J E S 2 J O B L O G

```
18.12.13 JOB  153  IEF677I WARNING MESSAGE(S) FOR JOB VSTESTK5 ISSUED
18.12.13 JOB  153  $HASP373 VSTESTK5 STARTED - INIT  1 - CLASS A - SYS HMVS
18.12.13 JOB  153  IEF403I VSTESTK5 - STARTED - TIME=18.12.13
18.12.13 JOB  153  CCI001C PL1L    /IEMAA   /00:00:00.21/        /00004/SYS     /VSTESTK5
18.12.13 JOB  153  CCI001C LKED    /IEWL    /00:00:00.05/        /00000/SYS     /VSTESTK5
18.12.13 JOB  153  CCI001C GO      /PGM=*.DD/00:00:00.02/        /00000/SYS     /VSTESTK5
18.12.13 JOB  153  IEF404I VSTESTK5 - ENDED - TIME=18.12.13
18.12.13 JOB  153  $HASP395 VSTESTK5 ENDED

------ JES2 JOB STATISTICS ------

 07 JUL 20 JOB EXECUTION DATE

        33 CARDS READ

     1,592 SYSOUT PRINT RECORDS

         0 SYSOUT PUNCH RECORDS

      0.00 MINUTES EXECUTION TIME
```

```
 1      //VSTESTK5 JOB (SYS),'VSAMIOP IVP KSDSRAND',CLASS=A,MSGCLASS=X,         JOB  153
        //             REGION=4096K
        ***
        ********************************************************************
        *** PL/1 MODULE: KSDSRAND     VSAM DATASET: VSTESTKS.CLUSTER (KSDS)
        ***
        *** RANDOMLY ADDS/DELETES/CHANGES RECORDS IN INDEXED DATASET
        ********************************************************************
        ***
 2      //PL1F    EXEC PL1LFCLG,
        //             PARM='LOAD,NODECK,ATR,XREF,CHAR60,MACRO'
 3      XXPL1L   EXEC  PGM=IEMAA,PARM='LOAD,NODECK',REGION=52K                  00000100
 4      XXSTEPLIB  DD DSN=SYSC.LINKLIB,DISP=SHR                                 00000200
 5      //PL1L.SYSPRINT DD SYSOUT=*
        X/SYSPRINT DD  SYSOUT=A                                                 00000300
 6      XXSYSLIN   DD  DSNAME=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSSQ,            *00000400
        XX             SPACE=(80,(250,100))                                     00000500
 7      XXSYSUT3  DD  DSNAME=&&SYSUT3,UNIT=SYSDA,SPACE=(80,(250,250)),         *00000600
        XX             DCB=BLKSIZE=80                                           00000700
 8      XXSYSUT1   DD  DSNAME=&&SYSUT1,UNIT=SYSDA,SPACE=(1024,(60,60),,CONTIG),*00000800
        XX             SEP=(SYSUT3,SYSLIN),DCB=BLKSIZE=1024                     00000900
 9      //PL1L.SYSIN DD DSN=SYSC.VSAMIOP.SOURCE(KSDSRAND),DISP=SHR
10      //PL1L.SYSLIB DD DSN=SYSC.VSAMIOP.MACLIB,DISP=SHR
11      XXLKED   EXEC  PGM=IEWL,PARM='XREF,LIST',COND=(9,LT,PL1L),              *00001000
        XX             REGION=96K                                               00001100
12      //LKED.SYSLIB DD
        X/SYSLIB   DD  DSNAME=SYSC.PL1LIB,DISP=SHR                              00001201
13      //         DD  DSN=SYSC.LINKLIB,DISP=SHR
14      XXSYSLMOD  DD  DSNAME=&&GOSET(GO),DISP=(MOD,PASS),                      *00001300
        XX             UNIT=SYSDA,SPACE=(1024,(50,20,1),RLSE)                   00001400
15      XXSYSUT1   DD  DSNAME=&&SYSUT1,UNIT=SYSDA,SPACE=(1024,(200,20)),        *00001500
        XX             SEP=(SYSLMOD,SYSLIB),DCB=BLKSIZE=1024                    00001600
16      //LKED.SYSPRINT DD SYSOUT=*
        X/SYSPRINT DD  SYSOUT=A                                                 00001700
17      XXSYSLIN   DD  DSNAME=&&LOADSET,DISP=(OLD,DELETE)                       00001800
18      XX         DD  DDNAME=SYSIN                                             00001900
19      XXGO     EXEC  PGM=*.LKED.SYSLMOD,COND=((9,LT,LKED),(9,LT,PL1L))       00002000
20      //GO.STEPLIB DD DSN=SYSC.PL1LIB,DISP=SHR
        X/STEPLIB  DD  DSN=SYSC.LINKLIB,DISP=SHR                                00002102
21      XX         DD  DSN=SYSC.PL1LIB,DISP=SHR                                 00002202
22      XXSYSPRINT DD  SYSOUT=A                                                 00002300
23      //GO.PRINTR DD SYSOUT=*
24      //GO.SYSUDUMP DD SYSOUT=*
25      //GO.SYSPRINT DD SYSOUT=*
26      //GO.KSDSF01 DD DSN=PUB001.VSTESTKS.CLUSTER,DISP=OLD
27      //GO.IMAGES DD *
```

```
 STMT NO. MESSAGE
-
   19     IEF686I DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
IEF236I ALLOC. FOR VSTESTK5 PL1L PL1F
IEF237I 253  ALLOCATED TO STEPLIB
IEF237I 253  ALLOCATED TO SYS00350
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I 380  ALLOCATED TO SYSLIN
IEF237I 251  ALLOCATED TO SYSUT3
IEF237I 370  ALLOCATED TO SYSUT1
IEF237I 253  ALLOCATED TO SYSIN
IEF237I 253  ALLOCATED TO SYSLIB
IEF142I VSTESTK5 PL1L PL1F - STEP WAS EXECUTED - COND CODE 0004
IEF285I   SYSC.LINKLIB                                KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   UCSYSCPK                                    KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   JES2.JOB00153.SO0102                        SYSOUT
IEF285I   SYS20189.T181213.RA000.VSTESTK5.LOADSET     PASSED        *------298
IEF285I   VOL SER NOS= MVS380.
IEF285I   SYS20189.T181213.RA000.VSTESTK5.SYSUT3      DELETED       *------467
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T181213.RA000.VSTESTK5.SYSUT1      DELETED       *--------0
IEF285I   VOL SER NOS= MVS370.
IEF285I   SYSC.VSAMIOP.SOURCE                         KEPT          *--------4
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   SYSC.VSAMIOP.MACLIB                         KEPT          *-------27
IEF285I   VOL SER NOS= SYSCPK.
IEF373I STEP /PL1L    / START 20189.1812
IEF374I STEP /PL1L    / STOP  20189.1812 CPU    0MIN 00.21SEC SRB    0MIN 00.05SEC VIRT  4096K SYS   212K
**** JOBCARD READ 20189 18:12:13 ************************************************************************************
*                                      PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                              *
*  STEP NAME  PL1L        USER CORE        4096K   TAPES USED/IO 000/000000000   START   TIME  18:12:13    TCB TIME  00:00:00.21  *
*  PGM  NAME  IEMAA       SYSTEM CORE       212K   DISKS USED/IO 004/000000796   STOP    TIME  18:12:13    SRB TIME  00:00:00.05  *
*   COND CODE    0004     PRIVATE AREA SZ   4096K       ALLOC TIME  18:12:13      ELAPSED TIME                PGM LOAD  18:12:13   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004          4291   00:00:00.29            0             0           0             0              0             0             0  *
