 or later
- IBM IMS V11 or later
- PD tools V12 or later
  - Debug Tool
  - Fault Analyzer
  - Application Performance Analyzer **(V13)**
- Rational Developer for System z V9

Pre-requisite software

- COBOL is pushing the limits!
  - Requires updates to many other products
  - As of Enterprise COBOL for z/OS V5.1, you must use SMP/E FIXCAT to identify the required PTFs on other products to work with COBOL for z/OS V5.1.
  - The required service PTFs for COBOL for z/OS V5.1 are not documented in
    COBOL for z/OS V5.1 Migration, are not included in PSP buckets, and are not included in any handouts for conferences.
  - SMP/E FIXCATs allow you to have the most up to date and correct information about Enterprise COBOL for z/OS V5.1, required service, and is the easiest way to quickly determine if you have all the necessary required service PTFs installed.

Pre-requisite software

- How to use FIXCAT
  - For Enterprise COBOL for z/OS V5.1, you should use the SMP/E V3R6 or SMP/E V3R5 product support for FIXCAT HOLDDATA to do programmatic target system PTFs verification.
  - These PTFs are identified with a FIXCAT called IBM.TargetSystem-RequiredService.Enterprise-COBOL.V5R1 in Enhanced HOLDDATA.
  - A HOLDDATA type FIXCAT (fix category) is used to associate an APAR to a particular category of fix for necessary target system PTFs.
  - To help identify those PTFs on your current system that would be needed for your upgrade to Enterprise COBOL for z/OS, V5R1 and are not yet installed, you can use the SMP/E REPORT MISSINGFIX command.
  - Here is a sample command used to run against your z/OS CSI:

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

For complete information about the REPORT MISSINGFIX command, see SMP/E Commands.

Pre-requisite software

- COBOL is pushing the limits!
  - Requires updates to many other products
- LE APARs – V1R13 (included in V2R1)
  - PM88047 - ADD SUPPORT FOR ENTERPRISE COBOL V5
    - COBOL V5 runtime compid 568819812
  - PM88048 - ENTERPRISE COBOL V4/V5 INTEROPERABILITY
    - ‘Old’ COBOL runtime compid 568819802
  - PM87183 - NEW FUNCTION SUPPORT COBOL V5
    - CEL core compid 568819801
- Binder APARs – V1R13 (included in V2R1)
  - OA41268 - for problems related to WSA size.
  - OA40593 - for problems with Debug Tool in CICS
  - OA42047 - for problems with double relocation
- DB2 APAR – V9 & V10 (included in V11)
  - PM77300 - DB2 Stored Procedures for COBOL V5

Pre-requisite software

- CDA
  - PM88903 – COBOL V5 support
- Debug Tool – V12
  - PM85967 – Add support for COBOL V5.1 programs
- CCCA
  - PM86253 – add support for changing the latest new reserved word
    - XML-INFORMATION
  - PM89219 – add support to correct programs that use COBOL V4.2 language that has been removed from COBOL

Pre-requisite software

- Enterprise COBOL V4 Migration Aids
  - PM85873 for new compiler option FLAGMIG4
    - This one will help customers identify if they have COBOL statements that are unsupported in V5
  - PM85035 for XML-INFORMATION (new function)
    - This one will help customers on V4 migrate to XMLPARSE(XMLSS) and therefore to V5 and later
  - PM87347(LE APAR) for XML-INFORMATION run-time support

Pre-requisite hardware

- Enterprise COBOL for z/OS, V5.1
  - Has minimum hardware requirement of z990/z890
  - z/990 and z/880 shipped in June 2003
  - Hopefully old enough not to affect you!
  - Older compilers would compile and programs would run almost anywhere
  - With new ARCH option we can now compile programs that will not run at all on some machines…progress? ☺

Differences in using the new compiler

- **Compiler resides in PDSE**
  - Can no longer be added to LPA via LPALST
  - Can add to LPA dynamically after system IPL
- **Compiler uses LE at compile time**
  - Must be the latest, with all V5 PTFs for installed
  - SCEERUN and SCEERUN2
- **Compiler EXITs (EXIT compiler option) changes**
  - Cannot use RTEREUS, IGZERRE or CEEPIPI
  - Can still use non-LE-conforming assembler
  - Can use LE-conforming assembler
  - Can use old or new COBOL
    - Cannot use STOP RUN

