

Comparing the Assembler source to the Cobol target program

Subject, issue	Assembler program	Cobol program
source code lines	361	199 source code lines. Reduction by 45%!
branch instructions	25	No (0) GO TO instructions!
VSAM macros	OPEN, CLOSE, GET, ACB, RPL, SHOWVCB (2)	Converted to one VSAM file declaration
Self-modifying code		replaced by ordinary switch logic
z/OS macros	SAVE, RETURN, WTO (2), CALL (2)	No z/OS macro needed to be emulated
Dead code sequences	Significant portions turn out to be never referenced	Detected and removed
Redundant operations	... like clearing a register and then loading it with a value	Detected and removed
Non-symbolic coding	e.g. 36(R2) instead of working with symbolic names	Replaced by artificial symbols

The assembler program is a typical example for „lethal maintenance“: the work of the past 30 years was done so awful that the program became almost unreadable.

The program is part of a productive customer application package. To keep privacy, it has been manually encrypted.

The New Cobol Program (Result of the Transformation)

```

*-----
* Program    SAMPLE02
* Converted  2016-09-01 10:25:13
* Using      Njema V4.2
* Revised    2016-09-02 esc
* Encrypted  2016-09-16 esc
*-----

* *****
* MODULE     NAME SAMPLE02
* FUNKTION   xxx xxx xxx xxx xxx xxx
*            READ VSAM FILE (MYFILE) LOCATE MODUS
*            RECORD          = VSAM-SCH          = 20          BYTE
*            1. 4 BYTE FOR xxxxxx
*            1.  O/C  OPEN/CLOSE xxxxxx
*            2.  1-6  S/U/G, ZERO,BM
*            3-4. VSAM-ERROR RETURN-CODE
*            2. 4 BYTE FOR xxxxxx
*            3. 8 BYTE FOR YYYYYY
*            4. 4 BYTE
*            1.  U = OVERFLOW MODE
*            2-4. EMPTY
*            5. 4 BYTE MYFILE-ADDR
*            6. 4 BYTE xxFILE-ADDR
*            = PL-33-DATA    = 53          BYTE
*            = PL-34-REC-CNT = 2          COMP  BYTE
*            = PL-34-TAB    = 2004       BYTE
* AUTHOR     author

```

```
* DATE          04.01.1993
* *****
*
```

```
ID DIVISION.
PROGRAM-ID. SAMPLE02.
```

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
DECIMAL-POINT IS COMMA.
```

```
INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT MYFILE
ASSIGN TO AS-MYFILE
ORGANIZATION IS INDEXED
ACCESS MODE IS RANDOM
RECORD KEY IS VSAMKEY
FILE STATUS IS MYFILE-FILE-STATUS.
```

Derived from the ACB and RPL macros

```
-----
DATA DIVISION.
-----
```

```
FILE SECTION.
FD MYFILE.
01 MYFILE-RECORD PIC X(10000).
01 FILLER REDEFINES MYFILE-RECORD.
02 FILLER PIC X(54).
02 MYFILE-RECORD-55B2 PIC S9(4) COMP.
02 FILLER PIC X(9944).
```

Record size is yet unknown, hence set to a fictional value of 10,000

Artificial Cobol names are created to avoid reference modifiers.

```
-----
WORKING-STORAGE SECTION.
-----
```

```
*
* DATA-AREA
01 RECAA PIC S9(8) COMP VALUE ZERO.
01 RECAA-PTR REDEFINES RECAA POINTER.
*
* HALFWORD-NUMBERS
* FULLWORD-NUMBERS
77 H1754 PIC S9(4) VALUE 1754.
77 H2004 PIC S9(4) VALUE 2004.
77 F56 PIC S9(4) VALUE 56.
```

These comments are a leftover after the fields below them were removed, because they are unused. Such comments can only be removed manually.

```
-----
* Definitions added by Njema
-----
```

```
77 MYFILE-FILE-STATUS PIC XX.
01 VSAMKEY PIC X(9).
01 FILLER REDEFINES VSAMKEY.
02 VSAMKEY-1C1 PIC X.
02 VSAMKEY-2C8 PIC X(8).
```

Status field automatically generated

```
-----
LINKAGE SECTION.
-----
```

```
COPY VSAM-SCH.
COPY VPL33A.
```

Two new COPY books were established for the whole application to allow for symbolic reference of fields.