************************************************************************************************************************
*  CPU $ (     0.07) + EXCP $ (     1.07) + MEMORY $ (     2.45) = TOTAL $ (     3.59)                                  *
************************************************************************************************************************
IEF236I ALLOC. FOR VSTESTK5 LKED PL1F
IEF237I 253  ALLOCATED TO SYSLIB
IEF237I 253  ALLOCATED TO
IEF237I 253  ALLOCATED TO SYS00352
IEF237I 251  ALLOCATED TO SYSLMOD
IEF237I 370  ALLOCATED TO SYSUT1
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I 380  ALLOCATED TO SYSLIN
IEF237I DMY  ALLOCATED TO
IEF142I VSTESTK5 LKED PL1F - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYSC.PL1LIB                                 KEPT          *------108
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   SYSC.LINKLIB                                KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   UCSYSCPK                                    KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   SYS20189.T181213.RA000.VSTESTK5.GOSET       PASSED        *-------60
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T181213.RA000.VSTESTK5.SYSUT1      DELETED       *--------6
IEF285I   VOL SER NOS= MVS370.
IEF285I   JES2.JOB00153.SO0103                        SYSOUT
```

```
IEF285I   SYS20189.T181213.RA000.VSTESTK5.LOADSET      DELETED       *------299
IEF285I   VOL SER NOS= MVS380.
IEF373I STEP /LKED    / START 20189.1812
IEF374I STEP /LKED    / STOP  20189.1812 CPU    0MIN 00.05SEC SRB    0MIN 00.01SEC VIRT   260K SYS   168K
********************************************************************************************************************
*                                        PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                             *
*  STEP NAME  LKED        USER CORE          260K   TAPES USED/IO 000/000000000   START   TIME  18:12:13    TCB TIME  00:00:00.05  *
*  PGM  NAME  IEWL        SYSTEM CORE        168K   DISKS USED/IO 004/000000473   STOP    TIME  18:12:13    SRB TIME  00:00:00.01  *
*   COND CODE    0000     PRIVATE AREA SZ   4096K       ALLOC TIME  18:12:13      ELAPSED TIME                PGM LOAD  18:12:13   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004          2414   00:00:00.08            0             0         0              0              0             0  *
********************************************************************************************************************
*  CPU $ (     0.01) + EXCP $ (     0.63) + MEMORY $ (     0.03) = TOTAL $ (     0.67)                               *
********************************************************************************************************************
IEF236I ALLOC. FOR VSTESTK5 GO PL1F
IEF237I 251  ALLOCATED TO PGM=*.DD
IEF237I 253  ALLOCATED TO STEPLIB
IEF237I 253  ALLOCATED TO
IEF237I 253  ALLOCATED TO SYS00354
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO PRINTR
IEF237I JES2 ALLOCATED TO SYSUDUMP
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I 190  ALLOCATED TO KSDSF01
IEF237I 190  ALLOCATED TO SYS00356
IEF237I JES2 ALLOCATED TO IMAGES
IEF142I VSTESTK5 GO PL1F - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS20189.T181213.RA000.VSTESTK5.GOSET        KEPT          *--------0
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYSC.PL1LIB                                  KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   SYSC.PL1LIB                                  KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   UCSYSCPK                                     KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   JES2.JOB00153.SO0104                         SYSOUT
IEF285I   JES2.JOB00153.SO0105                         SYSOUT
IEF285I   JES2.JOB00153.SO0106                         SYSOUT
IEF285I   JES2.JOB00153.SO0107                         SYSOUT
IEF285I   PUB001.VSTESTKS.CLUSTER                      KEPT          *-------12
IEF285I   VOL SER NOS= PUB001.
IEF285I   UCPUB001                                     KEPT          *--------0
IEF285I   VOL SER NOS= PUB001.
IEF285I   JES2.JOB00153.SI0101                         SYSIN
IEF373I STEP /GO      / START 20189.1812
IEF374I STEP /GO      / STOP  20189.1812 CPU    0MIN 00.02SEC SRB    0MIN 00.00SEC VIRT   108K SYS   196K
********************************************************************************************************************
*                                        PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                             *
*  STEP NAME  GO          USER CORE          108K   TAPES USED/IO 000/000000000   START   TIME  18:12:13    TCB TIME  00:00:00.02  *
*  PGM  NAME  PGM=*.DD    SYSTEM CORE        196K   DISKS USED/IO 003/000000012   STOP    TIME  18:12:13    SRB TIME  00:00:00.00  *
*   COND CODE    0000     PRIVATE AREA SZ   4096K       ALLOC TIME  18:12:13      ELAPSED TIME                PGM LOAD  18:12:13   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004            92   00:00:00.02            0             0         0              0              0             0  *
********************************************************************************************************************
*  CPU $ (     0.00) + EXCP $ (     0.01) + MEMORY $ (     0.00) = TOTAL $ (     0.01)                               *
********************************************************************************************************************
IEF237I 251  ALLOCATED TO SYS00001
IEF285I   SYS20189.T181213.RA000.VSTESTK5.R0000001     KEPT          *--------0
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T181213.RA000.VSTESTK5.GOSET        DELETED
IEF285I   VOL SER NOS= WORK00.
```

```
IEF375I  JOB /VSTESTK5/ START 20189.1812
IEF376I  JOB /VSTESTK5/ STOP  20189.1812 CPU    0MIN 00.28SEC SRB    0MIN 00.06SEC
```

