COBOL V5 Migration Strategies

Dave Kartzman, Compuware

Email: David.Kartzman@Compuware.com

Background

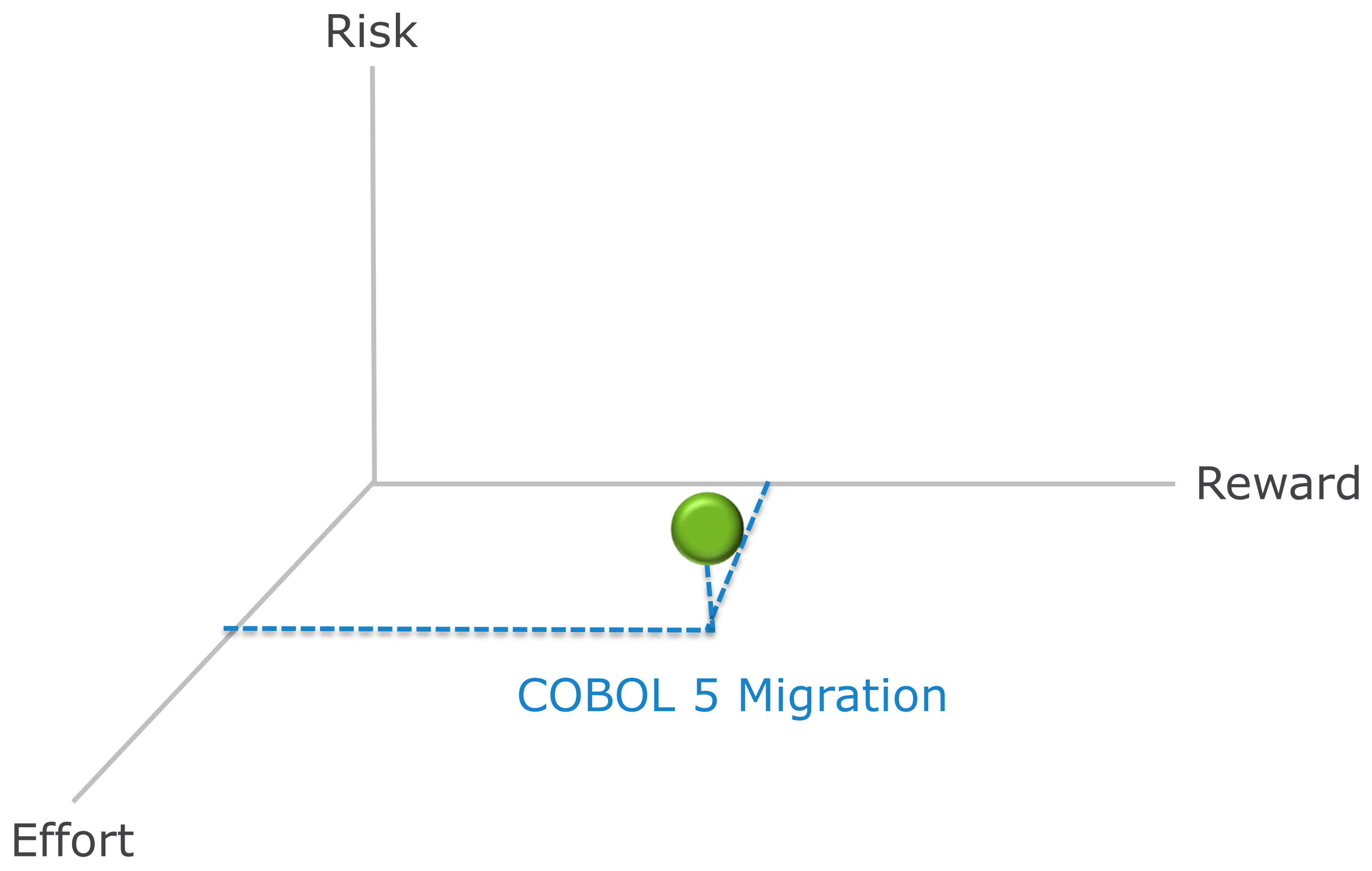
COBOL V5: CliffsNotes

- Significant rewrite by IBM
  - leverage Code Generator code used in Java and C/C++
  - catch up with z/OS hardware improvements
  - aggressive optimization (CPU and memory intensive compile)
  - (more or less) compatible with previous COBOL compilers
  - (more or less) can run combined with older COBOL executables
- Runtime Performance improvements
  - We see 5-7% at our customers (highs in the 9-11% range)
  - IBM says up to 20% or more in certain cases.

COBOL V5 Win-Win

For **customers**
- possible budget savings
- software catches up to hardware
- IBM commitment

For **IBM**
- common code paths
- reinvigorate a significant money maker
- growth path

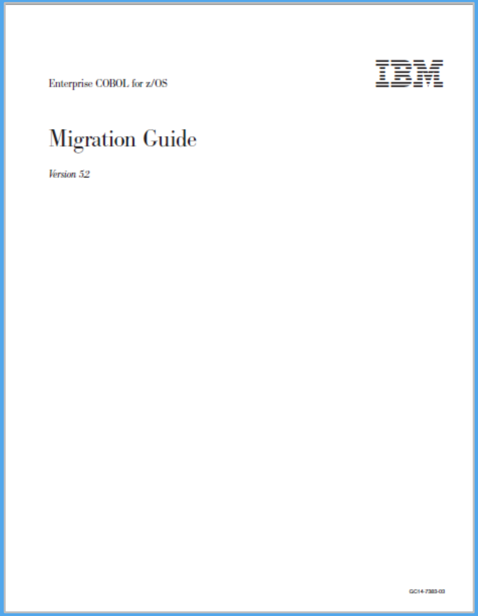

Risk
Reward
Effort
COBOL 5 Migration

Migration

First Steps

1. Read these Books!

Migration Guide –
GC14-7383-03
Programming Guide
SC14-7382-03



Enterprise COBOL for z/OS

IBM

Migration Guide

Version 5.2

GC14-7383-03

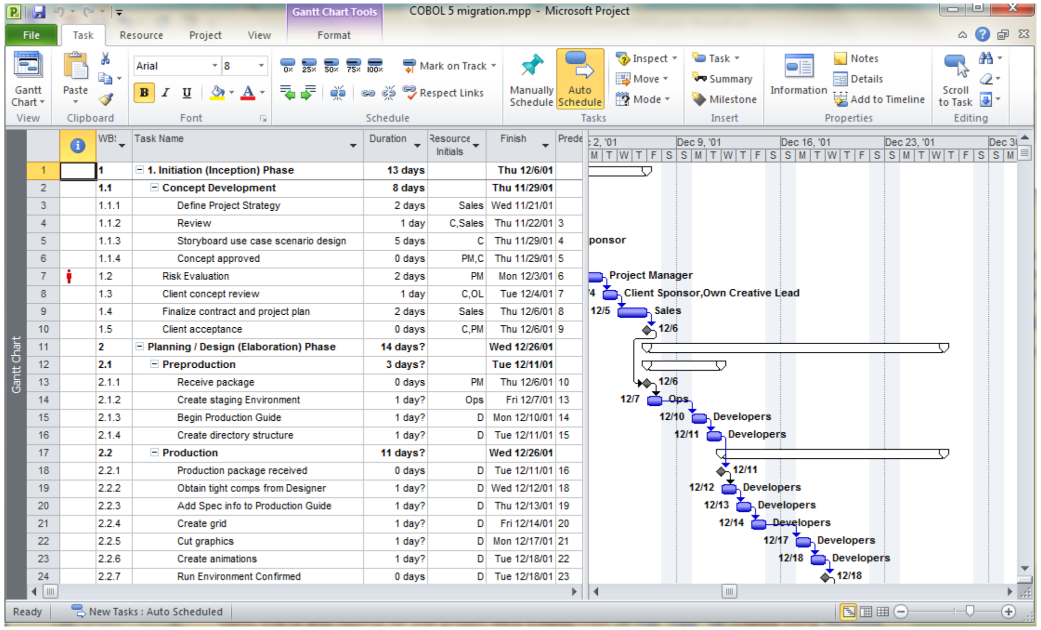
For Systems programmers:
Performance Guide – COBOL 5.1.1- Paper