```
-----
* Definitions added by Njema
-----
```

```
01 MVCL01-SOURCE PIC X(10000).
01 MVCL01-TARGET PIC X(10000).
```

These two fields were inserted by the assembler MVCL replacement rule.

```

*-----
PROCEDURE DIVISION USING VSAM-SCH, PL-33-34.
*-----

```

```

EVALUATE MYFILE-OPEN
  WHEN 'O'
    PERFORM GETIT
  WHEN 'C'
    PERFORM XXX-CLOSE
  WHEN 'E'
    PERFORM XXX-CLOSE
  WHEN SPACE
    PERFORM XXX-OPEN
END-EVALUATE
PERFORM ENDE.

```

The re-engineering process, applied after the conversion to Cobol, carves out the main control logic clearly.

```

*-----
* Subroutines created by Njema
*-----

```

```

*-----
ENDE SECTION.
*-----

```

```

***** REG. PROGRAM END
SET RETURN-CODE TO 0
GOBACK.

```

```

*-----
XXX-CLOSE SECTION.
*-----

```

If the Assembler program contains symbols, which are Reserved Words in Cobol, they are prefixed with an arbitrary string, here „XXX-“

```

***** CLOSE MYFILE
IF MYFILE-OPEN NOT = 'E'
  CLOSE MYFILE
  MOVE 'C' TO MYFILE-O-C
  MOVE 'C' TO MYFILE-OPEN
END-IF
IF VSAM22-OPEN NOT = 'E'
  CALL 'SUBPROG2' USING VSAM-SCH
END-IF
IF VSAM24-OPEN NOT = 'E'
  CALL 'SUBPROG1' USING VSAM-SCH
END-IF
PERFORM ENDE.

```

```

*-----
GET2 SECTION.
*-----

```

```

MOVE '3' TO MYFILE-O-C
MOVE VSAM22-PL35-KEY TO VSAMKEY-1C1
MOVE VSAM22-PL36-PRUEF-ID TO VSAMKEY-2C8
READ MYFILE
  INVALID KEY
    DISPLAY 'Error reading MYFILE, Status = '
      MYFILE-FILE-STATUS UPON CONSOLE
    SET RETURN-CODE TO 8
  NOT INVALID KEY
    MOVE ZERO TO MYFILE-O-C
    MOVE MYFILE-RECORD TO PL-33
    MOVE MYFILE-RECORD-55B2 TO PL34-SATZ-ANZ
    SET RECAA-PTR TO ADDRESS OF MYFILE-RECORD
    SET RECAA-PTR UP BY F56
END-READ

```

```

***** ERROR HANDLING
GOBACK.

```

```

*-----
XXX-OPEN SECTION.
*-----
***** OPEN MYFILE KEY LOC-MODE
      OPEN  INPUT MYFILE
      MOVE  'O' TO MYFILE-O-C
      MOVE  'O' TO MYFILE-OPEN
      MOVE  ZERO TO PL34-SATZ-ANZ
      PERFORM GETIT1.

*-----
      GETIT SECTION.
*-----
***** READ MYFILE WHEN PL34-REC-COUNT = ZERO
      IF PL34-SATZ-ANZ NOT = ZERO
*MVCL
      SET ADDRESS OF MVCL01-TARGET TO ADDRESS OF PL-34-TAB
      SET ADDRESS OF MVCL01-SOURCE TO RECAA-PTR
      MOVE MVCL01-SOURCE (1:H2004) TO MVCL01-TARGET
      SET RECAA-PTR UP BY H2004
      SET MYFILE-ADR-PTR UP BY H1754
*MVCL
      SUBTRACT 1 FROM PL34-SATZ-ANZ
      PERFORM ENDE
      END-IF.

*-----
      GETIT1 SECTION.
*-----
      MOVE ZERO TO VSAM22-STATUS
      MOVE ZERO TO VSAM24-STATUS
      IF VSAM22-PL35-KEY = 'S'
          CALL 'SUBPROG1' USING VSAM-SCH
          IF VSAM24-STATUS > ZERO
              PERFORM ENDE
          END-IF
          PERFORM GET2
      END-IF
      CALL 'SUBPROG2' USING VSAM-SCH
      IF VSAM22-STATUS > ZERO
          PERFORM ENDE
      END-IF
      PERFORM GET2.

```

This is the only remaining situation, where pointers are still needed. The logic of the MOVE LONG instruction requires it.

The original Assembler Program