PL/I F COMPILER OPTIONS SPECIFIED ARE AS FOLLOWS--

```
     LOAD,NODECK,ATR,XREF,CHAR60,MACRO
```

THE COMPLETE LIST OF OPTIONS USED DURING THIS COMPILATION IS--

```
  EBCDIC
  CHAR60
  MACRO
  SOURCE2
NOMACDCK
  COMP
  SOURCE
  ATR
  XREF
NOEXTREF
NOLIST
  LOAD
NODECK
  FLAGW
NOSTMT
  SIZE=4154608
  LINECNT=050
  OPT=01
  SORMGIN=(002,072)
NOEXTDIC
NONEST
OPLIST
SYNCHKT
```

```
*OPTIONS IN EFFECT*     EBCDIC,CHAR60,MACRO,SOURCE2,NOMACDCK,COMP,SOURCE,ATR,XREF,NOEXTREF,NOLIST,LOAD,
*OPTIONS IN EFFECT*     NODECK,FLAGW,NOSTMT,SIZE=4154608,LINECNT=050,OPT=01,SORMGIN=(002,072),NOEXTDIC,
*OPTIONS IN EFFECT*     NONEST,OPLIST,SYNCHKT
```

COMPILE-TIME MACRO PROCESSOR
MACRO SOURCE2 LISTING

```
  1    /*****************************************************************08020000
  2                                                                      08030000
  3      KSDSRAND - TESTS THE VSAMIO ROUTINE BY READING A KSDS CLUSTER   08040000
  4                 RANDOMLY AND ADDING/UPDATING/DELETING RECORDS.       08050000
  5                                                                      08060000
  6     ******************************************************************/08070000
  7    KSDSRND:                                                          08080000
  8       PROCEDURE OPTIONS(MAIN);                                       08090000
  9                                                                      08100000
 10          ON ERROR                                                   08110000
 11             BEGIN;                                                  08120000
 12                ON ERROR SYSTEM;                                     08130000
 13                PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*');08140000
 14                PUT SKIP DATA;                                       08150000
 15                PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*');08160000
 16             END;                                                    08170000
 17                                                                     08180000
 18          OPEN                                                       08190000
 19             FILE(IMAGES),                                           08200000
 20             FILE(PRINTR) LINESIZE(133);                             08210000
 21                                                                     08220000
 22          ON ENDFILE(IMAGES)                                         08230000
 23             MORE_RECORDS = NO;                                      08240000
 24                                                                     08250000
 25          PRINT_AREA = 'KSDSRAND: READ/UPDATE KSDS DIRECT';          08260000
 26          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       08270000
 27          PRINT_AREA = '-------------------------------';            08280000
 28          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       08290000
 29          PRINT_AREA = ' ';                                          08300000
 30          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       08310000
 31                                                                     08320000
 32          MORE_RECORDS = YES;                                        08330000
 33                                                                     08340000
 34    /*****************************************************************08350000
 35      ESTABLISH PARAMETERS FOR VSAM DATASET AND CALL ROUTINE TO OPEN  08360000
 36     ******************************************************************/08370000
 37          VSFB_DDNAME = 'KSDSF01';                                   08380000
 38          VSFB_ORGANIZATION = VSIO_KSDS;                             08390000
 39          VSFB_ACCESS = VSIO_DIRECT;                                 08400000
 40          VSFB_MODE = VSIO_INPUT_OUTPUT;                             08410000
 41          VSFB_RECORD_LENGTH = 80;                                   08420000
 42          VSFB_KEY_POSITION = 0;                                     08430000
 43          VSFB_KEY_LENGTH = 10;                                      08440000
 44          VSIO_COMMAND = VSIO_OPEN;                                  08450000
```