http://www-01.ibm.com/support/docview.wss?uid=swg27042388&aid=1

First Steps

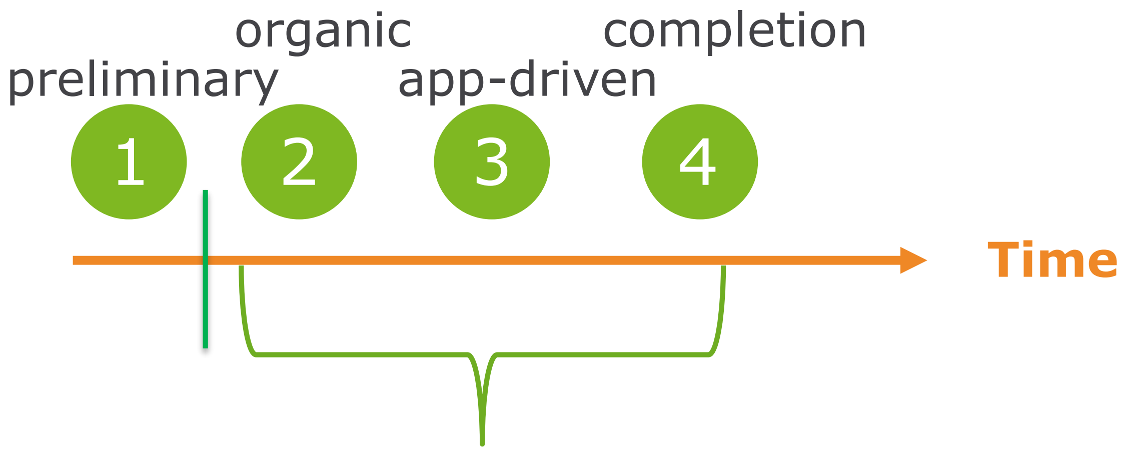
1. Read Migration Guide

2. **Create a project!**
   a) **Scope**
   b) **Effort**
   c) **Expectations**



	WBS	Task Name	Duration	Resource Initials	Finish	Prede
1	**1**	**1. Initiation (Inception) Phase**	**13 days**		**Thu 12/6/01**	
2	**1.1**	**Concept Development**	**8 days**		**Thu 11/29/01**	
3	1.1.1	Define Project Strategy	2 days	Sales	Wed 11/21/01	
4	1.1.2	Review	1 day	C,Sales	Thu 11/22/01	3
5	1.1.3	Storyboard use case scenario design	5 days	C	Thu 11/29/01	4
6	1.1.4	Concept approved	0 days	PM,C	Thu 11/29/01	5
7	1.2	Risk Evaluation	2 days	PM	Mon 12/3/01	6
8	1.3	Client concept review	1 day	C,OL	Tue 12/4/01	7
9	1.4	Finalize contract and project plan	2 days	Sales	Thu 12/6/01	8
10	1.5	Client acceptance	0 days	C,PM	Thu 12/6/01	9
11	**2**	**Planning / Design (Elaboration) Phase**	**14 days?**		**Wed 12/26/01**	
12	**2.1**	**Preproduction**	**3 days?**		**Tue 12/11/01**	
13	2.1.1	Receive package	0 days	PM	Thu 12/6/01	10
14	2.1.2	Create staging Environment	1 day?	Ops	Fri 12/7/01	13
15	2.1.3	Begin Production Guide	1 day?	D	Mon 12/10/01	14
16	2.1.4	Create directory structure	1 day?	D	Tue 12/11/01	15
17	**2.2**	**Production**	**11 days?**		**Wed 12/26/01**	
18	2.2.1	Production package received	0 days	D	Tue 12/11/01	16
19	2.2.2	Obtain tight comps from Designer	1 day?	D	Wed 12/12/01	18
20	2.2.3	Add Spec info to Production Guide	1 day?	D	Thu 12/13/01	19
21	2.2.4	Create grid	1 day?	D	Fri 12/14/01	20
22	2.2.5	Cut graphics	1 day?	D	Mon 12/17/01	21
23	2.2.6	Create animations	1 day?	D	Tue 12/18/01	22
24	2.2.7	Run Environment Confirmed	0 days	D	Tue 12/18/01	23

Project Timeline



IBM grace period for running two COBOL versions[1]

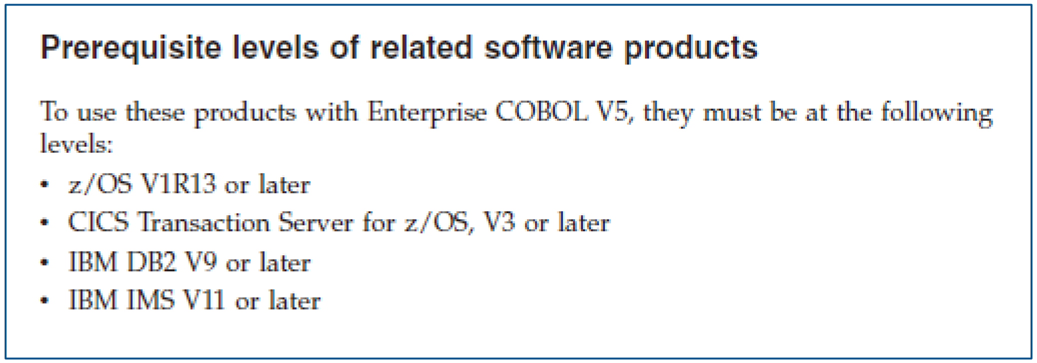
[1] *talk with IBM*

1 Preliminary Work

- Simplify migration by completing these items beforehand
- Do not order COBOL v5.2 until you're happy with the preliminary work!

1 Preliminary Work

a Get current



Prerequisite levels of related software products

To use these products with Enterprise COBOL V5, they must be at the following levels:

- z/OS V1R13 or later
- CICS Transaction Server for z/OS, V3 or later
- IBM DB2 V9 or later
- IBM IMS V11 or later

1 Preliminary Work

a Get current

b Complete LE runtime migration

1 Preliminary Work

a Get current

b Complete LE runtime migration

c Convert Load libraries to PDSE*

1 Preliminary Work

a Get current

b Complete LE runtime migration

c Convert Load libraries to PDSE

d SCM product to drive all compiles

1 Preliminary Work

e Order and install COBOL v5.2 and apply latest PTFs!

[1] Likely to involve many PTF’s – don’t forget ISV’s too.

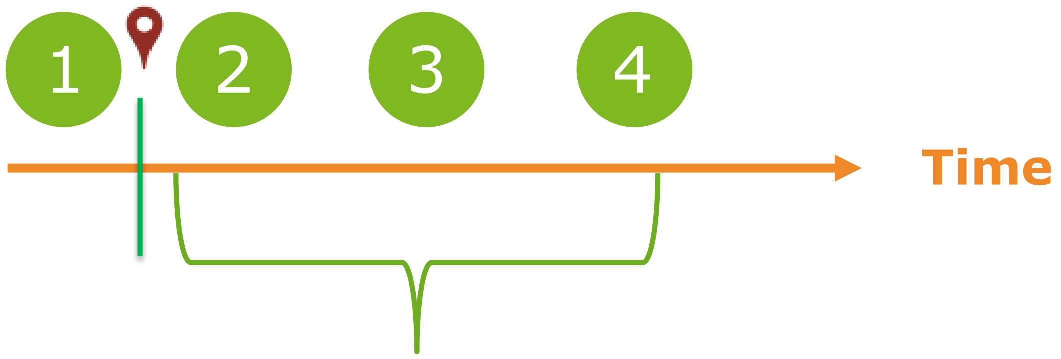
1 Preliminary Work

e Order and install COBOL v5

f Implement COBOL v5.2 in SCM driven compiles

- JCL changes
- decide on certain compile options
    - ARCH
    - NUMPROC
    - OPT
    - SSRANGE
    - STGOPT

Project Timeline



2 Organic changes

a Pilot project

2 Organic changes

a Pilot project

b Migrate programs as they come up for changes.

- Bug fixes
- Active development
- How much added regression testing?

2 Organic changes

a Pilot project

b Migrate programs as they come up for changes.

- Bug fixes
- Active development
- How much added regression testing?

c Publish results

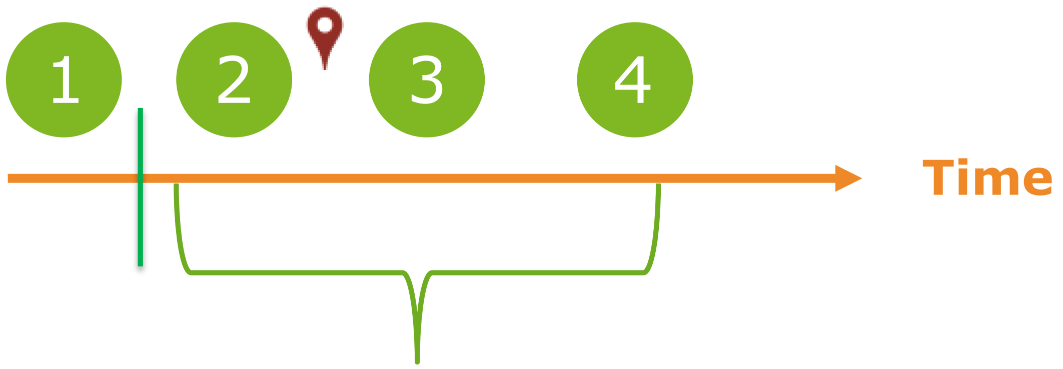
- CPU savings
- % complete (total, by application)

2 Organic changes

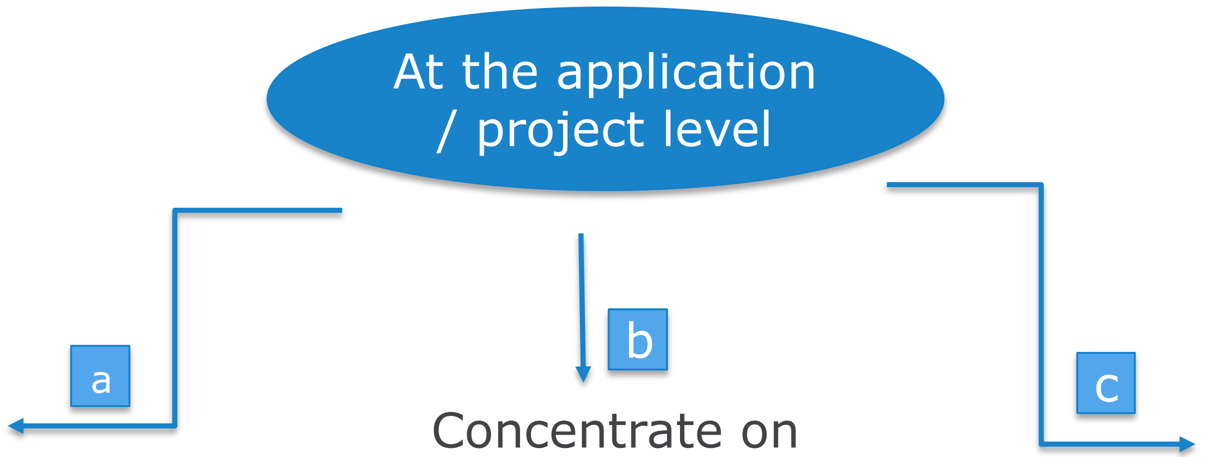
Challenges

- Expect “devil is in the details” type problems at this point.
- Exception criteria? Who decides the exceptions?
- When to move to step 3? What about code freeze time periods?

