CONPARMX

Dissecting A New Approach

John D Consulting Inc.

Overview

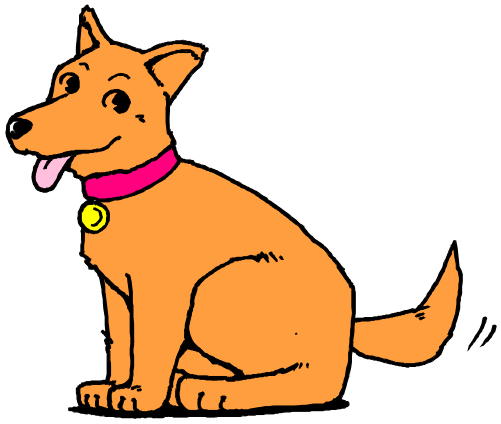
- With the introduction of the CONPARMX processor utility program, CA introduced an available solution to the specification of things such as compile or linkedit parameters.
- Adhering to the principles of Process Normalization, this allows for an easier definition and maintenance of variable processor groups using repeatable processors

Agenda

- Process Normalization Recap
- The Challenge
- CONPARMX Syntax
- Walkthrough
- Q&A

Process Normalization Recap

- An entire presentation unto itself
- Summary:
  - Normalized
    - COBOL, PLI, ASM
  - NOT normalized
    - COBSUB, COBDB2, COBMAIN…



A dog is a dog is a dog

Why Isn't It?

- Attributes of its handling, processing, and usage are mixed in with the name of the type
  - COBSUB - Subroutine
  - COBMAIN - Mainline
  - COBDB2 - DB2 calls
  - COBSDB2 - ???????
- In other words, there are many definitions for the same thing!

What Happens When You Normalize?

- Languages (that is, TYPES) are called only one thing
- COBOL is COBOL is COBOL
- Processor groups differentiate the manner in which individual elements are handled
- The concept can then be carried forward into Processors…

Agenda

- ☑ Process Normalization Recap
- The Challenge
- CONPARMX Syntax
- Walkthrough
- Q&A

The Challenge

- “The CONPARMX utility lets administrators reduce processor complexity and the number of processor groups. CONPARMX dynamically builds execution parameters for processor programs such as compilers or the linkage editor. The build process uses the symbols defined in the processor and one or more of the following groups of options, which may or may not be concatenated:
  - Default options
  - Processor group options
  - Element options
- Using this utility avoids the need to create and update large numbers of processors or processor groups with different input parameters.”

Put Another Way…

- Many generate processes require parameters to be specified in order to achieve desired results
  - Compile parms
  - Linkedit parms
  - Bind parms
- Traditional way
  - Processor Groups with unique combinations of parameters specified as a symbolic override
- Less traditional way
  - Symbolic string substitution
- Sneaky way
  - Specified as first line in the source (did you know that?) ☺

To illustrate…

- Consider the compile invocation step for a COBOL/LE processor:

```
//COB1     EXEC PGM=IGYCRCTL,
//              MAXRC=4,
//              PARM=('LIB,OPT,RES,X,MAP,TRUNC(BIN),OFF',
//              'NUMPROC(MIG),&COMPPARM')
//*
```

- Certain values are technical standards
- Other values are “developers choice” from within a pre-vetted and approved selection determined by processor group name