MACRO SOURCE2 LISTING

```
 45          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                          08460000
 46                        VSIO_FILE_BLOCK,                               08470000
 47                        VSAM_RECORD);                                  08480000
 48          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                08490000
 49             DO;                                                       08500000
 50                CALL VSIO_ERROR;                                       08510000
 51                RETURN;                                                08520000
 52             END;                                                      08530000
 53                                                                       08540000
 54          DO WHILE(MORE_RECORDS);                                      08550000
 55             READ FILE(IMAGES) INTO(MAINT_RECORD);                     08560000
 56             IF (MORE_RECORDS) THEN                                    08570000
 57                CALL PROCESS_MAINT;                                    08580000
 58          END;                                                         08590000
 59                                                                       08600000
 60          CLOSE FILE(IMAGES);                                          08610000
 61                                                                       08620000
 62     /*****************************************************************08630000
 63       CALL ROUTINE TO CLOSE VSAM DATASET                              08640000
 64      *****************************************************************/08650000
 65          VSIO_COMMAND = VSIO_CLOSE;                                   08660000
 66          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                          08670000
 67                        VSIO_FILE_BLOCK,                               08680000
 68                        VSAM_RECORD);                                  08690000
 69          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                08700000
 70             CALL VSIO_ERROR;                                          08710000
 71                                                                       08720000
 72          COUNTER_EDIT = ADD_COUNTER;                                  08730000
 73          PRINT_AREA = COUNTER_EDIT || ' RECORDS ADDED';               08740000
 74          WRITE FILE(PRINTR) FROM(PRINT_LINE);                         08750000
 75          COUNTER_EDIT = CHANGE_COUNTER;                               08760000
 76          PRINT_AREA = COUNTER_EDIT || ' RECORDS CHANGED';             08770000
 77          WRITE FILE(PRINTR) FROM(PRINT_LINE);                         08780000
 78          COUNTER_EDIT = DELETE_COUNTER;                               08790000
 79          PRINT_AREA = COUNTER_EDIT || ' RECORDS DELETED';             08800000
 80          WRITE FILE(PRINTR) FROM(PRINT_LINE);                         08810000
 81                                                                       08820000
 82          RETURN;                                                      08830000
 83                                                                       08840000
 84     PROCESS_MAINT:                                                    08850000
 85        PROCEDURE;                                                     08860000
 86                                                                       08870000
 87          RECORD_COUNTER = RECORD_COUNTER + 1;                         08880000
 88                                                                       08890000
 89          IF (MAINT_ACTION = 'A') THEN                                 08900000
```

MACRO SOURCE2 LISTING

```
 90            CALL PROCESS_ADD;                                        08910000
 91         ELSE                                                        08920000
 92            IF (MAINT_ACTION = 'C') THEN                             08930000
 93               CALL PROCESS_CHANGE;                                  08940000
 94            ELSE                                                     08950000
 95               IF (MAINT_ACTION = 'D') THEN                          08960000
 96                  CALL PROCESS_DELETE;                               08970000
 97               ELSE                                                  08980000
 98                  DO;                                                08990000
 99                     COUNTER_EDIT = RECORD_COUNTER;                  09000000
100                     PRINT_AREA = COUNTER_EDIT || ': INVALID ACTION: ' || 09010000
101                                  MAINT_RECORD_SCALAR;               09020000
102                     WRITE FILE(PRINTR) FROM(PRINT_LINE);            09030000
103                  END;                                               09040000
104                                                                     09050000
105         RETURN;                                                     09060000
106                                                                     09070000
107      END PROCESS_MAINT;                                             09080000
108                                                                     09090000
109   PROCESS_ADD:                                                      09100000
110      PROCEDURE;                                                     09110000
111                                                                     09120000
112         VSAM_RECORD_SCALAR = MAINT_IMAGE;                           09130000
113         PRINT_AREA = 'KEY: ' || VSAM_RECORD_KEY || '  DATA: ' ||    09140000
114                      VSAM_RECORD_SCALAR || '  ADDING';              09150000
115         WRITE FILE(PRINTR) FROM(PRINT_LINE);                        09160000
116         CALL WRITE_KS;                                              09170000
117         IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN                09180000
118            DO;                                                      09190000
119               PRINT_AREA = (105)' ' || 'ADDED OK';                  09200000
120               WRITE FILE(PRINTR) FROM(PRINT_LINE);                  09210000
121            END;                                                     09220000
122         ELSE                                                        09230000
123            IF (VSIO_VSAM_FEEDBACK = VSIO_FB_DUPLICATE_RECORD) THEN  09240000
124               DO;                                                   09250000
125                  PRINT_AREA = (105)' ' || 'DUPLICATE RECORD';       09260000
126                  WRITE FILE(PRINTR) FROM(PRINT_LINE);               09270000
127               END;                                                  09280000
128                                                                     09290000
129         RETURN;                                                     09300000
130                                                                     09310000
131      END PROCESS_ADD;                                               09320000
132                                                                     09330000
133   PROCESS_CHANGE:                                                   09340000
134      PROCEDURE;                                                     09350000
```