Project Timeline



3 Application Groups drive speed of migration

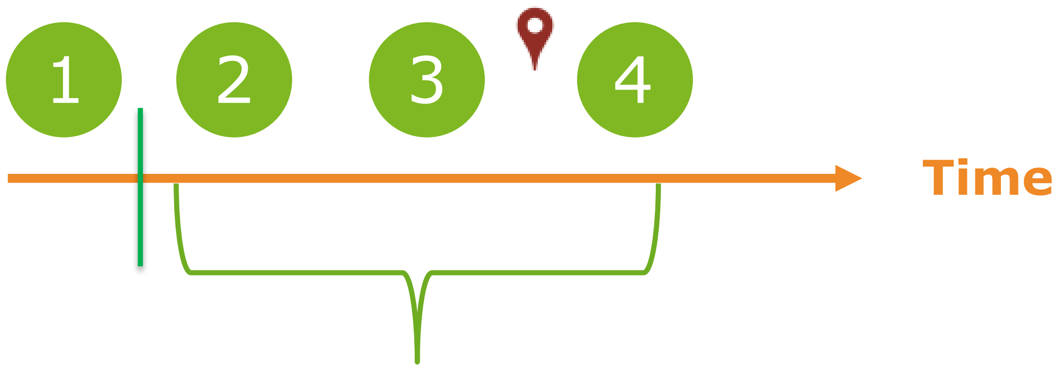


Continue to migrate as changes come up

Concentrate on performance opportunities

Convert entire application or application component

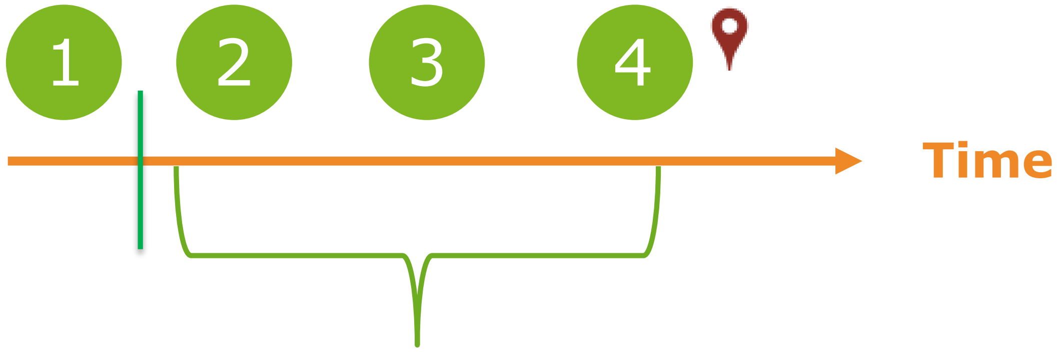
Project Timeline



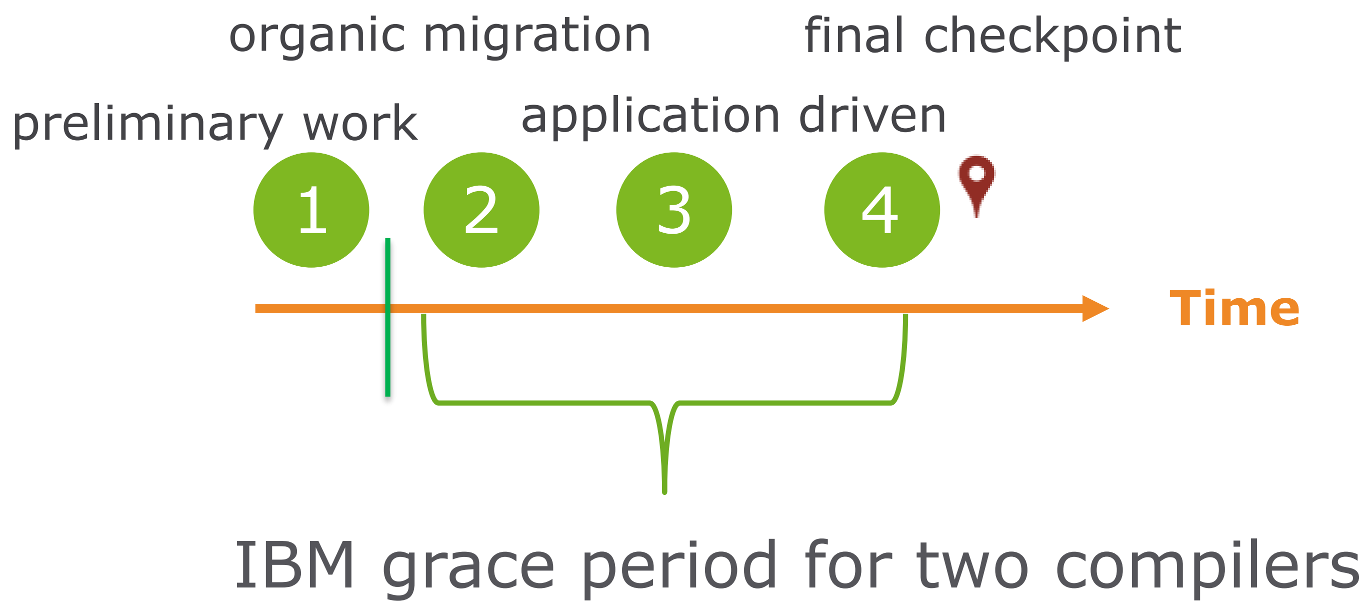
4 Final checkpoint

- Confident of conversion effort – willing to retire the older COBOL
- Go through one code freeze cycle?

Project Timeline



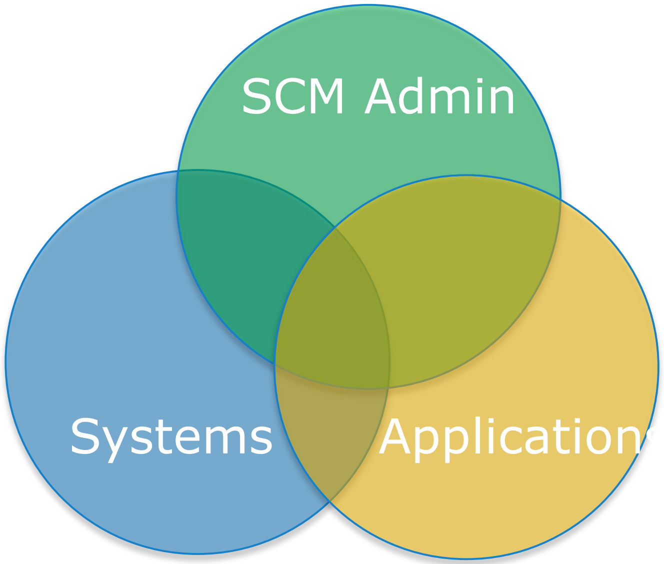
Project Timeline



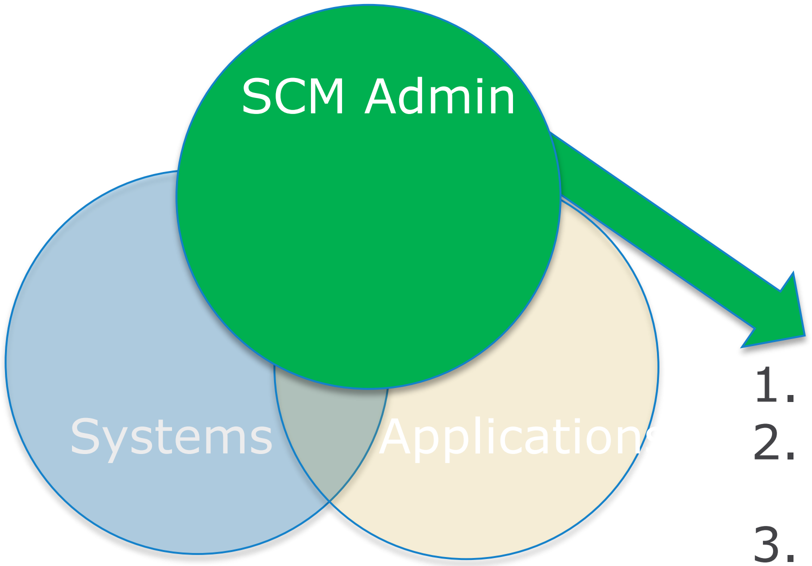
5 Project Analysis

- Did the project meet expectations?
  - CPU savings
  - $ savings
  - effort
- What about the remaining COBOL programs?

Project by Group

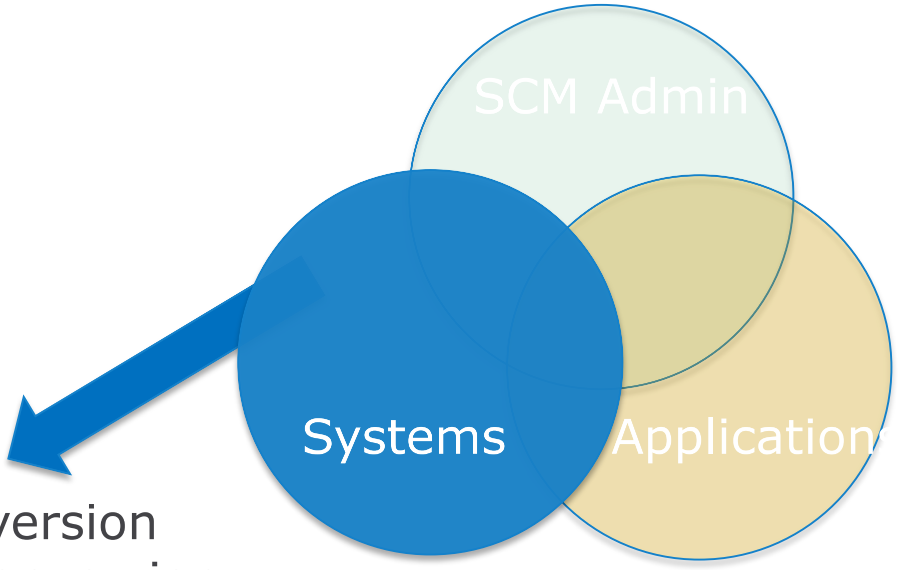


Project by Group



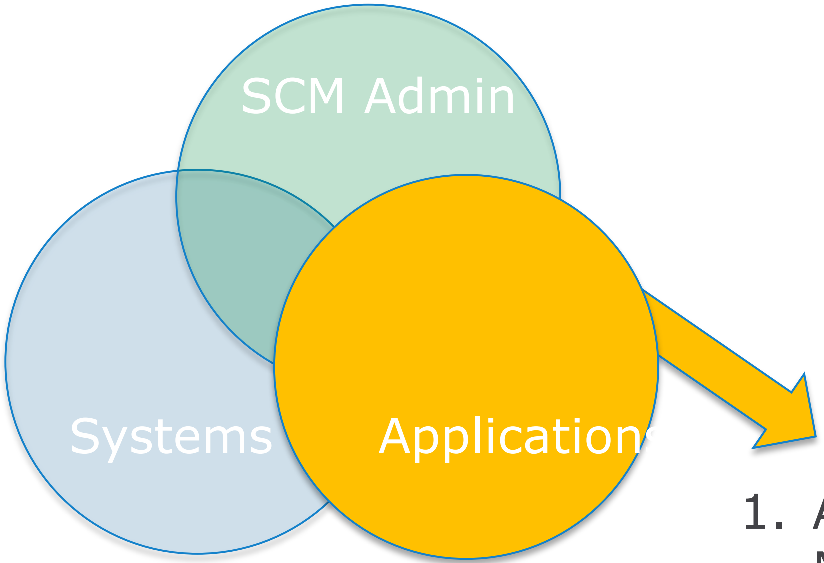
1. Compile JCL
2. Compile options
    a) At each promotion level
3. No-go gates