Processor	LIB	OPT	RES	CMPR2	DYN	X	MAP	AWO	TRUNC(BIN(	TRUNC(OPT(	OFF	NUMPROC(NOPFD(	NUMPROC(MIG(	APOST	RENT	Link NCAL	LET	XREF	LIST	TERM	REUS	DB2 QUOTE	QUOTESQL	HOST(COBOL(	HOST(COB2(	SOURCE	APOST	APOSTSQL	COPY2	CICS PRECOMPILE
CB2AA	x	x	x			x	x		x		x		x			x	x	x	x	x											
CB2AB	x	x	x			x	x		x		x		x	x		x	x	x	x	x											
CB2AC	x	x	x			x	x		x		x		x	x	x	x	x	x	x	x	x				x	x	x	x			
CB2AD	x	x	x			x	x		x		x		x			x	x	x	x	x	x										
CB2AE	x		x			x	x		x		x		x	x		x	x	x	x	x											
CB2AF	x	x	x			x	x		x		x		x	x	x	x	x	x	x	x	x										
CB2AG	x	x	x			x	x		x		x		x			x	x	x	x	x									x		
CB2AH	x	x	x			x	x		x		x		x	x	x	x	x	x	x	x	x				x	x	x	x	x		
CB2AI	x	x	x			x	x			x	x		x	x		x	x	x	x	x											
CB2AJ	x	x	x			x	x		x		x		x	x	x	x	x	x	x	x	x			x		x	x	x			
CB2AK	x	x	x			x	x		x		x	x		x		x	x	x	x	x											
CB2AL	x	x	x			x	x		x		x		x			x	x	x	x	x	x								x		
CB2AM	x	x	x		x	x	x		x		x	x		x		x	x	x	x	x											
CB2AN	x	x	x			x	x		x		x		x	x		x	x	x	x	x	x								x		
CB2AO	x	x	x			x	x		x	x	x		x	x		x	x	x	x	x	x	x	x		x	x			x		
CB2AP	x	x	x			x	x		x		x		x	x	x	x	x	x	x	x	x									x	
CB2AQ	x	x	x			x	x		x		x		x	x	x	x	x	x	x	x	x			x		x	x	x		x	
CB2AR	x	x	x	x	x	x	x	x	x		x		x	x	x	x	x	x	x	x	x										
CB2AS	x	x	x			x	x		x		x		x		x	x	x	x	x	x	x				x	x	x	x			

To illustrate (cont.)…

- Each processor group name requires Endevor administration to provide additional values to the symbolic override &COMPPARM symbolic according to the name of the processor group
  - Labour-intensive “at first”
    - Then seldom (if ever) has to be modified
  - Only need to define processor groups that make sense to the system that needs it
    - Not everyone needs everything
      - The grid is the “meta-view”
- Same for Link step but with symbolic &LKEDPARM

To illustrate (cont)…

- CB2AA
  - Needs no overrides
- CB2AB
  - &COMPPARM='APOST'
  - No override for &LKEDPARM
- CB2AC
  - &COMPARM='APOST,RENT'
  - &LKEDPARM='REUS'
- And so on through the various entries

Agenda

- ☑ Process Normalization Recap
- ☑ The Challenge
- CONPARMX Syntax
- Walkthrough
- Q&A

CONPARMX Syntax

```
//stepname EXEC PGM=CONPARMX,
//                  MAXRC=n,
// PARM=(parm1,'(parm2)',parm3,parm4,parm5,'(parm6)','parm7','parm8')
//PARMSDEF DD DSN=library.PRGRP,
//            MONITOR=COMPONENTS,
//            ALLOC=PMAP
```

- Each parm translates to a different purpose
  - Parm1 = program name
  - Parm2 = processor symbolic
  - Parm3 = default options member name
  - Parm4 = processor group options member name
  - Parm5 = element options member name
  - Parm6 = another processor symbolic entry
  - Parm7 = concatenation instruction
  - Parm8 = write to file instruction

Agenda

- ☑ Process Normalization Recap
- ☑ The Challenge
- ☑ CONPARMX Syntax
- Walkthrough
- Q&A

Walkthrough

- The easiest way to wrap one's head around CONPARMX is to go through a conversion
  - So let's perform one on the COBOL processor and the processor groups

```
//COB1     EXEC PGM=IGYCRCTL,
//              MAXRC=4,
//              PARM=('LIB,OPT,RES,X,MAP,TRUNC(BIN),OFF',
//              'NUMPROC(MIG),&COMPPARM')
//*
```

Step 1: Isolate the program being executed into PARM1

- In this case, the program is IGYCRCTL
- That changes the JCL in the processor to:

```
//COB1     EXEC PGM=CONPARMX,
//              MAXRC=4,
//              PARM=(IGYCRTL,........
//*
```

Step 2: Define the mandatory parameters

```
//COB1    EXEC PGM=IGYCRCTL,
//             MAXRC=4,
//             PARM=('LIB,OPT,RES,X,MAP,TRUNC(BIN),OFF',
//             'NUMPROC(MIG),&COMPPARM')
//*
```

- PARM2 are the “first” parameters you want the program to use.
  - In our case, it was LIB, OPT, RES, X, MAP, TRUNC(BIN), OFF, and NUMPROC(MIG).
  - We’ll define those in the PROC statement as a symbolic