```

* Encrypted 2016-09-16 esc
* ***** 00001000
* MODULE          NAME SAMPLE02          * 00002001
* FUNKTION        xxx xxx xxx xxx xxx xxx * 00003000
*                READ VSAM FILE (MYFILE) LOCATE MODUS * 00003100
*                RECORD          = VSAM-SCH          = 20          BYTE * 00003200
*                1. 4 BYTE FOR xxxxxxxx          * 00003301
*                1.  O/C  OPEN/CLOSE xxxxxxxx          * 00003400
*                2.  1-6  S/U/G, ZERO,BM          * 00003500
*                3-4. VSAM-ERROR RETURN-CODE          * 00003600
*                2. 4 BYTE FOR xxxxxxxx          * 00003701
*                3. 8 BYTE FOR yyyyyy          * 00003801
*                4. 4 BYTE          * 00003900
*                1.  U = OVERFLOW MODE          * 00004000
*                2-4. EMPTY          * 00004100
*                5. 4 BYTE MYFILE-ADDR          * 00004200
*                6. 4 BYTE xxFILE-ADDR          * 00004300
*                = PL-33-DATA          = 53          BYTE * 00004400
*                = PL-34-REC-CNT      = 2          COMP BYTE * 00004500
*                = PL-34-TAB          = 2004          BYTE * 00004600
* AUTHOR          author          * 00004700
* DATE            04.01.1993          * 00004800
* ***** 00004900
*                FRAME FOR ASSEMBLER-SUBROUTINES WITH PARAMETERS * 00005000
* ***** 00006000
                SPACE          00007000
R0          EQU  0          00008000
R1          EQU  1          00009000
R2          EQU  2          00010000
R3          EQU  3          00020000
R4          EQU  4          00030000
R5          EQU  5          00040000
R6          EQU  6          00050000
R7          EQU  7          00060000
R8          EQU  8          00070000
R9          EQU  9          00080000
R10         EQU 10          00090000
R11         EQU 11          00100000
R12         EQU 12          00110000
R13         EQU 13          00120000
R14         EQU 14          00130000
R15         EQU 15          00140000
*                00150000
SAMPLE02 CSECT          00160001
SAMPLE02 RMODE ANY          00170001
SAMPLE02 AMODE 31          00180001
                DS  0H          00190000
*                SAVE  (14,12)          00200000
                STM  R14,R12,12(R13)          00201000
                USING SAMPLE02,11,10          00202001
                LR   R11,R15          00203000
                LA   R10,4095(R11)          CONNECT SAVE-AREAS          00204000
                LA   R10,1(R10)          CONNECT SAVE-AREAS          00205000
                ST   R13,SAVEAREA+4          00206000
                LR   R12,R13          00207000
                LA   R13,SAVEAREA          00208000
                ST   R13,8(R12)          00209000
*                00210000
*                00220000
*                00230000
ANFANG DS  0H          00240000
*                00250000
                LM   R2,R5,0(R1)          SWAP PARAMETER-ADDRESSES          00260000
                STM  R2,R5,VSSCH          00270000

```