Project by Group



1. LE Conversion
2. PDSE conversion
3. Currency

Project by Group



1. Application Migration
2. Regression testing

Considerations

Compile Options

Option	Consideration
OPT(n)	Recommend OPT(0) during development; OPT(2) for last compile.
ARCH	Lowest common denominator
SSRANGE	not in production
NUMPROC	PFD. If NOPFD, why?
RULES	Helps identify performance and coding issues

Optimization

- OPTIMIZE(0) specifies limited optimizations, which result in the shortest compilation time. TEST option is not needed to use Xpediter for full debugging capability
- OPTIMIZE(1) specifies optimizations that improve application runtime performance. Optimizations include:
  - basic inlining
  - simplification of complex operations into equivalent simpler operations
  - removal of some unreachable code and block rearrangement.
  - Compiling with TEST will allow full debugging
- OPTIMIZE(2) specifies further optimizations:
  - more aggressive simplifications and instruction scheduling.
  - When the TEST option is specified, some debug capabilities are available.

Older Environments

Environment	Consideration
OS/VS COBOL	Doesn’t mix with COBOL 5
VS COBOL II	If NORES – cannot mix with COBOL 5
Storage Eye-catchers	May be removed (STGOPT) during COMPILE.
AMODE(24)	Part of migration – to remove this restriction?

Abend Personality

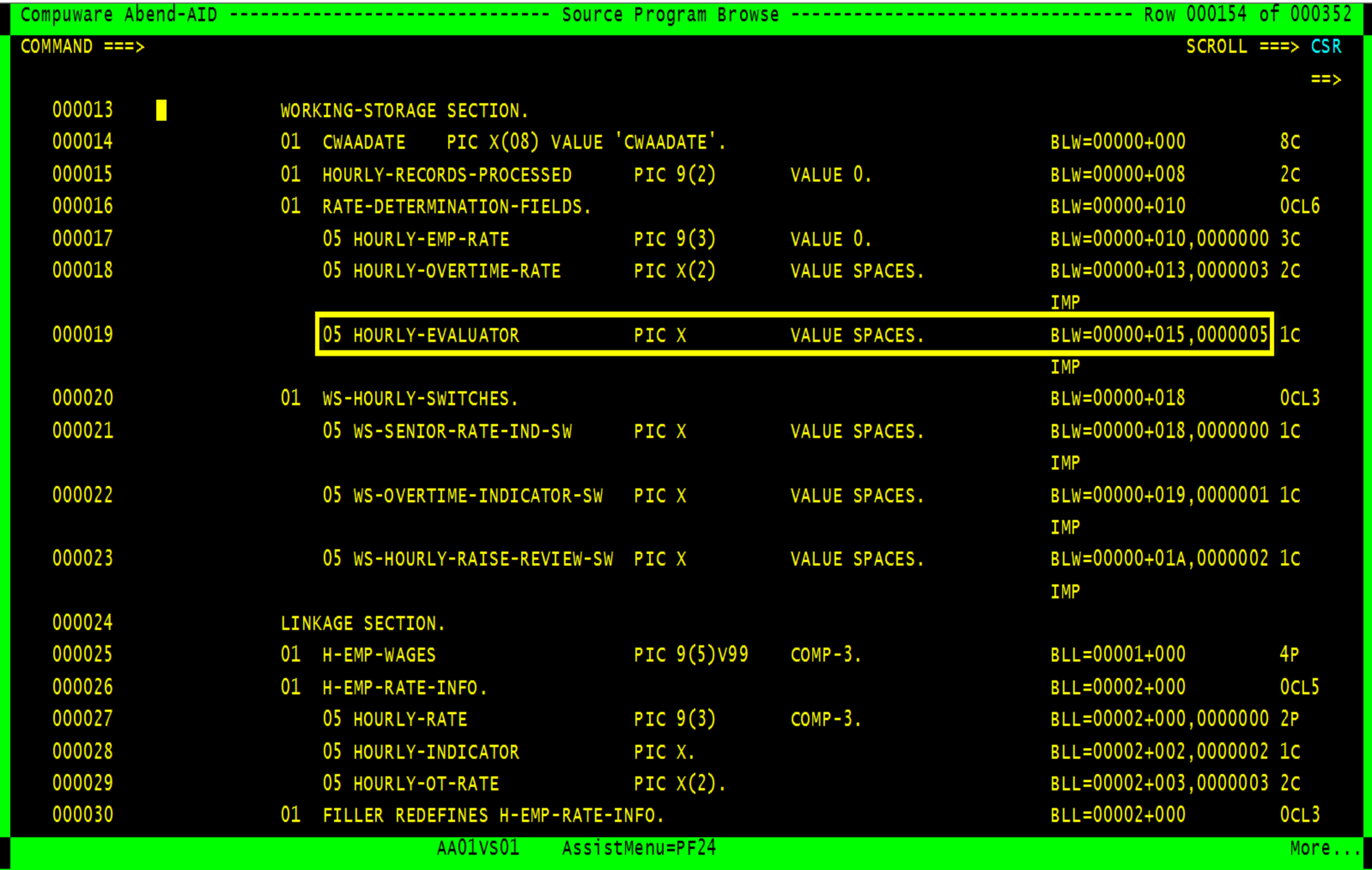
Index over-runs:

- May change from S0C7 to S0C4
- Over-run itself may corrupt / re-corrupt / un-corrupt index
  - Removes forensics
  - Applications may reach out to systems to help solve

New IBM Compiler output

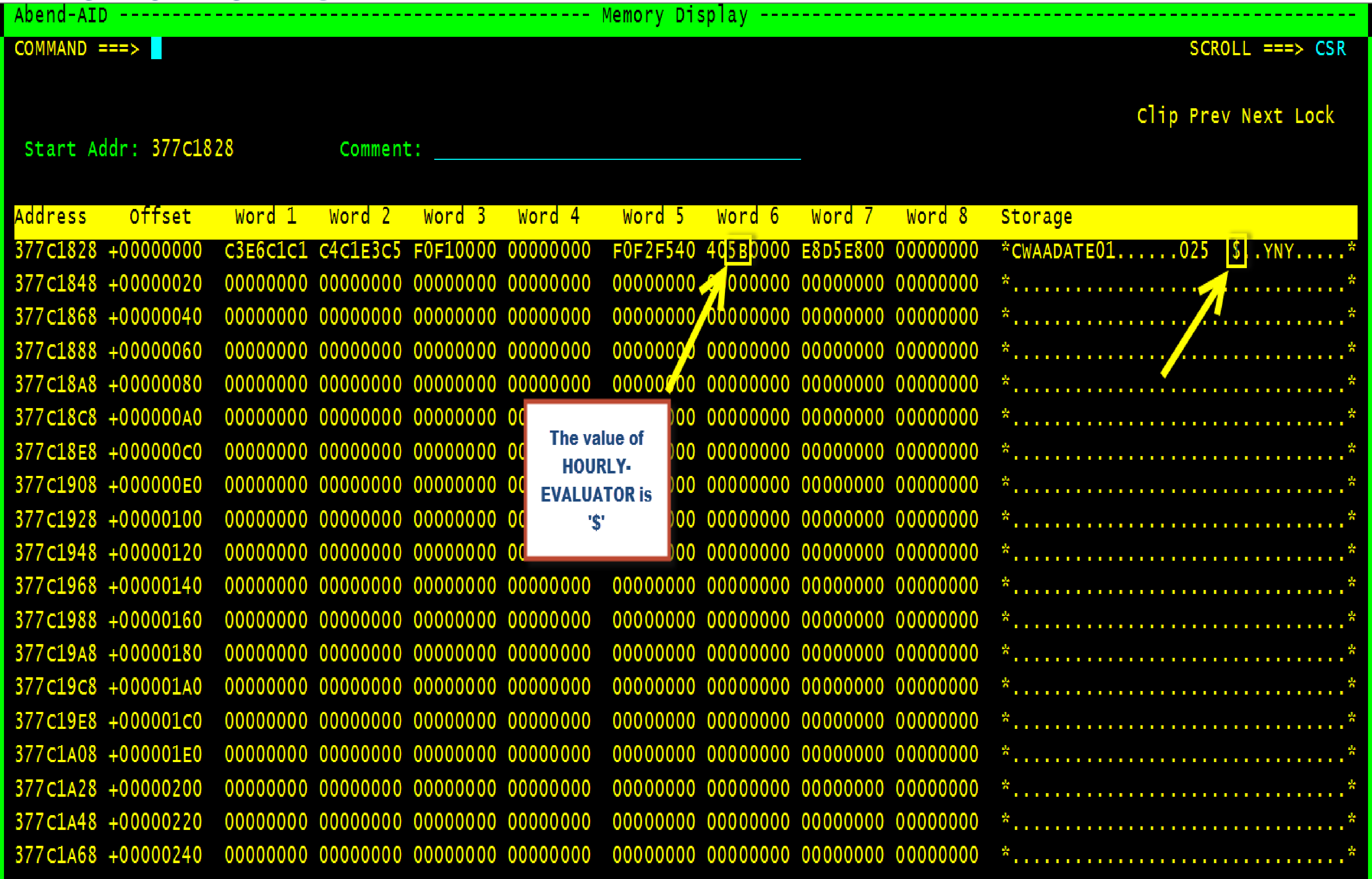
- Previous versions of the compiler output would display the BLL, BLF and BLW cells for each of the variables in the File Section, Program Storage Section and Linkage Section
- The new compiler output does not display the offset from the BLW pointer anymore. All 77, 88 and 01 group level variable names are located in the 'Static Map'.
  - Elementary variables are not listed in the static map. In the Working-Storage area, the elementary level variables are denoted by an offset from the group level
    - To find the value of the variable, one must find the location of the group level in the static map and add the offset of the variable from the group level (found in the program-storage section)