MACRO SOURCE2 LISTING

```
135                                                                   09360000
136          VSAM_RECORD_SCALAR = MAINT_IMAGE;                        09370000
137          CALL READ_KS;                                            09380000
138          IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN             09390000
139             CALL COMPLETE_CHANGE;                                 09400000
140          ELSE                                                     09410000
141             DO;                                                   09420000
142                PRINT_AREA = 'READING FOR UPDATE: ' ||             09430000
143                             VSAM_RECORD_KEY ||                    09440000
144                             '  *** NOT FOUND ***';                09450000
145                WRITE FILE(PRINTR) FROM(PRINT_LINE);               09460000
146             END;                                                  09470000
147                                                                   09480000
148          RETURN;                                                  09490000
149                                                                   09500000
150       END PROCESS_CHANGE;                                         09510000
151                                                                   09520000
152    COMPLETE_CHANGE:                                               09530000
153       PROCEDURE;                                                  09540000
154                                                                   09550000
155          PRINT_AREA = 'KEY: ' || VSAM_RECORD_KEY || '  DATA: ' || 09560000
156                       VSAM_RECORD_SCALAR || '  BEFORE';          09570000
157          WRITE FILE(PRINTR) FROM(PRINT_LINE);                     09580000
158          VSAM_RECORD_SCALAR = MAINT_IMAGE;                        09590000
159          CALL REWRITE_KS;                                         09600000
160          IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN             09610000
161             DO;                                                   09620000
162                PRINT_AREA = (23)' ' || VSAM_RECORD_SCALAR || '  AFTER';  09630000
163                WRITE FILE(PRINTR) FROM(PRINT_LINE);               09640000
164             END;                                                  09650000
165                                                                   09660000
166          RETURN;                                                  09670000
167                                                                   09680000
168       END COMPLETE_CHANGE;                                        09690000
169                                                                   09700000
170    PROCESS_DELETE:                                                09710000
171       PROCEDURE;                                                  09720000
172                                                                   09730000
173          VSAM_RECORD_SCALAR = MAINT_IMAGE;                        09740000
174          CALL READ_KS;                                            09750000
175          IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN             09760000
176             CALL COMPLETE_DELETE;                                 09770000
177          ELSE                                                     09780000
178             DO;                                                   09790000
179                PRINT_AREA = 'READING FOR DELETE: ' ||             09800000
```

MACRO SOURCE2 LISTING

```
                          VSAM_RECORD_KEY ||                          09810000
                          '  *** NOT FOUND ***';                      09820000
               WRITE FILE(PRINTR) FROM(PRINT_LINE);                   09830000
            END;                                                      09840000
                                                                      09850000
         RETURN;                                                      09860000
                                                                      09870000
      END PROCESS_DELETE;                                             09880000
                                                                      09890000
   COMPLETE_DELETE:                                                   09900000
      PROCEDURE;                                                      09910000
                                                                      09920000
         PRINT_AREA = 'KEY: ' || VSAM_RECORD_KEY || '  DATA: ' ||     09930000
                      VSAM_RECORD_SCALAR || '  BEFORE';              09940000
         WRITE FILE(PRINTR) FROM(PRINT_LINE);                         09950000
         CALL DELETE_KS;                                              09960000
         IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN                 09970000
            DO;                                                       09980000
               PRINT_AREA = (105)' ' || 'DELETED';                    09990000
               WRITE FILE(PRINTR) FROM(PRINT_LINE);                   10000000
            END;                                                      10010000
                                                                      10020000
         RETURN;                                                      10030000
                                                                      10040000
      END COMPLETE_DELETE;                                            10050000
                                                                      10060000
   DELETE_KS:                                                         10070000
      PROCEDURE;                                                      10080000
                                                                      10090000
   /*****************************************************************10100000
     CALL ROUTINE TO DELETE LAST RECORD READ FROM VSAM DATASET        10110000
    *****************************************************************/10120000
        VSIO_COMMAND = VSIO_DELETE;                                   10130000
        CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                           10140000
                      VSIO_FILE_BLOCK,                                10150000
                      VSAM_RECORD);                                   10160000
        IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                 10170000
           CALL VSIO_ERROR;                                           10180000
        ELSE                                                          10190000
           DELETE_COUNTER = DELETE_COUNTER + 1;                       10200000
                                                                      10210000
        RETURN;                                                       10220000
                                                                      10230000
      END DELETE_KS;                                                  10240000
                                                                      10250000
```

```
225    READ_KS:                                                          10260000
226       PROCEDURE;                                                     10270000
227                                                                      10280000
228    /*****************************************************************10290000
229      CALL ROUTINE TO READ NEXT RECORD FROM VSAM DATASET              10300000
230     *****************************************************************/10310000
231          VSIO_COMMAND = VSIO_READ;                                   10320000
232          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                         10330000
233                        VSIO_FILE_BLOCK,                              10340000
234                        VSAM_RECORD);                                 10350000
235          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN               10360000
236             IF (VSIO_VSAM_FEEDBACK ^= VSIO_FB_RECORD_NOT_FOUND) THEN 10370000
237                CALL VSIO_ERROR;                                      10380000
238                                                                      10390000
239          RETURN;                                                     10400000
240                                                                      10410000
241       END READ_KS;                                                   10420000
242                                                                      10430000
243    REWRITE_KS:                                                       10440000
244       PROCEDURE;                                                     10450000
245                                                                      10460000
246    /*****************************************************************10470000
247      CALL ROUTINE TO REWRITE PREVIOUSLY READ RECORD TO VSAM DATASET  10480000
248     *****************************************************************/10490000
249          VSIO_COMMAND = VSIO_REWRITE;                                10500000
250          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                         10510000
251                        VSIO_FILE_BLOCK,                              10520000
252                        VSAM_RECORD);                                 10530000
253          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN               10540000
254             CALL VSIO_ERROR;                                         10550000
255          ELSE                                                        10560000
256             CHANGE_COUNTER = CHANGE_COUNTER + 1;                     10570000
257                                                                      10580000
258          RETURN;                                                     10590000
259                                                                      10600000
260       END REWRITE_KS;                                                10610000
261                                                                      10620000
262    WRITE_KS:                                                         10630000
263       PROCEDURE;                                                     10640000
264                                                                      10650000
265    /*****************************************************************10660000
266      CALL ROUTINE TO WRITE RECORD INTO VSAM DATASET                  10670000
267     *****************************************************************/10680000
268          VSIO_COMMAND = VSIO_WRITE;                                  10690000
269          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                         10700000
```

```
270                      VSIO_FILE_BLOCK,                                10710000
271                      VSAM_RECORD);                                   10720000
272          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN               10730000
273             IF (VSIO_VSAM_FEEDBACK ^= VSIO_FB_DUPLICATE_RECORD) THEN 10740000
274                CALL VSIO_ERROR;                                     10750000
275             ELSE                                                    10760000
276                ;                                                    10770000
277          ELSE                                                       10780000
278             ADD_COUNTER = ADD_COUNTER + 1;                          10790000
279                                                                     10800000
280          RETURN;                                                    10810000
281                                                                     10820000
282       END WRITE_KS;                                                 10830000
283                                                                     10840000
284    VSIO_ERROR:                                                      10850000
285       PROCEDURE;                                                    10860000
286          PRINT_AREA = 'VSAMIO ERROR OCCURRED DURING ' ||            10870000
287                       VSIO_COMMAND;                                 10880000
288          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       10890000
289          PRINT_AREA = 'VSIO_RETURN_CODE = ' ||                      10900000
290                       VSIO_RETURN_CODE;                             10910000
291          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       10920000
292          PRINT_AREA = 'VSIO_VSAM_RETURN_CODE = ' ||                 10930000
293                       VSIO_VSAM_RETURN_CODE;                        10940000
294          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       10950000
295          PRINT_AREA = 'VSIO_VSAM_FUNCTION_CODE = ' ||               10960000
296                       VSIO_VSAM_FUNCTION_CODE;                      10970000
297          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       10980000
298          PRINT_AREA = 'VSIO_VSAM_FEEDBACK_CODE = ' ||               10990000
299                       VSIO_VSAM_FEEDBACK_CODE;                      11000000
300          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       11010000
301          PRINT_AREA = ' ';                                          11020000
302                                                                     11030000
303          RETURN;                                                    11040000
304                                                                     11050000
305       END VSIO_ERROR;                                               11060000
306                                                                     11070000
307       DECLARE                                                       11080000
308          IMAGES FILE INPUT RECORD SEQUENTIAL EXTERNAL               11090000
309             ENV(F),                                                 11100000
310          PRINTR FILE OUTPUT RECORD SEQUENTIAL EXTERNAL              11110000
311             ENV(F CTLASA);                                          11120000
312                                                                     11130000
313       DECLARE                                                       11140000
314          ADD_COUNTER                    FIXED BINARY(15,0) INIT(0),  11150000
```