Step 2 (cont):

```
//*****************************************************************
//*                                                               *
//* PROCESSOR NAME: GCOB22S                                       *
//* PURPOSE: COBOL2 STANDARD PROCESSOR                            *
//*                                                               *
//*****************************************************************
//GCOB22S PROC ADMNLIB='NDVLIB.ADMIN.STG6.LOADLIB',
//             COMPPARM='LIB,OPT,RES,X,MAP,TRUN(BIN),OFF,NUMPROC(MIG)',
:
:
:
```

Step 2 (cont):

- The CONPARMX step now looks like...

```
//COB1     EXEC PGM=CONPARMX,
//              MAXRC=4,
//              PARM=(IGYCRTL,'(&COMPPARM)',....
//*
```

Step 3: Defining the next parameters (and second guessing the PARM2 parameters!)

- PARM3 is provided as a way of providing “overall” parameters to the invoked program.
- CONPARMX will search within the library we specified earlier to find the member name specified in PARM3

```
//stepname EXEC PGM=CONPARMX,
//                  MAXRC=n,
// PARM=(parm1,'(parm2)',parm3,parm4,parm5,'(parm6)','parm7','parm8')
//PARMSDEF DD DSN=library.PRGRP,
//            MONITOR=COMPONENTS,
//            ALLOC=PMAP
```

Step 3 (cont.):

- Let's make the value of PARM3 to be $$$$DFLT.
- CONPARMX will now search dataset library.PRGRP for member $$$$DFLT.
  - NOTE: This will be an Endevor controlled library using PDS as the base, RBD, unencrypted/uncompressed dataset.
  - Member $$$$DFLT is contained as an element in Endevor.

Step 3 (cont.):

- What does member $$$$DFLT look like?
- A line exists for each program (ultimately) that will be leveraging CONPARMX.
  - However, at this point, there is only one line:

IGYCRCTL = ‘LIB,OPT,RES,X,MAP,TRUNC(BIN),OFF,’ +
‘NUMPROC(MIG),’

So now we go back and revisit what we did during Step 2...
Do we really need that COMPPARM symbolic anymore?

Step 3 (cont.):

- Remove &COMPPARM from the PROC statement and add $$$$DFLT as PARM3

```
//*****************************************************************
//*                                                               *
//* PROCESSOR NAME: GCOB22S                                       *
//* PURPOSE: COBOL2 STANDARD PROCESSOR                            *
//*                                                               *
//*****************************************************************
//GCOB22S PROC ADMNLIB='NDVLIB.ADMIN.STG6.LOADLIB',
:
:
:
//COB1    EXEC PGM=CONPARMX,
//             MAXRC=4,
//             PARM=(IGYCRCTL,,$$$$DFLT,....
//*
//PARMSDEF DD DSN=library.PRGRP,
//            MONITOR=COMPONENTS,
//            ALLOC=PMAP
:
```

Step 4: Continue with PARM4 for Processor Groups

- PARM4 follows the example set by PARM3
- Based on the provided processor group name, CONPARMX will now search dataset library.PRGRP for a match

```
//COB1     EXEC PGM=CONPARMX,
//              MAXRC=4,
//              PARM=(IGYCRCTL,,$$$$DFLT,&C1PRGRP,
//              ....
//*
//PARMSDEF DD DSN=library.PRGRP,
//            MONITOR=COMPONENTS,
//            ALLOC=PMAP
:
```

Step 4 (cont.):

- The dataset library.PRGRP will have a member for each processor group name defined.
- Contents of each member will identify program parameters needed for that group
  - CB2AA
    - Needs no overrides
  - CB2AB
    - &COMPPARM='APOST'
    - No override for &LKEDPARM
  - CB2AC
    - &COMPARM='APOST,RENT'
    - &LKEDPARM='REUS'

Step 4 (cont.):

- Since CB2AA has no additional parameters, it does not need to be defined.
  - CONPARMX will look for but ignore non-existent
- CB2AB's content would look like:
  - IGYCRTL = 'APOST,'
- We haven't included IEWL/Binder in our example, but the content of CB2AC would have 2 lines:
  - IGYCRTL = 'APOST,RENT'
  - IEWL = 'REUS'

Are You Done?