Pre 5.2 Compiler Listing



```
Compuware Abend-AID ---------------------------- Source Program Browse ------------------------------ Row 000154 of 000352
COMMAND ===>                                                                                            SCROLL ===> CSR
                                                                                                                    ==>
   000013              WORKING-STORAGE SECTION.
   000014              01  CWAADATE    PIC X(08) VALUE 'CWAADATE'.                  BLW=00000+000         8C
   000015              01  HOURLY-RECORDS-PROCESSED     PIC 9(2)      VALUE 0.      BLW=00000+008         2C
   000016              01  RATE-DETERMINATION-FIELDS.                               BLW=00000+010         0CL6
   000017                  05 HOURLY-EMP-RATE           PIC 9(3)      VALUE 0.      BLW=00000+010,0000000 3C
   000018                  05 HOURLY-OVERTIME-RATE      PIC X(2)      VALUE SPACES. BLW=00000+013,0000003 2C
                                                                                    IMP
   000019                  05 HOURLY-EVALUATOR          PIC X         VALUE SPACES. BLW=00000+015,0000005 1C
                                                                                    IMP
   000020              01  WS-HOURLY-SWITCHES.                                      BLW=00000+018         0CL3
   000021                  05 WS-SENIOR-RATE-IND-SW     PIC X         VALUE SPACES. BLW=00000+018,0000000 1C
                                                                                    IMP
   000022                  05 WS-OVERTIME-INDICATOR-SW  PIC X         VALUE SPACES. BLW=00000+019,0000001 1C
                                                                                    IMP
   000023                  05 WS-HOURLY-RAISE-REVIEW-SW PIC X         VALUE SPACES. BLW=00000+01A,0000002 1C
                                                                                    IMP
   000024              LINKAGE SECTION.
   000025              01  H-EMP-WAGES                  PIC 9(5)V99   COMP-3.       BLL=00001+000         4P
   000026              01  H-EMP-RATE-INFO.                                         BLL=00002+000         0CL5
   000027                  05 HOURLY-RATE               PIC 9(3)      COMP-3.       BLL=00002+000,0000000 2P
   000028                  05 HOURLY-INDICATOR          PIC X.                      BLL=00002+002,0000002 1C
   000029                  05 HOURLY-OT-RATE            PIC X(2).                   BLL=00002+003,0000003 2C
   000030              01  FILLER REDEFINES H-EMP-RATE-INFO.                        BLL=00002+000         0CL3
                                  AA01VS01     AssistMenu=PF24                                                   More...
```

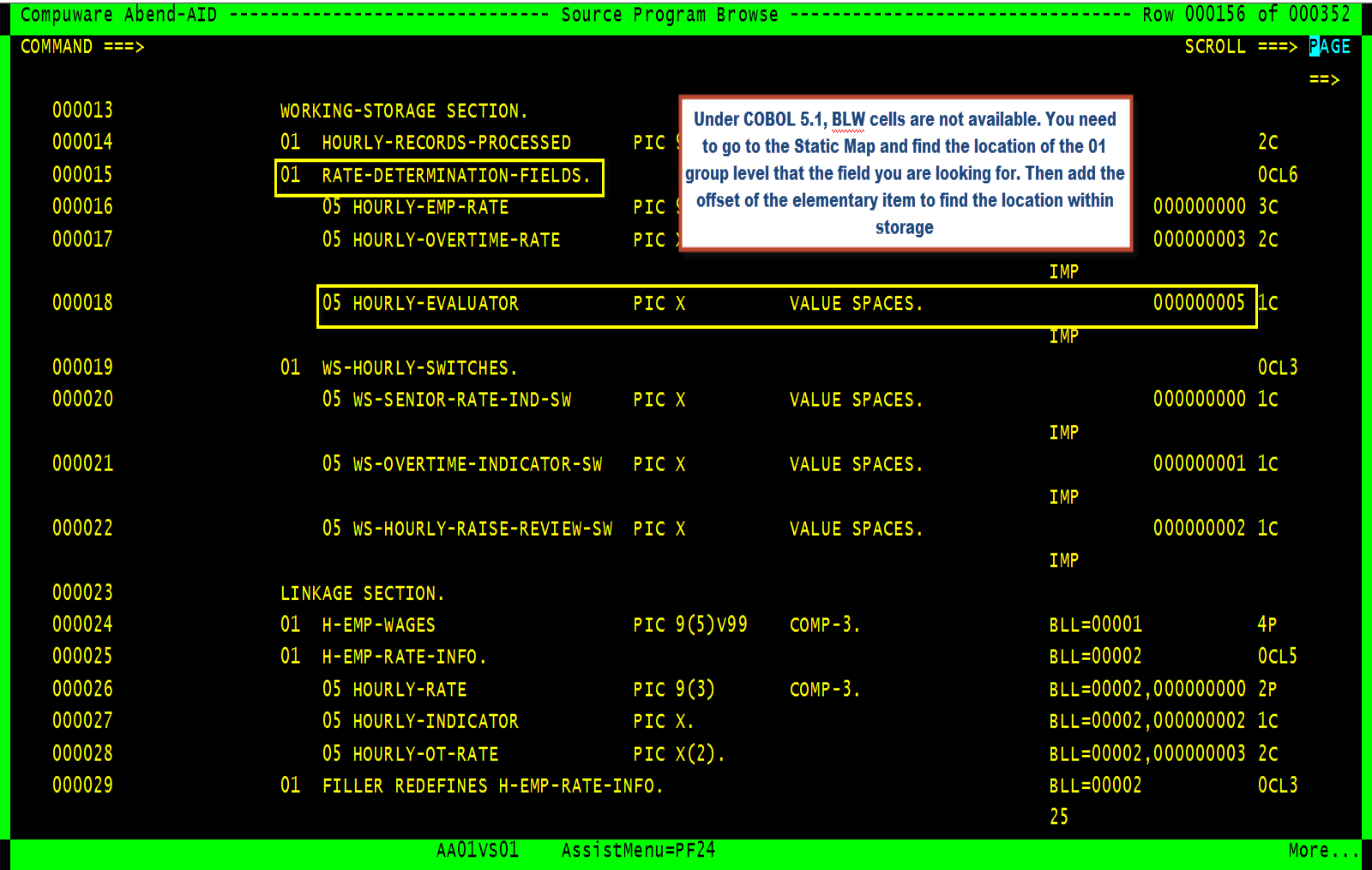
Finding Value of Variable using COBOL 4.2 and Earlier



```
Abend-AID ------------------------------------- Memory Display ---------------------------------------------
COMMAND ===>                                                                                 SCROLL ===> CSR


                                                                                        Clip Prev Next Lock
 Start Addr: 377C1828          Comment: ____________________________

Address     Offset      Word 1     Word 2     Word 3     Word 4      Word 5     Word 6     Word 7     Word 8      Storage
377C1828 +00000000  C3E6C1C1 C4C1E3C5 F0F10000 00000000  F0F2F540 405B0000 E8D5E800 00000000  *CWAADATE01......025  $..YNY.....*
377C1848 +00000020  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C1868 +00000040  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C1888 +00000060  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C18A8 +00000080  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C18C8 +000000A0  00000000 00000000 00000000 00                00 00000000 00000000 00000000  *................................*
377C18E8 +000000C0  00000000 00000000 00000000 00                00 00000000 00000000 00000000  *................................*
377C1908 +000000E0  00000000 00000000 00000000 00                00 00000000 00000000 00000000  *................................*
377C1928 +00000100  00000000 00000000 00000000 00                00 00000000 00000000 00000000  *................................*
377C1948 +00000120  00000000 00000000 00000000 00                00 00000000 00000000 00000000  *................................*
377C1968 +00000140  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C1988 +00000160  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C19A8 +00000180  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C19C8 +000001A0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C19E8 +000001C0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C1A08 +000001E0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C1A28 +00000200  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C1A48 +00000220  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
377C1A68 +00000240  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
```

The value of HOURLY-EVALUATOR is '$'