MACRO SOURCE2 LISTING

```
315          CHANGE_COUNTER                 FIXED BINARY(15,0) INIT(0),      11160000
316          COUNTER_EDIT                   PICTURE 'ZZ,ZZZ,ZZ9V',           11170000
317          DELETE_COUNTER                 FIXED BINARY(15,0) INIT(0),      11180000
318          MORE_RECORDS                   BIT(1),                          11190000
319          NO                             BIT(1)    INIT('0'B),            11200000
320          YES                            BIT(1)    INIT('1'B);            11210000
321                                                                          11220000
322       DECLARE                                                            11230000
323          1 MAINT_RECORD,                                                 11240000
324             2 MAINT_ACTION              CHAR(1),                         11250000
325             2 MAINT_SKIP                CHAR(1),                         11260000
326             2 MAINT_IMAGE               CHAR(78);                        11270000
327                                                                          11280000
328       DECLARE                                                            11290000
329          MAINT_RECORD_SCALAR             DEFINED MAINT_RECORD           11300000
330                                          CHAR(80);                       11310000
331                                                                          11320000
332       DECLARE                                                            11330000
333          1 VSAM_RECORD,                                                  11340000
334             2 VSAM_RECORD_KEY           CHAR(10),                        11350000
335             2 VSAM_RECORD_FIELDS        CHAR(70);                        11360000
336                                                                          11370000
337       DECLARE                                                            11380000
338          VSAM_RECORD_SCALAR              DEFINED VSAM_RECORD            11390000
339                                          CHAR(80);                       11400000
340                                                                          11410000
341       DECLARE                                                            11420000
342          1 PRINT_LINE,                                                   11430000
343             2 CARRIAGE_CONTROL          CHAR(1)   INIT(' '),             11440000
344             2 PRINT_AREA                CHAR(132);                       11450000
345                                                                          11460000
346    %INCLUDE (VSAMIO);                                                    11470000
347    %INCLUDE (VSAMIOFB);                                                  11480000
348                                                                          11490000
349       END KSDSRND;                                                       11500000
```

INCLUDED TEXT FOLLOWS FROM DD.MEMBER = SYSLIB .VSAMIO

```
350    /*****************************************************************31100000
351                                                                          31110000
352             VV   VV    SSSSS       A      M     M    IIII      OOOOO       31120000
353             VV   VV   SS   SS     AAA     MM   MM     II      OO   OO      31130000
354             VV   VV   SS         AA AA    MMM MMM     II      OO   OO      31140000
355             VV   VV    SSSSS    AA   AA   MMMMMMM     II      OO   OO      31150000
```