Differences in using the new compiler

- **More datasets used by COBOL V5**
  - More working datasets
    - SYSUT8-SYSUT15 are now required
    - SYSMDECK is now required
- **More memory required at compile time**
  - Recommend 200M region size
    - Old compiler could compile most programs with 10M
    - You may have to change system user id limits
- **More time required to compile**
  - 5 to 15 times as much time, depending on OPT level and source program size

Differences in using the new compiler

- **If your COBOL V5 compile has strange ABENDs:**
  - First make sure you made the changes from previous slide
    - Add region and time before you call IBM ☺
  - With less than 80M region, you get one ABEND

    ```
    IEA995I SYMPTOM DUMP OUTPUT  869
        USER COMPLETION CODE=4093 REASON CODE=0000001C
    ```
  - 80M to 120M another ABEND, at least with message

    ```
    IEW4000I FETCH FOR MODULE IGYCBE   FROM DDNAME STEPLIB  FAILED BECAUSE
    INSUFFICIENT STORAGE WAS AVAILABLE.
    ```
  - Not enough time may get yet another strange ABEND
    I could not reproduce, but it hit me during development

New compiler options for performance:

- **New ARCH compiler option requires system knowledge**
  - ARCH allows you to fully exploit your hardware
  - What is your hardware?
  - New challenge for COBOL, not new for PL/I or C
- **What hardware do you run COBOL on?**
  - What hardware is your disaster recovery system?
    - Compile for lowest level of hardware
- **A supported system with z/OS V1R13 could be on old hardware that does not support COBOL V5!**
  - Operation Exception (0C1) should trigger checking of ARCH setting and hardware level

New compiler options for performance:

- ARCH(6)
  - 2084-xxx models (z990)
  - 2086-xxx models (z890)
- ARCH(7)
  - 2094-xxx models (IBM System z9 EC)
  - 2096-xxx models (IBM System z9® BC)
- ARCH(8)
  - 2097-xxx models (IBM System z10 EC)
  - 2098-xxx models (IBM System z10 BC)
- ARCH(9)
  - 2817-xxx models (IBM zEnterprise z196 EC)
  - 2818-xxx models (IBM zEnterprise z114 BC)
- ARCH(10)
  - 2827-xxx models (IBM zEnterprise EC12)
  - 2828-xxx models (IBM zEnterprise BC12)

New compiler options for performance:

- ***OPTIMIZE(0 | 1 | 2)***
  - Levels of optimization
    - Higher levels improve run time performance
      - *But also increase compile time*
    - Highest level has slightly reduced “debuggability”
  - Applications that have user-written condition handlers are restricted, must use OPT(0) w/NOTEST
    - With TEST compiler option can use any OPT level
- ***MAXPCF(nnnnn)***
  - *Some programs can take too much time or memory to compile*
  - *Sets OPT level to 0 if program is too large or complex*
  - *Automatic control of OPT levels for large or complex programs*
  - *Default = 60,000   (MAXPCF(0) means always try to optimize)*

New compiler options for performance:

- ***HGPR (PRESERVE | NOPRESERVE)***
  - Use high word of registers (upper 32 bits of 64-bit registers)
  - Effectively adds 16 more registers to improve optimization
- ***AFP(VOLATILE | NOVOLATILE)***
  - Use full complement of floating point registers.
- ***STGOPT / NOSTGOPT***
  - Allows compiler to delete unreferenced data items
  - Analogous to FULL suboption of OPTIMIZE in V4
  - Do not use if you have ‘eye-catchers’ in WORKING-STORAGE
    - *Or use a fake reference:*
    - *IF X NOT = X THEN Display Eye-Catcher*
- **Use IBM defaults as your installation defaults for all of these**
  - Change only for special cases, individual applications

Changed compiler options:

- *NOTEST suboptions for debugging information*
    - *NOTEST(DWARF)*
        - *Full optimization plus variables in CEEDUMP*
    - *NOTEST(NODWARF)*
- *TEST suboptions*
    - *HOOK removed (always no compiled-in hooks)*
    - *SEPARATE removed (debug information always in object)*
        - *NOLOAD class segments mean no increase in LOAD size*
    - *EJPD still supported*
        - *Debug Tool now allows GOTO/JUMPTO with NOEJPD*
            - “Dirty GOTO”, much like competitors debuggers
    - *SOURCE new suboption*
        - *To control whether source information in Debug information*