- PARM5 allows you to have element-specific PARMS
  - Personally, I would strongly discourage the use of this
    - If it's good enough for one, it's good enough to be made available for everyone through the declaration/definition of another processor group name
    - BUT we can still code it and "not use it"
- PARM6 through PARM8 are "optional" and not covered here today BUT need to be specified (best with "N")

```
//COB1     EXEC PGM=CONPARMX,
//              MAXRC=4,
//              PARM=(IGYCRCTL,,$$$$DFLT,&C1PRGRP,&C1ELEMENT,,'N','N')
//*
//PARMSDEF DD DSN=library.PRGRP,
//            MONITOR=COMPONENTS,
//            ALLOC=PMAP
:
```

What Have We Accomplished?

- Elimination of parameter overrides as part of defining new processor group names
  - Vastly simplifies introduction of new groups identifying unique combinations of program parameters
  - No longer necessary to visit every override in every system
    - Assured that processor group parameters are the same Endevor-wide
- Eased definition of processor groups
  - Names drive included parameters
  - Included parameters are now part of tracked Endevor elements
    - Endevor for Endevor administrators!
- One stop definition and override
  - The dataset library.PRGRP contains all your processor group names and what they override, independent of environment, system, etc.

Real Life Setup – All the pieces

```
--------------------------  ELEMENT SELECTION LIST  --------- Row 1 to 6 of 6
COMMAND ===>                                                  SCROLL ===> PAGE

                                                            --- DATES - More==>
   ELEMENT    TYPE    NS ENVIRON  S SYSTEM   SUBSYS   VVLL CURRENT GENERATE
   $$$$DFLT   PRGRP      DEVDMSB@ T SDSG     PROCESS  0101 08MAR16 08MAR16
   #COBCMP2   INCLUDE    DEVDMSB@ T SDSG     PROCESS  0105 08MAR16 08MAR16
   #GOBJ      INCLUDE    DEVDMSB@ T SDSG     PROCESS  0100 08MAR16 08MAR16
   CB2AB      PRGRP      DEVDMSB@ T SDSG     PROCESS  0100 08MAR16 08MAR16
   CB2AC      PRGRP      DEVDMSB@ T SDSG     PROCESS  0100 08MAR16 08MAR16
   GCOB22S    PROCESS    DEVDMSB@ T SDSG     PROCESS  0101 08MAR16 08MAR16
****************************** Bottom of data ********************************
```

Real Life Setup – Parm Type

```
DISPLAY ---------------------  TYPE DEFINITION  -----------------------------
COMMAND ===>
CURRENT ENV:  DEVDMSB@   STAGE ID:  T   SYSTEM:  SDSG       TYPE:  PRGRP
NEXT ENV   :  DEVDMSB@   STAGE ID:  S   SYSTEM:  SDSG       TYPE:  PRGRP

DESCRIPTION:  ===> JRD TEST PROCESSOR GROUP TYPE
UPDATED:           08MAR16 11:31 BY JQD000
             -----------------  ELEMENT OPTIONS  -------------------
DELTA FORMAT(F/R/I/L) ===> R  SOURCE LEN   ===> 80     ELE RECFM(N/F/V) ===> N
COMPRESS/ENCRYPT(Y/N) ===> N  COMPARE FROM ===> 1      DFLT PROC  ===> *NOPROC*
AUTO CONSOL (Y/N)     ===> Y  COMPARE TO   ===> 72     LANGUAGE   ===> JCL
CONSOL AT LVL  ===> 96        REGRESSION   ===> 75     PV/LB LANG ===> JCL
LVLS TO CONSOL ===> 50        REG SEV(I/W/C/E)  ===> W DATA FORMAT(T/B) ===> T
HFS RECFM(COMP/CR/CRLF/CRNL/F/LF/NL/V)     ===> NL     FILE EXT   ===>
         -----------------  COMPONENT LIST OPTIONS  ------------------
FWD/REV DELTA(F/R)    ===> R  AUTO CONSOL (Y/N)      Y CONSOL AT LVL       96
                                                       LVLS TO CONSOL      49
         -------------------------  LIBRARIES  ----------------------
  BASE/IMAGE LIBRARY ===> SIMD.PROD.&C1EN..&C1SY..&C1ST..&C1TY
  DELTA LIBRARY      ===> SIMD.PROD.DEVDMSB@.TEST.SDSG.DELTA.CL
  INCLUDE LIBRARY    ===>
  SOURCE O/P LIBRARY ===>
    EXPAND INCLUDES(Y/N)  ===> N
```