```

      CLI  36(R2),C'O'          MYFILE = OPEN ?          00280002
      BE  GETIT                 YES                          00290000
      CLI  36(R2),C'C'          CLOSE MYFILE/22/24          00300002
      BE  CLOSE                 YES                          00301000
      CLI  36(R2),C'E'          CLOSE SYS022/24            00301102
      BE  CLOSE                 YES                          00301202
      CLI  36(R2),C' '          OPEN  MYFILE/22/24          00301302
      BE  OPEN                  YES                          00301402
      B   ENDE                  YES                          00303302
*
***** OPEN MYFILE KEY LOC-MODE 00306000
*
OPEN   DS    0H                00307000
*
*                                00307100
*                                00307200
*                                00307300
*                                00307400
      OPEN  (MYFILE),MODE=31    00307608
      LTR  R15,R15              00307800
*      BP  ERRORX               00308000
      BP  ERRORX10             00308100
      MVI  0(R2),C'O'          MYFILE OPEN OK             00308200
      MVI  36(R2),C'O'          MYFILE OPEN OK             00309000
      SR   R7,R7                00309100
      STH  R7,0(0,R4)          PL34-REC-COUNT = ZERO       00309200
*
      B    GETIT1              00309300
*                                00309400
*                                00309500
***** CLOSE MYFILE 00309600
*
CLOSE  DS    0H                00309700
*                                00309800
*                                00309900
*      WTO ('CLOSE MYFILE',D),(' ',DE),ROUTCDE=(11),DESC=(7) 00310000
      CLI  36(R2),C'E'          CLOSE MYFILE ?             00310102
      BE  CLO22                 YES                          00310202
      CLOSE (MYFILE),MODE=31    00311008
*      LTR  R15,R15              00312000
*      BP  ERRORX               00312100
      MVI  0(R2),C'C'          MYFILE CLOSE OK             00312200
      MVI  36(R2),C'C'          MYFILE CLOSE OK             00312300
*
*                                00312400
*                                00312500
      WTO ('CALLC 10=>20',D),(' ',DE),ROUTCDE=(11),DESC=(7) 00312602
CLO22  DS    0H                00312702
      CLI  37(R2),C'E'          CLOSE SYS022 ?             00312802
      BE  CLO23                 YES                          00312901
      CALL SUBPROG2,((R2))      00313000
*      WTO ('CALLC 10=>30',D),(' ',DE),ROUTCDE=(11),DESC=(7) 00313102
CLO23  DS    0H                00313202
      CLI  38(R2),C'E'          CLOSE SYS024 ?             00313302
      BE  ENDE                  YES                          00313401
      CALL SUBPROG1,((R2))      00313600
      B    ENDE                00313700
*
***** READ MYFILE WHEN PL34-REC-COUNT = ZERO 00313800
*
GETIT  DS    0H                00313900
*
      SR   R8,R8                00314000
      CH  R8,0(0,R4)           PL34-REC-COUNT = ZERO ?     00314100
      BNE GETIT2               NEXT PL-34-TAB              00314200
*
      MVC  6(2,R2),C00          00314300
      MVC  22(2,R2),C00         00314400
*
      CLI  8(R2),C'S'          S-CODE ?                     00314500
      BE  GET1                  00314600
*      WTO ('CALLG 10=>20',D),(' ',DE),ROUTCDE=(11),DESC=(7) 00314700
      CALL SUBPROG2,((R2))      V-CODE                     00315000
*
*                                00315100
*                                00315200
*                                00315300
*                                00315400
*                                00315500
*                                00315601
*                                00315700

```

```

        CLC      6(2,R2),C00          16=REC NOT THERE, 99=COUNT = ZERO    00315800
        BH      ENDE
        B       GET2
*
*
GET1    DS      0H
*
*
*      WTO ('CALLG 10=>30',D),('*',DE),ROUTCDE=(11),DESC=(7)
        CALL   SUBPROG1,((R2))      S-CODE
*
*
        CLC      22(2,R2),C00        16=REC NOT THERE, 99=COUNT = ZERO    00316700
        BH      ENDE
*
*
GET2    DS      0H
*
*
        MVI     0(R2),C'3'          VSSCH = 3 (GET-MODUS)
*
*
        MVC     VSAMKEY(1),8(R2)
        MVC     VSAMKEY+1(8),12(R2)
        GET     RPL=MYRPL           READ MYFILE
        LTR     R15,R15
*
        BP     ERRORX
        BP     ERRORX10
*
*
        MVI     0(R2),C'O'          MYFILE GET OK (OPEN MODUS)
        L       R6,AR2             START REC-ADDR. IN MYFILE-REC
        MVC     0(54,R3),0(R6)
        MVC     0(2,R4),54(R6)
        L       R9,F56             R9 = 56
        AR      R6,R9
        ST      R6,RECAA
*
*
GETIT2  DS      0H
*MVCL
        LR      R8,R5
        L       R6,RECAA
        XR      R7,R7
        XR      R9,R9
        LH      R7,H2004
        LH      R9,H2004
*
        ICM     R7,B'1000',=X'40'
*
        LR      R9,R7
        MVCL    R8,R6
        L       R6,RECAA
        AH      R6,H2004
        ST      R6,RECAA
        AH      R5,H1754
        ST      R5,ST5             R5 = 8
        MVC     24(4,R2),ST5
*MVCL
*
*      L       R6,RECAA
*
*      MVC     0(254,R5),0(R6)     1. 254 BYTE TO 1. PL-34-TAB
*
*      L       R9,F254             R9 = 254
*
*      AR      R5,R9
*
*      AR      R6,R9
*
*      L       R9,F250             R9 = 250
*
*      LA      R7,GETIT3          R7 = ADDR. GETIT2
*
*      L       R8,F7              R8 = 7
*
*
GETIT3  DS      0H
*
*
*      MVC     0(250,R5),0(R6)
*
*      ST      R5,ST5             R5 = 8
*
*      MVC     24(4,R2),ST5
*
*      AR      R5,R9
*
*      AR      R6,R9
*
*      BCTR   R8,R7              BRANCH TO GETIT3
*
*      ST      R6,RECAA
*
*
*      L       R5,PL34T           R5 = START TAB AREA

```