Changed compiler options:

- *SIZE compiler option*
  - The SIZE option value is no longer an upper-limit for the total storage used by a COBOL compilation. Storage used by the code-generation and optimization phases of the compiler is not restricted by the SIZE option value.
  - SIZE(MAX) is no longer supported
    - *SIZE(MAX) converted into SIZE(5000K) if specified*
  - *Code generation and optimization can require a variable amount of compile-time storage, depending on the size and complexity of the COBOL source program being compiled, and is limited only by the MVS region size.*
  - *A minimum region size of 200M is recommended*

Changed compiler options:

Removed compiler options:

- *NUMPROC(MIG)*
- *NOLIB*
  - *LIB behavior always in effect*
- *DATEPROC/YEARWINDOW*
- *XMLPARSE*
  - *XMLPARSE(XMLSS) behavior always in effect*
- *SSRANGE range checks cannot be disabled*
  - *Runtime options CHECK(OFF) and NOSSRANGE have no effect*
- *NORENT + RMODE(ANY)*
  - *Now illegal: NORENT programs must have RMODE 24*

Compiler options restrictions removed:

- *MDECK does not require LIB*
- *EXIT does not conflict with DUMP*

COBOL language removed

- *Millennium Language Extensions*
- **The removed elements are:**
  - **`DATE  FORMAT` clause on data description entries**
  - **`DATEVAL` intrinsic function**
  - **`UNDATE` intrinsic function**
  - **`YEARWINDOW` intrinsic function**
  - **`DATEPROC` compiler option**
  - **`YEARWINDOW` compiler option**

COBOL language removed

- *LABEL DECLARATIVES*

  Format 2 declarative syntax:

  ```
  USE ... AFTER ... LABEL PROCEDURE ...
  ```

  and the syntax:

  ```
  GO TO MORE-LABELS
  ```

  are no longer supported.
- **Old XML PARSER**
  - **The original COBOL Version 3 Parser**
  - **Supported in V4 via XMLPARSE(COMPAT) option**

Binding COBOL V5 programs

- No more load modules
  - Compiler output is GOFF format object, input to binder
  - Binder output is Program Object
  - Program Object cannot reside in PDS, PDSE only
- NOLOAD segments
  - With TEST, debugging information is in program object
  - Always matches executable
  - Never needs to be searched for
  - Does not take memory when program is loaded!
  - The perfect solution for debug information

Restrictions for running COBOL V5 programs

- Interoperability with old COBOL programs
  - OS/VS COBOL: not supported
  - VS COBOL II NORES: not supported
  - Recompile both types with COBOL V5
- Interoperability with AMODE 24 programs
  - Dynamic CALLs between COBOL V5 and AMODE 24 programs are supported (use DATA(24) option)
  - Static CALLs are not supported
    - Convert AMODE 24 to AMODE 31 or
    - Convert to dynamic CALL with DATA(24)

Differences in running the new programs

- More memory required
- STORAGE and HEAP options have no effect because WORKING-STORAGE is not in heap for programs that
  - Are compiled w/DATA(31) and run in non-CICS
  - Programs with XML GENERATE or XML PARSE
  - Programs compiled with DLL (including programs with OO)
  - Programs linked with C/C++ or PL/I programs
- STORAGE and HEAP still affect WORKING-STORAGE for programs compiled with DATA(24) and/or run in CICS
    - WORKING-STORAGE is in HEAP
- Applications with user-written condition handlers
  - Cannot use higher OPT compiler option levels with NOTEST

Differences in running the new programs

- Not all incorrect programs are diagnosed as incorrect (by either Version 4 or 5)
  - And the results may differ between the two
- Over-populated numeric values
  - E.g., S99 PACKED DECIMAL set to value 123 (X'123C')
  - E.g., S9(6) USAGE BINARY set to +123456789
  - E.g., PACKED or ZONED with invalid sign code
- These issues are not common. We found them writing unit tests developing the new compiler
  - None in our vast test suite or in the beta program
- We recommend more vigorous testing then previously
  - Migrating from V3 to V4, almost no testing was necessary
  - If possible, try invalid data, bad data, extremes, as well as normal testing