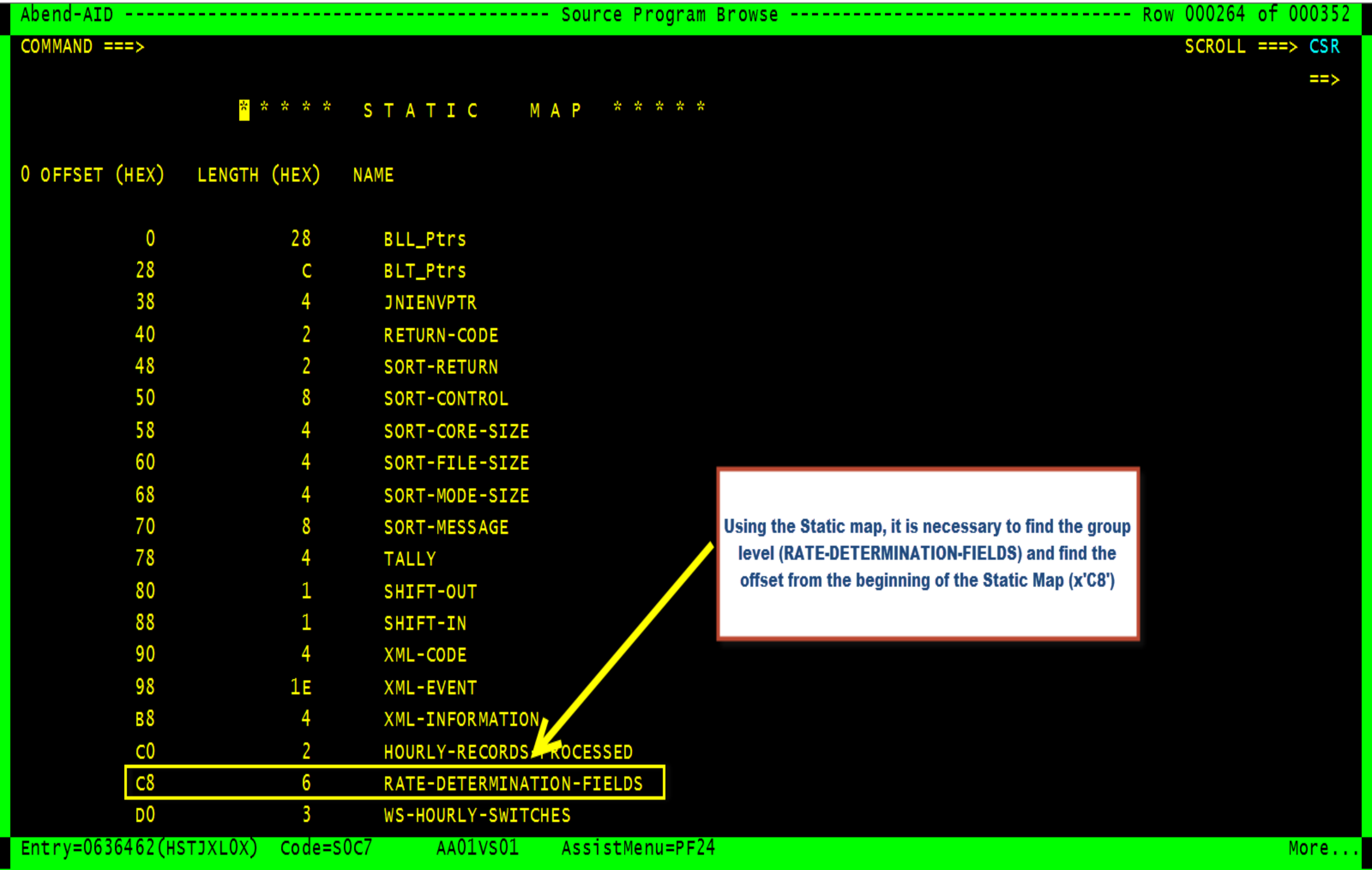
With the 5.2 compiler listing



```
Compuware Abend-AID ------------------------------ Source Program Browse ------------------------------ Row 000156 of 000352
COMMAND ===>                                                                                              SCROLL ===> PAGE
                                                                                                                       ==>
   000013                WORKING-STORAGE SECTION.
   000014                01  HOURLY-RECORDS-PROCESSED     PIC                                                       2C
   000015                01  RATE-DETERMINATION-FIELDS.                                                             0CL6
   000016                    05  HOURLY-EMP-RATE          PIC                                             000000000 3C
   000017                    05  HOURLY-OVERTIME-RATE     PIC                                             000000003 2C
                                                                                                 IMP
   000018                    05  HOURLY-EVALUATOR         PIC X          VALUE SPACES.                    000000005 1C
                                                                                                 IMP
   000019                01  WS-HOURLY-SWITCHES.                                                                    0CL3
   000020                    05  WS-SENIOR-RATE-IND-SW    PIC X          VALUE SPACES.                    000000000 1C
                                                                                                 IMP
   000021                    05  WS-OVERTIME-INDICATOR-SW PIC X          VALUE SPACES.                    000000001 1C
                                                                                                 IMP
   000022                    05  WS-HOURLY-RAISE-REVIEW-SW PIC X         VALUE SPACES.                    000000002 1C
                                                                                                 IMP
   000023                LINKAGE SECTION.
   000024                01  H-EMP-WAGES                  PIC 9(5)V99    COMP-3.                 BLL=00001           4P
   000025                01  H-EMP-RATE-INFO.                                                    BLL=00002           0CL5
   000026                    05  HOURLY-RATE              PIC 9(3)       COMP-3.                 BLL=00002,000000000 2P
   000027                    05  HOURLY-INDICATOR         PIC X.                                 BLL=00002,000000002 1C
   000028                    05  HOURLY-OT-RATE           PIC X(2).                              BLL=00002,000000003 2C
   000029                01  FILLER REDEFINES H-EMP-RATE-INFO.                                   BLL=00002           0CL3
                                                                                                 25
                                        AA01VS01    AssistMenu=PF24                                                More...
```

Under COBOL 5.1, BLW cells are not available. You need to go to the Static Map and find the location of the 01 group level that the field you are looking for. Then add the offset of the elementary item to find the location within storage

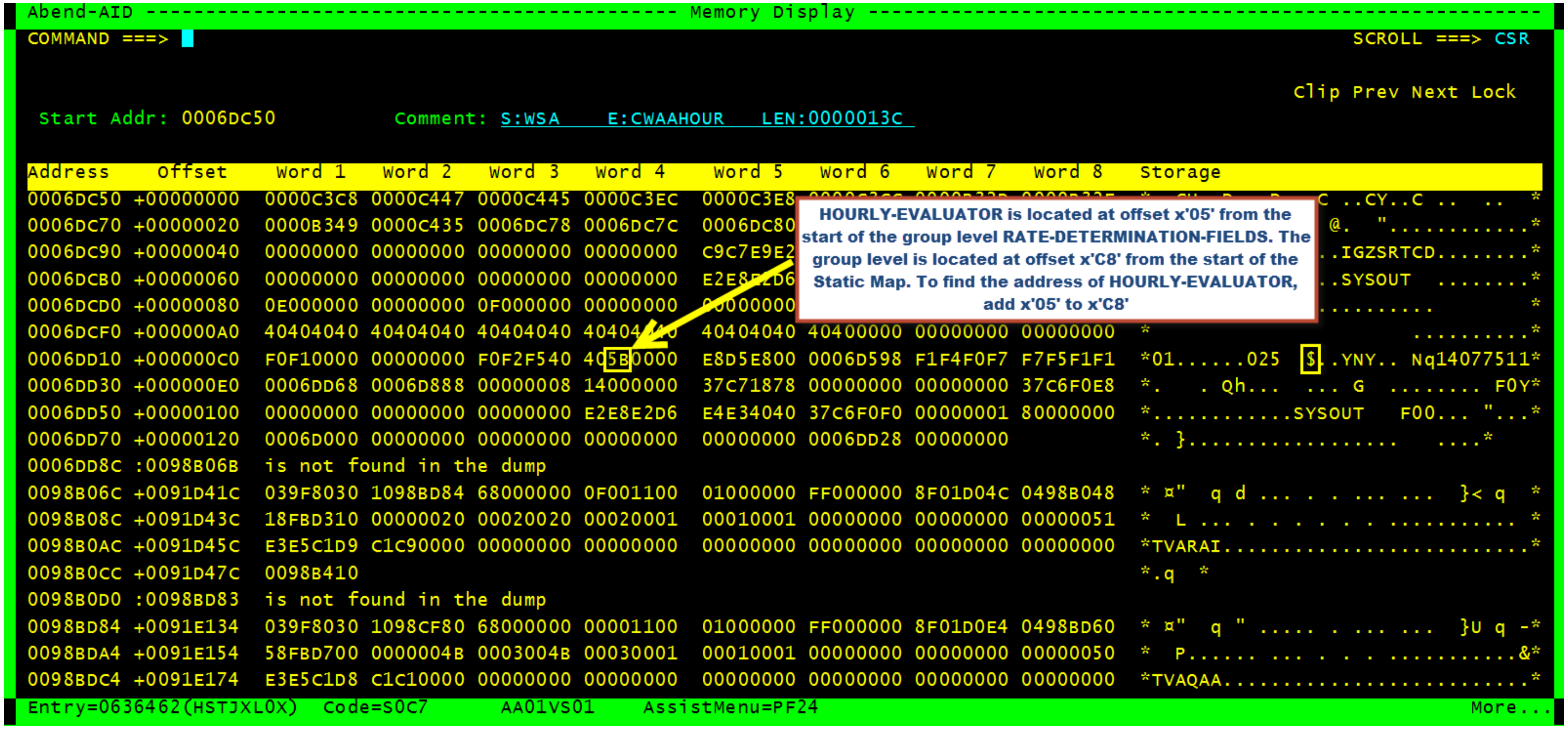
With the 5.2 compiler listing



```
Abend-AID ---------------------------------------- Source Program Browse -------------------------------- Row 000264 of 000352
COMMAND ===>                                                                                                  SCROLL ===> CSR
                                                                                                                          ==>

                      * * * * *   S T A T I C     M A P   * * * * *

0 OFFSET (HEX)    LENGTH (HEX)    NAME

           0              28         BLL_Ptrs
          28               C         BLT_Ptrs
          38               4         JNIENVPTR
          40               2         RETURN-CODE
          48               2         SORT-RETURN
          50               8         SORT-CONTROL
          58               4         SORT-CORE-SIZE
          60               4         SORT-FILE-SIZE
          68               4         SORT-MODE-SIZE
          70               8         SORT-MESSAGE
          78               4         TALLY
          80               1         SHIFT-OUT
          88               1         SHIFT-IN
          90               4         XML-CODE
          98              1E         XML-EVENT
          B8               4         XML-INFORMATION
          C0               2         HOURLY-RECORDS-PROCESSED
          C8               6         RATE-DETERMINATION-FIELDS
          D0               3         WS-HOURLY-SWITCHES
Entry=0636462(HSTJXL0X)  Code=S0C7     AA01VS01     AssistMenu=PF24                                                    More...
```

Using the Static map, it is necessary to find the group level (RATE-DETERMINATION-FIELDS) and find the offset from the beginning of the Static Map (x'C8')