```

LH      R8,0(0,R4)                00324500
SH      R8,H1                      00324600
STH     R8,0(0,R4)                00324700
B       ENDE                       00324800
*                                     00324900
***** ERROR HANDLING             00325000
*                                     00325100
ERRORX  DS      0H                  ERROR HANDLING          00325200
*                                     00325300
MVC     LAUS+27(18),=C'10 R15/ERROR/MESS=' 00325400
CVD     R15,DWORT                   00325500
UNPK    HILF(16),DWORT(8)           00325600
MVZ     HILF+15(1),FF               ZONES                   00325700
MVC     LAUS+47(4),HILF+12          00325800
*                                     00325900
CLI     0(R2),C'3'                  MYFILE GET ERROR      00326000
BE      ERRORX3                     YES                    00327000
*                                     00328000
SHOWCB  ACB=MYFILE,AREA=AR1,LENGTH=12,FIELDS=(ERROR,MAREA,MLEN) 00329000
*                                     00330000
L       R9,AR1                      ADDR. OF AR1          00340000
CVD     R9,DWORT                    00350000
UNPK    HILF(16),DWORT(8)           00360000
MVZ     HILF+15(1),FF               00370000
MVC     LAUS+52(4),HILF+12          ERROR TO LISTING     00380000
*                                     00390000
L       R9,AR1+4                    MESSAGE ADDRESS       00400000
L       R10,AR1+8                   MESSAGE LENGTH        00410000
BCTR    R10,0                       MESSAGE LENGTH - 1    00420000
STC     R10,MVC1+1                  MESSAGE LENGTH TO MVC 00430000
MVC1    MVC LAUS+57(0),0(R9)        00440000
*                                     00450000
CLI     AR1+7,X'01'                 MYFILE OPEN ERROR?   00460000
CLI     AR1+7,X'02'                 MYFILE CLOSE ERROR? 00470000
BE      ERRORX2                     YES                    00480000
*                                     00490000
ERRORX1 DS      0H                  MYFILE OPEN ERROR    00500000
*                                     00510000
MVC     LAUS+1(24),=C'OPEN ERROR IN MYFILE ' 00520001
B       ERRORX9                     YES                    00530000
*                                     00540000
ERRORX2 DS      0H                  MYFILE CLOSE ERROR   00540100
*                                     00540200
MVC     LAUS+1(24),=C'CLOSE ERROR IN MYFILE ' 00540301
B       ERRORX9                     YES                    00540400
*                                     00540500
ERRORX3 DS      0H                  MYFILE GET ERROR     00540600
*                                     00540700
MVC     2(2,R2),C16                 REC-NOT-FOUND => VSAM-SCH 00540800
MVC     LAUS+1(24),=C'GET ERROR IN MYFILE ' 00540901
MVC     LAUS+27(15),=C'R15/FDBK/FTNCD=' 00541000
*                                     00541100
SHOWCB  RPL=RPL21,AREA=AR1,LENGTH=8,FIELDS=(FDBK,FTNCD) 00541200
*                                     00541300
L       R9,AR1                      ADDR. OF FDBK        00541400
CVD     R9,DWORT                    00541500
UNPK    HILF(16),DWORT(8)           00541600
MVZ     HILF+15(1),FF               00541700
MVC     LAUS+52(4),HILF+12          00541800
*                                     00541900
L       R9,AR1+4                    ADDR. OF FTNCD       00542000
CVD     R9,DWORT                    00542100
UNPK    HILF(16),DWORT(8)           00542200
MVZ     HILF+15(1),FF               00542300
MVC     LAUS+57(4),HILF+12          00542400
B       ERRORX9                     YES                    00542500
*                                     00542600
ERRORX9 DS      0H                  EXIT SUBROUTINE IF ERROR 00542700
*                                     00542800

```