Real Life Setup – Initial Processor

```
 Menu  Utilities  Compilers  Help
──────────────────────────────────────────────────────────────────────────────
BROWSE    JQD000.GCOB22S#.S1.EMVSJ.PROCESS          Line 00000020 Col 001 080
Command ===>                                                 Scroll ===> CSR
+0100  //*********************************************************************
+0100  //*                                                                   *
+0100  //* PROCESSOR NAME: GCOB22S                                           *
+0100  //* PURPOSE: COBOL2 STANDARD PROCESSOR                                *
+0100  //*                                                                   *
+0100  //*********************************************************************
+0100  //GCOB22S PROC   ADMNLIB='NDVLIB.ADMIN.STG6.LOADLIB',
+0100  //               COMPPARM='',
+0100  //               COMCOP1='SIMD.PROD.COMMON.&C1ST..COPYLIB',
+0100  //               COPYLIBA='SIMD.PROD.&C1SY..&C1ST..COPYLIB',
+0100  //               EXPINC=N,
+0100  //               LKDPARM='',
+0100  //               LNKLIB='SYS1.LINKLIB',
+0100  //               LNKPGM=IEWL,
+0100  //               MONITOR=COMPONENTS,
+0100  //               SYSOUT='*',
+0100  //               WRKUNIT=VIO
+0100  //*
+0100  //*********************************************************************
+0100  //*   GENER THE PROGRAM TO A GENERIC READABLE LIBRARY                 *
+0100  //*********************************************************************
+0100  //CONW    EXEC PGM=CONWRITE,
+0100  //             PARM='EXPINCL(&EXPINC)'
+0100  //ELMOUT    DD DSN=&&PROGRAM,
+0100  //             DISP=(NEW,PASS),
+0100  //             SPACE=(TRK,(100,100),RLSE),
+0100  //             UNIT=&WRKUNIT,
+0100  //             DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
+0100  //             MONITOR=&MONITOR
+0100  //*
+0100  //*********************************************************************
+0100  //*   BEGIN STANDARD COBOL II COMPILER                                *
+0100  //*********************************************************************
+0100         ++INCLUDE #COBCMP2
+0100  //*
+0100  //**END
****************************** Bottom of Data ********************************
```

Real Life Setup – Include Member Part 1

```
:
:
//*********************************************************************
//*    COB1    EXECUTE COBOL COMPILER                                 *
//*********************************************************************
//*
//COB1    EXEC PGM=IGYCRCTL,
//             MAXRC=4,
//             PARM=('LIB,OPT,RES,X,MAP,TRUNC(BIN),OFF',
//             'NUMPROC(MIG),&COMPPARM')
//*
//SYSLIB    DD DSN=SYS2.SSC.COPYLIB,
//             DISP=(SHR,PASS),
//             MONITOR=&MONITOR
//          DD DSN=&COMCOP1,
//             DISP=(SHR,PASS),
//             MAP=LMAP,
//             MONITOR=&MONITOR
//          DD DSN=&COPYLIBA,
//             DISP=(SHR,PASS),
//             MAP=LMAP,
//             MONITOR=&MONITOR
//*
//SYSIN     DD DSN=&&PROGRAM,
//             DISP=SHR
//*
//SYSLIN    DD DSN=&&SYSLIN,
//             DISP=(OLD,PASS),
:
:
```

Real Life Setup – Include Member Part 2

```
:
:
//*
//*********************************************************************
//*            EXECUTE LINKAGE EDITOR                                 *
//*********************************************************************
//*
//LKNCAL  EXEC PGM=&LNKPGM,
//             COND=(5,LT),
//             MAXRC=04,
//             PARM='NCAL,TERM,LET,XREF,LIST,&LKDPARM'
//*
//STEPLIB   DD DSN=&LNKLIB,
//             DISP=SHR
//SYSLIB    DD DSN=SYS2.COBVS.LOAD.LIBRARY,DISP=SHR
//          DD DSN=SYS2.ISPF.LINK.SISPLOAD,DISP=SHR
//          DD DSN=SYS2.SSC.LOAD,DISP=SHR
//*
//SYSLIN    DD DSN=&&SYSLIN,
//             DISP=(OLD,KEEP)
//*         DD DSN=SYS2.SETSSI,
//*            DISP=(SHR,PASS)
//SYSLMOD   DD DSN=SIMD.PROD.&C1SY..&C1ST..OBJLIB(&C1ELEMENT),
//             DISP=SHR,
//             FOOTPRNT=CREATE,
//             MONITOR=&MONITOR
:
:
```