With the 5.2 compiler listing



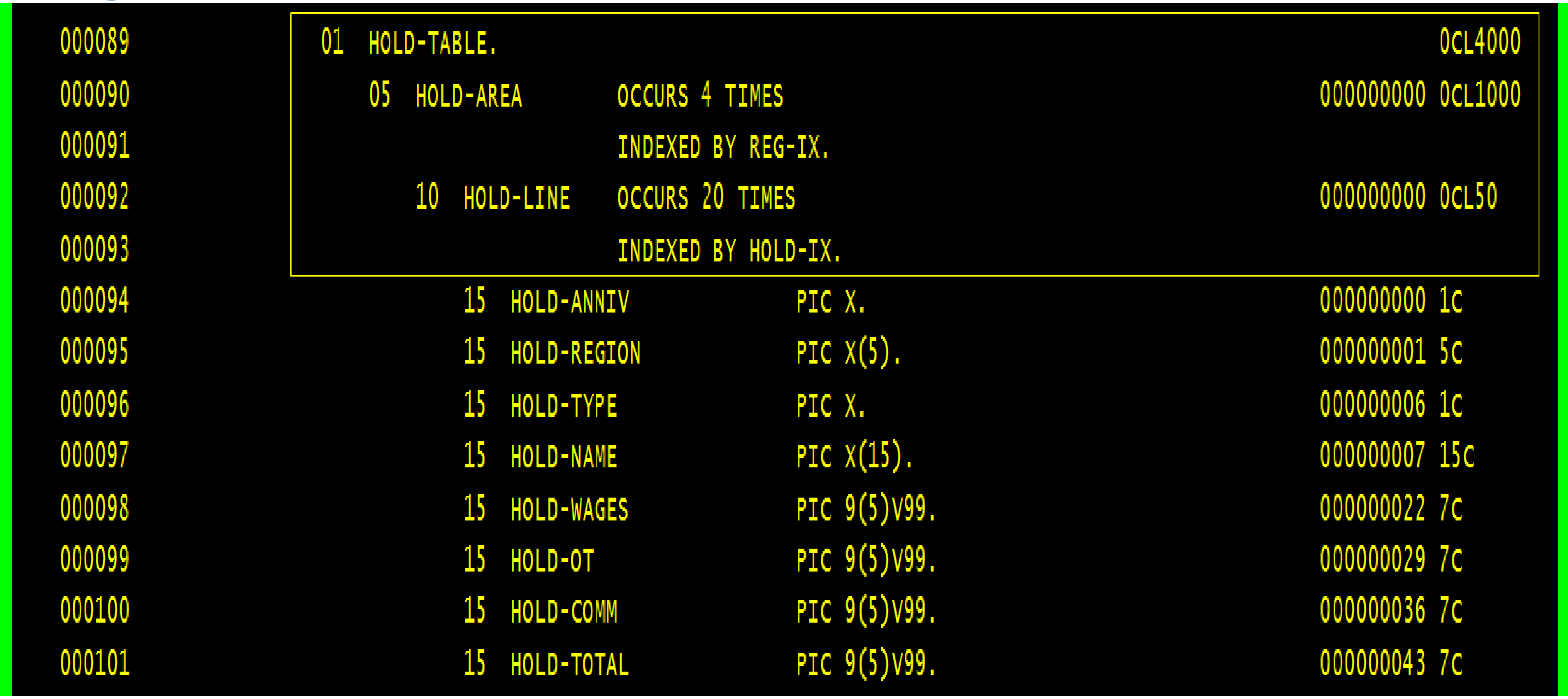
```
Abend-AID ---------------------------------------- Memory Display ---------------------------------------------
COMMAND ===>                                                                                    SCROLL ===> CSR

                                                                                            Clip Prev Next Lock
 Start Addr: 0006DC50          Comment: S:WSA    E:CWAAHOUR   LEN:0000013C

Address    Offset     Word 1   Word 2   Word 3   Word 4    Word 5   Word 6   Word 7   Word 8   Storage
0006DC50 +00000000  0000C3C8 0000C447 0000C445 0000C3EC  0000C3E8                             C ..CY..C ..   ..  *
0006DC70 +00000020  0000B349 0000C435 0006DC78 0006DC7C  0006DC80                              @.  "...........*
0006DC90 +00000040  00000000 00000000 00000000 00000000  C9C7E9E2                            ..IGZSRTCD........*
0006DCB0 +00000060  00000000 00000000 00000000 00000000  E2E8  D6                            ..SYSOUT  ........*
0006DCD0 +00000080  0E000000 00000000 0F000000 00000000   000000                             .........        *
0006DCF0 +000000A0  40404040 40404040 40404040 40404  0  40404040 40400000 00000000 00000000  *                 ..........*
0006DD10 +000000C0  F0F10000 00000000 F0F2F540 405B0000  E8D5E800 0006D598 F1F4F0F7 F7F5F1F1  *01......025 $..YNY.. Nq14077511*
0006DD30 +000000E0  0006DD68 0006D888 00000008 14000000  37C71878 00000000 00000000 37C6F0E8  *.    . Qh...  ... G ........ F0Y*
0006DD50 +00000100  00000000 00000000 00000000 E2E8E2D6  E4E34040 37C6F0F0 00000001 80000000  *...........SYSOUT   F00... "...*
0006DD70 +00000120  0006D000 00000000 00000000 00000000  00000000 0006DD28 00000000           *. }................   ....*
0006DD8C :0098B06B  is not found in the dump
0098B06C +0091D41C  039F8030 1098BD84 68000000 0F001100  01000000 FF000000 8F01D04C 0498B048  * ¤"  q d ... . . ... ...  }< q  *
0098B08C +0091D43C  18FBD310 00000020 00020020 00020001  00010001 00000000 00000000 00000051  *  L ... . . . . . . ..........  *
0098B0AC +0091D45C  E3E5C1D9 C1C90000 00000000 00000000  00000000 00000000 00000000 00000000  *TVARAI.......................*
0098B0CC +0091D47C  0098B410                                                                  *.q  *
0098B0D0 :0098BD83  is not found in the dump
0098BD84 +0091E134  039F8030 1098CF80 68000000 00001100  01000000 FF000000 8F01D0E4 0498BD60  * ¤"  q " ..... . ... ...  }U q -*
0098BDA4 +0091E154  58FBD700 0000004B 0003004B 00030001  00010001 00000000 00000000 00000050  *  P...... ... . . . ..........&*
0098BDC4 +0091E174  E3E5C1D8 C1C10000 00000000 00000000  00000000 00000000 00000000 00000000  *TVAQAA.......................*
Entry=0636462(HSTJXL0X)  Code=S0C7      AA01VS01     AssistMenu=PF24                                      More...
```

HOURLY-EVALUATOR is located at offset x'05' from the start of the group level RATE-DETERMINATION-FIELDS. The group level is located at offset x'C8' from the start of the Static Map. To find the address of HOURLY-EVALUATOR, add x'05' to x'C8'

New IBM Compiler output

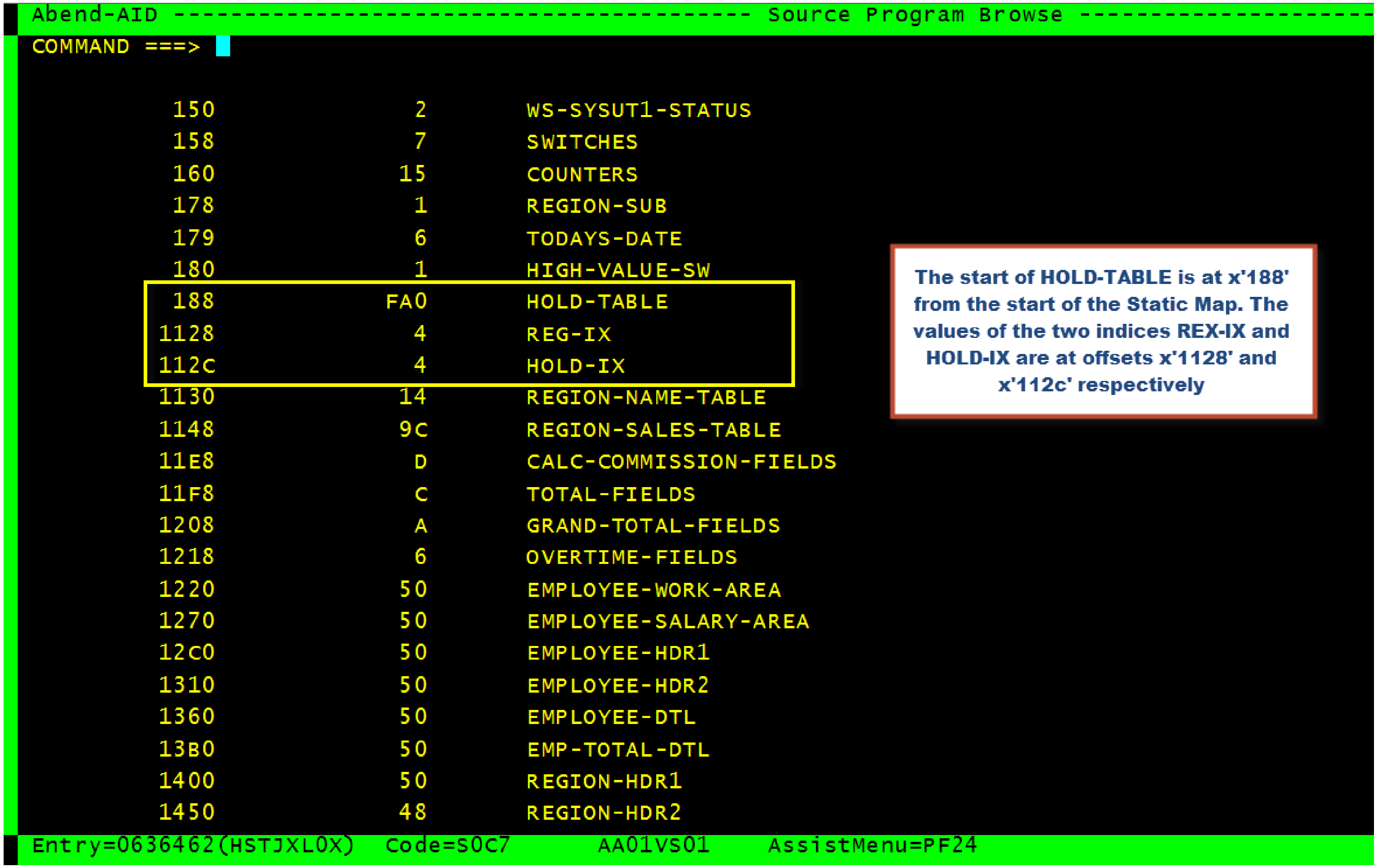
- Finding the value of the index has become more problematic under 5.2. The Indices and the offset are listed in the static map. However, when you go to the storage, the value is an offset.
  - You have to calculate the value of the offset against the length of the array level plus 1. The initial index value location was at offset 0 of the array.