MACRO SOURCE2 LISTING

```
356            VV   VV        SS   AA   AA   MM M MM     II      OO   OO      31160000
357             VV VV    SS   SS   AAAAAAA   MM   MM     II      OO   OO      31170000
358              VVV     SS   SS   AA   AA   MM   MM     II      OO   OO      31180000
359               V       SSSSS    AA   AA   MM   MM    IIII      OOOOO       31190000
360                                                                            31200000
361     ***************************************************************      31210000
362      THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS   31220000
363      ROUTINE.                                                              31230000
364                                                                            31240000
365      THE VSIO_PARAMETER_VALUES SUPPLY THE VALUES USED TO MOVE INTO         31250000
366      PARAMETER ENTRIES TO TAILOR THE ROUTINE TO A SPECIFIC DATASET AND     31260000
367      TO PROVIDE COMMANDS TO DRIVE THE ROUTINE.                             31270000
368     ***************************************************************      /31280000
369                                                                            31290000
370             DECLARE                                                        31300000
371                1 VSIO_PARAMETER_VALUES   STATIC,                           31310000
372                   2 VSIO_OPEN            CHAR(8)   INIT('OPEN    '),      31320000
373                   2 VSIO_CLOSE           CHAR(8)   INIT('CLOSE   '),      31330000
374                   2 VSIO_READ            CHAR(8)   INIT('READ    '),      31340000
375                   2 VSIO_WRITE           CHAR(8)   INIT('WRITE   '),      31350000
376                   2 VSIO_REWRITE         CHAR(8)   INIT('REWRITE '),      31360000
377                   2 VSIO_DELETE          CHAR(8)   INIT('DELETE  '),      31370000
378                   2 VSIO_START_EQUAL     CHAR(8)   INIT('STARTEQ '),      31380000
379                   2 VSIO_START_NOTLESS   CHAR(8)   INIT('STARTGE '),      31390000
380                   2 VSIO_KSDS            CHAR(4)   INIT('KSDS'),          31400000
381                   2 VSIO_ESDS            CHAR(4)   INIT('ESDS'),          31410000
382                   2 VSIO_RRDS            CHAR(4)   INIT('RRDS'),          31420000
383                   2 VSIO_SEQUENTIAL      CHAR(10)  INIT('SEQUENTIAL'),    31430000
384                   2 VSIO_DIRECT          CHAR(10)  INIT('DIRECT    '),    31440000
385                   2 VSIO_DYNAMIC         CHAR(10)  INIT('DYNAMIC   '),    31450000
386                   2 VSIO_INPUT           CHAR(6)   INIT('INPUT '),        31460000
387                   2 VSIO_OUTPUT          CHAR(6)   INIT('OUTPUT'),        31470000
388                   2 VSIO_INPUT_OUTPUT    CHAR(6)   INIT('UPDATE'),        31480000
389                   2 (VSIO_RC_SUCCESS               INIT(0),               31490000
390                      VSIO_RC_LOGIC_ERROR           INIT(8),               31500000
391                      VSIO_RC_END_OF_FILE           INIT(9999),            31510000
392                      VSIO_RC_UNKNOWN_COMMAND       INIT(20),              31520000
393                      VSIO_RC_DATASET_ALREADY_OPEN  INIT(21),              31530000
394                      VSIO_RC_DATASET_NOT_OPEN      INIT(22),              31540000
395                      VSIO_RC_ORGANIZATION_UNKNOWN  INIT(23),              31550000
396                      VSIO_RC_ACCESS_UNKNOWN        INIT(24),              31560000
397                      VSIO_RC_ORG_ACCESS_MISMATCH   INIT(25),              31570000
398                      VSIO_RC_MODE_UNKNOWN          INIT(26),              31580000
399                      VSIO_RC_MODE_UNSUPPORTED      INIT(27),              31590000
400                      VSIO_RC_DDNAME_BLANK          INIT(28))              31600000
```

MACRO SOURCE2 LISTING

```
                                      FIXED BINARY(15,0),               31610000
               2 (VSIO_FB_DUPLICATE_RECORD      INIT(8),                31620000
                  VSIO_FB_KEY_SEQUENCE          INIT(12),               31630000
                  VSIO_FB_RECORD_NOT_FOUND      INIT(16),               31640000
                  VSIO_FB_NO_MORE_SPACE         INIT(28),               31650000
                  VSIO_FB_READ_WITHOUT_START    INIT(88))               31660000
                                      FIXED BINARY(15,0),               31670000
/**********************************************************************31680000
  THE VSIO_PARAMETER_BLOCK IS THE COMMUNICATION INTERFACE TO THE       31690000
  THE ROUTINE.                                                         31700000
 *********************************************************************/31710000
                                                                       31720000
            1 VSIO_PARAMETER_BLOCK    STATIC,                          31730000
               2 VSIO_COMMAND         CHAR(8)   INIT(' '),             31740000
               2 (VSIO_RETURN_CODE,                                    31750000
                  VSIO_VSAM_RC,                                        31760000
                  VSIO_VSAM_FUNCTION,                                  31770000
                  VSIO_VSAM_FEEDBACK) FIXED BINARY(15,0) INIT(0);      31780000
                                                                       31790000
/**********************************************************************31800000
                          END OF VSAMIO COPY BOOK                      31810000
 *********************************************************************/31820000
```

INCLUDED TEXT FOLLOWS FROM DD.MEMBER = SYSLIB .VSAMIOFB

```
/**********************************************************************00000100
                                                                       00000200
   VV   VV   SSSSS      A     M     M  IIII   OOOOO   FFFFFFF  BBBBBB  00000300
   VV   VV  SS   SS    AAA    MM   MM   II   OO   OO  FF       BB   BB 00000400
   VV   VV  SS         AA AA   MMM MMM   II   OO   OO  FF       BB   BB 00000500
   VV   VV   SSSSS   AA   AA  MMMMMMM   II   OO   OO  FFFFF    BBBBBB  00000600
   VV   VV       SS  AA   AA  MM M MM   II   OO   OO  FF       BB   BB 00000700
    VV VV   SS   SS  AAAAAAA  MM   MM   II   OO   OO  FF       BB   BB 00000800
     VVV    SS   SS  AA   AA  MM   MM   II   OO   OO  FF       BB   BB 00000900
      V      SSSSS   AA   AA  MM   MM  IIII   OOOOO   FF       BBBBBB  00001000
                                                                       00001100
 **********************************************************************00001200
   THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS 00001300
   ROUTINE, AND ARE USED TO COMMUNICATE CHARACTERISTICS FOR A SINGLE   00001400
   VSAM DATASET.                                                       00001500
                                                                       00001600
   WITH THE 2 EXCEPTIONS FOR RECORD LENGTH (TO ACCOMODATE VARIABLE     00001700
   LENGTH RECORDS) AND RELATIVE RECORD (TO ACCOMODATE RELATIVE RECORD  00001800
   DATASETS), THESE DATA NAMES MUST BE POPULATED PRIOR TO CALLING THE  00001900
```