Real Life Setup – Altered Include Member Part 1

```
:
:
//*********************************************************************
//*    COB1    EXECUTE COBOL COMPILER                                 *
//*********************************************************************
//*
//COB1    EXEC PGM=CONPARMX,
//             MAXRC=4,
//             PARM=(IGYCRCTL,,$$$$DFLT,&C1PRGRP,&C1ELEMENT,,'N','N')
//PARMSDEF  DD DSN=&PARMLIB,
//             MONITOR=&MONITOR,
//             ALLOC=LMAP,
//             DISP=SHR
//*
//SYSLIB    DD DSN=SYS2.SSC.COPYLIB,
//             DISP=(SHR,PASS),
//             MONITOR=&MONITOR
//          DD DSN=&COMCOP1,
//             DISP=(SHR,PASS),
//             ALLOC=LMAP,
//             MONITOR=&MONITOR
//          DD DSN=&COPYLIBA,
//             DISP=(SHR,PASS),
//             ALLOC=LMAP,
//             MONITOR=&MONITOR
//*
//SYSIN     DD DSN=&&PROGRAM,
//             DISP=SHR
//*
:
:
```

Real Life Setup – Altered Include Member Part 2

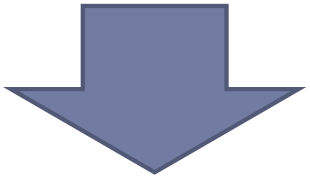
```
:
:
//*
//*******************************************************************
//*             EXECUTE LINKAGE EDITOR                              *
//*******************************************************************
//*
//LKNCAL  EXEC PGM=CONPARMX,
//             COND=(5,LT),
//             MAXRC=04,
//             PARM=(&LNKPGM,,$$$$DFLT,&C1PRGRP,&C1ELEMENT,,'N','N')
//PARMSDEF  DD DSN=&PARMLIB,
//             MONITOR=&MONITOR,
//             ALLOC=LMAP,
//             DISP=SHR
//*
//STEPLIB   DD DSN=&LNKLIB,
//             DISP=SHR
//SYSLIB    DD DSN=SYS2.COBVS.LOAD.LIBRARY,DISP=SHR
//          DD DSN=SYS2.ISPF.LINK.SISPLOAD,DISP=SHR
//          DD DSN=SYS2.SSC.LOAD,DISP=SHR
:
:
```

Real Life Setup – Symbolic Overrides?

```
DISPLAY ----------------  PROCESSOR GROUP SYMBOLICS  --------- Row 1 to 9 of 9
COMMAND ===>                                                  SCROLL ===> PAGE

CURRENT ENV:  DEVDMSB@   STAGE ID:  T   SYSTEM:  SDSG       TYPE:  COBOL

PROCESSOR GROUP: CB2AC           PROCESSOR:      GCOB22S
DESCRIPTION:     COBOL GEN
LOAD LIBRARY:    SIMD.PROD.DEVDMSB@.TEST.SDSG.LOADPRC
DEFAULT  VALUES ARE INDICATED BY  -
OVERRIDE VALUES ARE INDICATED BY  O

SYMBOLIC -/O VALUE
COMCOP1    -  SIMD.PROD.COMMON.&C1ST..COPYLIB
COPYLIBA   -  SIMD.PROD.&C1SY..&C1ST..COPYLIB
EXPINC     -  N
LNKLIB     -  SYS1.LINKLIB
LNKPGM     -  IEWL
MONITOR    -  COMPONENTS
PARMLIB    -  SIMD.PROD.DEVDMSB@.STST.PRGRP
SYSOUT     -  *
WRKUNIT    -  VIO
****************************** Bottom of data ********************************
```