How to migrate - Check list

- First, things to find out about your COBOL:
  - Hardware levels where COBOL applications are run (Include disaster recovery site hardware)
  - Region size available for compiles
  - Compilations that use the EXIT compiler option?
  - OS/VS COBOL or VS COBOL II NORES?
  - AMODE 24 programs statically linked?
  - User-written condition handlers?
  - IBM COBOL programs compiled with CMPR2
  - Enterprise COBOL Programs that use XML PARSE
  - Any programs that use LABEL DECLARATIVES or MLE
  - Default compiler options in current compilers

How to migrate – TO Do Check list

- **Hardware levels used by COBOL applications**
  - Choose ARCH level appropriately
- **Region size available for compiles**
  - Increase to 200M if not there already
- **Compilations that use the EXIT compiler option?**
  - Modify EXIT programs as necessary
- **OS/VS COBOL or VS COBOL II NORES?**
  - Convert to COBOL V5.1 along with other programs in the applications that interact with these or
  - Convert to Enterprise COBOL V4 in preparation for V5

How to migrate – TO Do Check list

- **AMODE 24 assembler programs?**
  - Convert to AMODE 31
  - It is not as hard as you think!
    - See conference presentation
    - http://www-01.ibm.com/support/docview.wss?uid=swg27001514
    - Do not just re-link with AMODE 31!
- **User-written condition handlers?**
  - Compile with TEST or with OPT(0)

How to migrate – To Do Check list

- **IBM COBOL programs compiled with CMPR2**
  - VS COBOL II, COBOL/370 V1.1 thru COBOL OS/390 V2.2
  - Convert source code to NOCMPR2
  - CCCA can do it automatically
  - Use FLAGMIG option and make changes by hand
  - Once converted to NOCMPR2, can compile with COBOL V5.1
- **Any programs that use LABEL DECLARATIVES or MLE (probably rare)**
  - Remove any such code

How to migrate – To Do Check list

- **Enterprise COBOL Programs that use XML PARSE**
  - If V3 or V4 w/XMLPARSE(COMPAT) must change code
    - Convert to new parser on Ent COBOL V4
  - If V4 w/XMLPARSE(XMLSS)
    - Skip this topic, code is ready for COBOL V5.1
  - Once converted to XMLPARSE(XMLSS) on COBOL V4, you can compile with COBOL V5.1 with no other changes!
  - New tool to help is available in Enterprise COBOL V4
    - XML-INFORMATION special register added by APARs (PM85035 and PM87347)

How to migrate – To Do Check list

- **Enterprise COBOL Programs that use XML PARSE**
  - One big difference between old parser and new parser is new parser can receive multiple events for the same content (ATTRIBUTE-CHARACTERS and CONTENT-CHARACTERS)
    - Use XML-INFORMATION to tell you when you are ‘done’ collecting data for an element or attribute
  - Before XML-INFORMATION it was difficult to collect the data
    - For *-CHARACTERS events, save data always ‘just in case’ there is more
    - When one of several new different events is encountered, then close out the old one

How to migrate – To Do Check list

- **Install Enterprise COBOL V5.1**
  - Set up default compiler options
  - The same settings will give you the same results
  - ADV, ARITH, AWO, NUMPROC, TRUNC, ZWB, etc
  - **Special Cases**
    - NUMPROC
      - *If currently use NUMPROC(MIG) you will have to migrate*
      - *COBOL applications always result in correct signs*
      - *If you have non-COBOL applications or sources of data*
        - *Use NUMPROC(NOPFD)*
    - If currently use OPT(FULL), use OPT(1 or 2) and STGOPT

How to migrate!

- **Which programs to compile first**
  - PERFORMANCE
    - Programs with lots of arithmetic will get most performance benefit from new code generator
    - The code generator is all new for COBOL V5. It is not the same code generator with improvements. So, some code will run faster than V4, some may run slower
  - Need for XML GENERATE and PARSE features
  - Need for Large data items

How to migrate!

- Questions?
- Good luck!