```

MVC      WTO2+8(100),LAUS                00542900
DS       0H                              00543000
WTO2     WTO  ('
                                     -00544000
                                     ',D),-00544100
                                     ( '*' ,DE),ROUTCDE=(11),DESC=(7)
                                     00544200
*                                               00544300
CLI      0(R2),C'3'                      MYFILE GET  ERROR  00544400
BNE      ERRORX10                        00544500
MVC      LAUS+1(8),=C'VSAMKEY='          00544600
MVC      LAUS+9(9),VSAMKEY              VSAMKEY  00544700
MVC      LAUS+18(16),=C'VSAM-PL35-KEY='  00544800
MVC      LAUS+34(4),8(R2)                00544900
MVC      WTO3+8(100),LAUS                00545000
DS       0H                              00545100
WTO3     WTO  ('
                                     -00545200
                                     ',D),-00545300
                                     ( '*' ,DE),ROUTCDE=(11),DESC=(7)
                                     00545400
*                                               00545500
ERRORX10 DS  0H                          EXIT SUBROUTINE IF ERROR 00545600
L        R13,4(R13)                       00545700
RETURN   (14,12),RC=15                    00545800
*                                               00545900
***** REG. PROGRAM END                   00546000
*                                               00546100
ENDE     DS  0H                            00546200
*                                               00546300
L        R13,4(R13)                       00546400
*                                               00546500
RETURN   (14,12),RC=0                    00546600
LM       R14,R12,12(R13)                  00546700
LA       R15,0(0,0)                       00546800
BR       R14                              00546900
*                                               00547000
* DATA--AREA                             00547100
*                                               00547200
DS       0F                               00547300
SAVEAREA DC 18F'0'                        SAVE AREAS ...
DS       0F                               00547400
ST0      DS  F                            R0    00547500
ST1      DS  F                            R1    00547600
ST2      DS  F                            R2    00547700
ST3      DS  F                            R3    00547800
ST4      DS  F                            R4    00547900
ST5      DS  F                            R5    00548000
ST6      DS  F                            R6    00548100
ST7      DS  F                            R7    00548200
ST8      DS  F                            R8    00548300
ST9      DS  F                            R9    00548400
ST15     DS  F                            R15   00548500
DS       0F                               00548600
VSSCH    DS  F                            VSAM-SWITCH (OPEN/CLOSE) 00548700
PL33     DS  F                            PL-33 DATA (53 BYTES)  00548800
PL34A    DS  F                            PL34-REC-CNT (S9999 COMP) 00548900
PL34T    DS  F                            PL-34-TAB (2004 LANG)  00549000
DS       0F                               SAVE INT.AREAS      00549100
RECAA    DS  F                            ADDR. MYFILE-RECORD 00549200
*                                               00549300
DS       0F                               00549400
P1       DC  P'1'                          00549500
* HALFWORD-NUMBERS                         00549600
* FULLWORD-NUMBERS                        00549700
DS       0F                               BINARY-REGISTER-NUMBERS FULLWORD 00549800
F1       DC  F'1'                          00549900
H0       DC  H'0'                          00550000
H1       DC  H'1'                          00550100
H1754    DC  H'1754'                       00550200
H2004    DC  H'2004'                       00550300
DS       0F                               00550400
F7       DC  F'7'                          00550500
F56      DC  F'56'                         00550600

```

```

F250      DC      F'250'                FOR MOVE > 250                00550700
F254      DC      F'254'                00550800
          DS      0D                    00550900
DWORDT    DS      D                    00551000
AR1       DS      10F                   AREA FOR SHOWCB           00551100
AR2       DS      CL4                   00551200
VSAMKEY   DS      CL9                   KEY FOR MYFILE             00551300
HILF      DS      CL16                  FOR UNPACK (SHOWCB)       00551400
          DS      0F                    00551500
FF        DC      X'FF'                 00551600
LAUS      DC      120X'40'              00551700
C00       DC      C'00'                 OK-STATUS                  00551800
C16       DC      C'16'                 REC-NOT-FOUND             00551900
C99       DC      C'99'                 00552000
          DS      0F                    00553000
          DC      C'SAMPLE02 CREATION : 08.11.1994,09.30 ' 00554001
          LTORG                          00555000
MYFILE    ACB     AM=VSAM,DDNAME=MYFILE,MACRF=(KEY,DIR,IN) 00556000
MYRPL     RPL     ACB=MYFILE,AM=VSAM,AREA=AR2,AREALEN=4,ARG=VSAMKEY,KEYLEN-00557000
          =9,OPTCD=(KEY,DIR,SYN,NUP,KEQ,FKS,LOC)           00558106
          END      SAMPLE02                          00559001

```