Real Life Setup - PRGRP Type Contents

```
--------------------------  ELEMENT SELECTION LIST  --------- Row 1 to 3 of 3
COMMAND ===>                                                  SCROLL ===> PAGE

                                                           --- DATES - More==>
   ELEMENT    TYPE    NS ENVIRON  S SYSTEM   SUBSYS   VVLL CURRENT GENERATE
   $$$$DFLT   PRGRP      DEVDMSB@ T SDSG     PROCESS  0101 08MAR16 08MAR16
   CB2AB      PRGRP      DEVDMSB@ T SDSG     PROCESS  0100 08MAR16 08MAR16
   CB2AC      PRGRP      DEVDMSB@ T SDSG     PROCESS  0100 08MAR16 08MAR16
****************************** Bottom of data ********************************
```



```
 Menu  Functions  Confirm  Utilities  Help
_____________________________________________________________________________
BROWSE            SIMD.PROD.DEVDMSB@.SDSG.TEST.PRGRP        Row 00001 of 00003
Command ===>                                                  Scroll ===> PAGE
           Name     Prompt       Size   Created          Changed          ID
_________ $$$$DFLT
_________ CB2AB
_________ CB2AC
          **End**
```

Real Life Setup – Browse members

```
 Menu  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
 BROWSE    SIMD.PROD.DEVDMSB@.SDSG.TEST.PRGRP($$$$DFL Line 00000000 Col 001 080
 Command ===>                                                 Scroll ===> CSR
******************************** Top of Data ********************************
IGYCRCTL = 'LIB,OPT,RES,X,MAP,TRUNC(BIN),OFF,NUMPROC(MIG),'
IEWL = 'NCAL,LET,XREF,LIST,TERM,'
******************************** Bottom of Data *****************************
```

$$$$DFLT

```
 Menu  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
 BROWSE    SIMD.PROD.DEVDMSB@.SDSG.TEST.PRGRP(CB2AB)  Line 00000000 Col 001 080
 Command ===>                                                 Scroll ===> CSR
******************************** Top of Data ********************************
IGYCRCTL = 'APOST,'
******************************** Bottom of Data *****************************
```

CB2AB

```
 Menu  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
 BROWSE    SIMD.PROD.DEVDMSB@.SDSG.TEST.PRGRP(CB2AC)  Line 00000000 Col 001 080
 Command ===>                                                 Scroll ===> CSR
******************************** Top of Data ********************************
IGYCRCTL = 'APOST,RENT,'
IEWL = 'REUS,'
******************************** Bottom of Data *****************************
```

CB2AC

If We Browse Components (BX) Against A Generated Program…

```
 Menu  Utilities  Compilers  Help
 ──────────────────────────────────────────────────────────────────────────────
 BROWSE    JQD000.C1UEXT07.S1.EMVSJ.COBOL             Line 00000064 Col 001 080
 Command ===>                                                 Scroll ===> HALF

---------------------------   INPUT COMPONENTS   -------------------------------


STEP: COB1     DD=PARMSDEF VOL=USJ096 DSN=SIMD.PROD.DEVDMSB@.SDSG.TEST.PRGRP

             MEMBER     VVLL   DATE   TIME  SYSTEM   SUBSYS   ELEMENT    TYPE
 +0112       $$$$DFLT   0101  08MAR16 11:50 SDSG     PROCESS  $$$$DFLT   PRGRP
 +0112       CB2AB      0100  08MAR16 11:42 SDSG     PROCESS  CB2AB      PRGRP

STEP: LKNCAL   DD=PARMSDEF VOL=USJ096 DSN=SIMD.PROD.DEVDMSB@.SDSG.TEST.PRGRP

             MEMBER     VVLL   DATE   TIME  SYSTEM   SUBSYS   ELEMENT    TYPE
%+0117       $$$$DFLT   0101  08MAR16 11:50 SDSG     PROCESS  $$$$DFLT   PRGRP
%+0117       CB2AB      0100  08MAR16 11:42 SDSG     PROCESS  CB2AB      PRGRP


---------------------------  OUTPUT COMPONENTS  -------------------------------


STEP: LKNCAL   DD=SYSLMOD  VOL=USJ091 DSN=SIMD.PROD.SDSG.TEST.OBJLIB

             MEMBER     VVLL   DATE   TIME  SYSTEM   SUBSYS   ELEMENT    TYPE
%+0117       C1UEXT07   0100  10MAR16 10:20 SDSG     PROCESS  C1UEXT07   COBOL
****************************** Bottom of Data ********************************
```

Agenda

- ☑ Process Normalization Recap
- ☑ The Challenge
- ☑ CONPARMX Syntax
- ☑ Walkthrough
- Q&A