Finding the Value of the Indices under COBOL 5.2



```
000089        01  HOLD-TABLE.                                                  0CL4000
000090            05  HOLD-AREA       OCCURS 4 TIMES                 000000000 0CL1000
000091                                INDEXED BY REG-IX.
000092                10  HOLD-LINE   OCCURS 20 TIMES                000000000 0CL50
000093                                INDEXED BY HOLD-IX.
000094                    15  HOLD-ANNIV          PIC X.             000000000 1C
000095                    15  HOLD-REGION         PIC X(5).          000000001 5C
000096                    15  HOLD-TYPE           PIC X.             000000006 1C
000097                    15  HOLD-NAME           PIC X(15).         000000007 15C
000098                    15  HOLD-WAGES          PIC 9(5)V99.       000000022 7C
000099                    15  HOLD-OT             PIC 9(5)V99.       000000029 7C
000100                    15  HOLD-COMM           PIC 9(5)V99.       000000036 7C
000101                    15  HOLD-TOTAL          PIC 9(5)V99.       000000043 7C
```

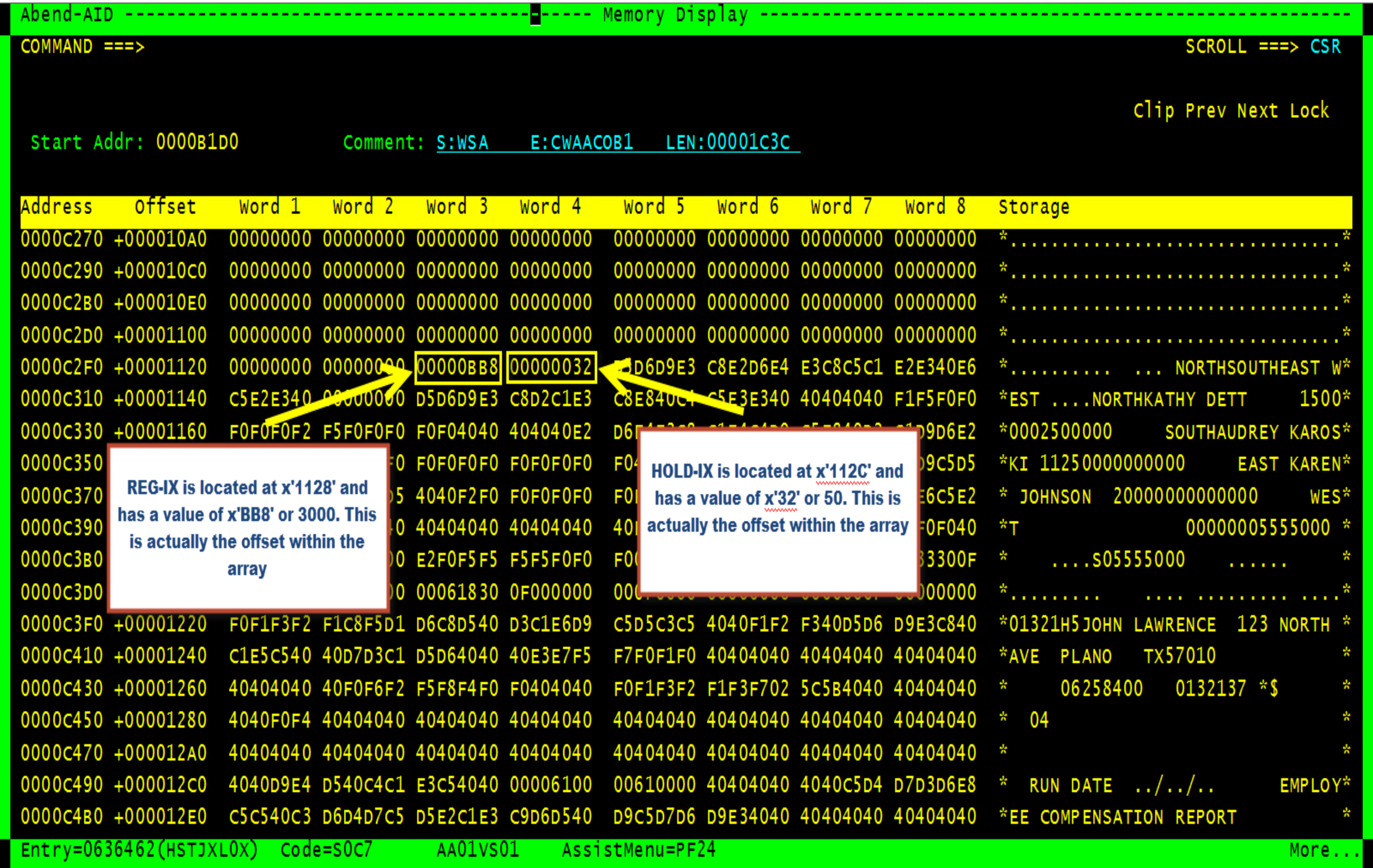
Finding the Value of the Indices under COBOL 5.2



```
Abend-AID ---------------------------------------- Source Program Browse --------------------
COMMAND ===>

         150              2       WS-SYSUT1-STATUS
         158              7       SWITCHES
         160             15       COUNTERS
         178              1       REGION-SUB
         179              6       TODAYS-DATE
         180              1       HIGH-VALUE-SW
         188            FA0       HOLD-TABLE
        1128              4       REG-IX
        112C              4       HOLD-IX
        1130             14       REGION-NAME-TABLE
        1148             9C       REGION-SALES-TABLE
        11E8              D       CALC-COMMISSION-FIELDS
        11F8              C       TOTAL-FIELDS
        1208              A       GRAND-TOTAL-FIELDS
        1218              6       OVERTIME-FIELDS
        1220             50       EMPLOYEE-WORK-AREA
        1270             50       EMPLOYEE-SALARY-AREA
        12C0             50       EMPLOYEE-HDR1
        1310             50       EMPLOYEE-HDR2
        1360             50       EMPLOYEE-DTL
        13B0             50       EMP-TOTAL-DTL
        1400             50       REGION-HDR1
        1450             48       REGION-HDR2
Entry=0636462(HSTJXLOX)  Code=S0C7     AA01VS01     AssistMenu=PF24
```

The start of HOLD-TABLE is at x'188' from the start of the Static Map. The values of the two indices REX-IX and HOLD-IX are at offsets x'1128' and x'112c' respectively

Finding the Value of the Indices under COBOL 5.2



```
Abend-AID -------------------------------------------- Memory Display ----------------------------------------------------------------
COMMAND ===>                                                                                              SCROLL ===> CSR

                                                                                                      Clip Prev Next Lock
 Start Addr: 0000B1D0          Comment: S:WSA    E:CWAACOB1   LEN:00001C3C

Address    Offset     Word 1   Word 2   Word 3   Word 4    Word 5   Word 6   Word 7   Word 8   Storage
0000C270 +000010A0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
0000C290 +000010C0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
0000C2B0 +000010E0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
0000C2D0 +00001100  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
0000C2F0 +00001120  00000000 00000000 00000BB8 00000032  D5D6D9E3 C8E2D6E4 E3C8C5C1 E2E340E6  *..........  ... NORTHSOUTHEAST W*
0000C310 +00001140  C5E2E340 00000000 D5D6D9E3 C8D2C1E3  C8E840C4 C5E3E340 40404040 F1F5F0F0  *EST ....NORTHKATHY DETT     1500*
0000C330 +00001160  F0F0F0F2 F5F0F0F0 F0F04040 404040E2  D6                           D9D6E2  *0002500000     SOUTHAUDREY KAROS*
0000C350                              0 F0F0F0F0 F0F0F0F0  F0                           D9C5D5  *KI 1125000000000         EAST KAREN*
0000C370                              5 4040F2F0 F0F0F0F0  F0                           E6C5E2  * JOHNSON   20000000000000      WES*
0000C390                              0 40404040 40404040  40                           F0F040  *T                    00000005555000 *
0000C3B0                              0 E2F0F5F5 F5F5F0F0  F0                           3300F  *      ....S05555000      ......    *
0000C3D0                              0 00061830 0F000000  00                           00000  *.........      .... ......... ....*
0000C3F0 +00001220  F0F1F3F2 F1C8F5D1 D6C8D540 D3C1E6D9  C5D5C3C5 4040F1F2 F340D5D6 D9E3C840  *01321H5JOHN LAWRENCE   123 NORTH *
0000C410 +00001240  C1E5C540 40D7D3C1 D5D64040 40E3E7F5  F7F0F1F0 40404040 40404040 40404040  *AVE  PLANO    TX57010            *
0000C430 +00001260  40404040 40F0F6F2 F5F8F4F0 F0404040  F0F1F3F2 F1F3F702 5C5B4040 40404040  *      06258400    0132137 *$     *
0000C450 +00001280  4040F0F4 40404040 40404040 40404040  40404040 40404040 40404040 40404040  *  04                            *
0000C470 +000012A0  40404040 40404040 40404040 40404040  40404040 40404040 40404040 40404040  *                                *
0000C490 +000012C0  4040D9E4 D540C4C1 E3C54040 00006100  00610000 40404040 4040C5D4 D7D3D6E8  *  RUN DATE  ../../..      EMPLOY*
0000C4B0 +000012E0  C5C540C3 D6D4D7C5 D5E2C1E3 C9D6D540  D9C5D7D6 D9E34040 40404040 40404040  *EE COMPENSATION REPORT           *
Entry=0636462(HSTJXL0X)  Code=S0C7       AA01VS01     AssistMenu=PF24                                                      More...
```

REG-IX is located at x'1128' and has a value of x'BB8' or 3000. This is actually the offset within the array

HOLD-IX is located at x'112C' and has a value of x'32' or 50. This is actually the offset within the array

Finding the Value of the Indices under COBOL 5.2

- From the compiled listing, the HOLD-TABLE array is 4000 bytes long. Each occurrence of HOLD-AREA is 1000 bytes and HOLD-LINE is 50 bytes long.
- Since the value of REG-IX is 3000, and represents the offset within the array, the value of the index can be calculated by dividing the offset by the length of the array (3000/1000 = 3) and then adding 1. This is necessary because the array actually starts at offset 0. So the value of the index is 4
- HOLD-IX's value is 50. The length of HOLD-LINE is 50, so the value of HOLD-IX is 50/50 + 1 or 2

Questions?

The Mainframe Software Partner
For The Next 50 Years