MACRO SOURCE2 LISTING

```
442       ROUTINE TO OPEN THE DATASET AND MUST NOT THEN BE CHANGED UNTIL THE  00002000
443       DATASET HAS BEEN CLOSED.                                            00002100
444     *****************************************************************/00002200
445                                                                         00002300
446          DECLARE                                                        00002400
447             1 VSIO_FILE_BLOCK            STATIC,                        00002500
448                2 VSFB_DDNAME             CHAR(8)   INIT(' '),           00002600
449                2 VSFB_ORGANIZATION       CHAR(4)   INIT(' '),           00002700
450                2 VSFB_ACCESS             CHAR(10)  INIT(' '),           00002800
451                2 VSFB_MODE               CHAR(6)   INIT(' '),           00002900
452                2 (VSFB_RECORD_LENGTH,                                   00003000
453                   VSFB_KEY_POSITION,                                    00003100
454                   VSFB_KEY_LENGTH)       FIXED BINARY(15,0) INIT(0),    00003200
455                2 VSFB_FILE_STATUS        CHAR(1)   INIT('C'),           00003300
456                2 VSFB_RESERVED           CHAR(161);                     00003400
457                                                                         00003500
458     /*****************************************************************00003600
459                           END OF VSAMIOFB COPY BOOK                     00003700
460      *****************************************************************/00003800
```

NO ERROR OR WARNING CONDITION HAS BEEN DETECTED FOR THIS MACRO PASS.

SOURCE LISTING.

```
                /*****************************************************************    1
                                                                                       1
                  KSDSRAND - TESTS THE VSAMIO ROUTINE BY READING A KSDS CLUSTER        1
                             RANDOMLY AND ADDING/UPDATING/DELETING RECORDS.            1
                                                                                       1
                 *****************************************************************/   1
                                                                                       6
  1              KSDSRND:                                                             7
                    PROCEDURE OPTIONS(MAIN);                                           8
                                                                                       9
  2                    ON ERROR                                                     10
  3                       BEGIN;                                                    11
  4                          ON ERROR SYSTEM;                                       12
  5                          PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*'); 13
  6                          PUT SKIP DATA;                                         14
  7                          PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*'); 15
  8                       END;                                                      16
                                                                                    17
  9                    OPEN                                                         18
                          FILE(IMAGES),                                             19
                          FILE(PRINTR) LINESIZE(133);                               20
                                                                                    21
 10                    ON ENDFILE(IMAGES)                                           22
 11                       MORE_RECORDS = NO;                                        23
                                                                                    24
 12                    PRINT_AREA = 'KSDSRAND: READ/UPDATE KSDS DIRECT';            25
 13                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                         26
 14                    PRINT_AREA = '---------------------------------';            27
 15                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                         28
 16                    PRINT_AREA = ' ';                                            29
 17                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                         30
                                                                                    31
 18                    MORE_RECORDS = YES;                                          32
                                                                                    33
                 /*****************************************************************   34
                   ESTABLISH PARAMETERS FOR VSAM DATASET AND CALL ROUTINE TO OPEN       34
                  *****************************************************************/   34
                                                                                       36
 19                    VSFB_DDNAME = 'KSDSF01';                                     37
 20                    VSFB_ORGANIZATION = VSIO_KSDS;                               38
 21                    VSFB_ACCESS = VSIO_DIRECT;                                   39
 22                    VSFB_MODE = VSIO_INPUT_OUTPUT;                               40
 23                    VSFB_RECORD_LENGTH = 80;                                     41
```

```
                    VSFB_KEY_POSITION = 0;                                          42
                    VSFB_KEY_LENGTH = 10;                                           43
                    VSIO_COMMAND = VSIO_OPEN;                                       44
                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                             45
                                  VSIO_FILE_BLOCK,                                  46
                                  VSAM_RECORD);                                     47
                    IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                   48
                       DO;                                                          49
                          CALL VSIO_ERROR;                                          50
                          RETURN;                                                   51
                       END;                                                         52
                                                                                   53
                    DO WHILE(MORE_RECORDS);                                         54
                       READ FILE(IMAGES) INTO(MAINT_RECORD);                        55
                       IF (MORE_RECORDS) THEN                                       56
                          CALL PROCESS_MAINT;                                       57
                    END;                                                            58
                                                                                   59
                    CLOSE FILE(IMAGES);                                             60
                                                                                   61
                 /*****************************************************************  62
                   CALL ROUTINE TO CLOSE VSAM DATASET                               62
                  *****************************************************************/  62
                                                                                   64
                    VSIO_COMMAND = VSIO_CLOSE;                                      65
                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                             66
                                  VSIO_FILE_BLOCK,                                  67
                                  VSAM_RECORD);                                     68
                    IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                   69
                       CALL VSIO_ERROR;                                             70
                                                                                   71
                    COUNTER_EDIT = ADD_COUNTER;                                     72
                    PRINT_AREA = COUNTER_EDIT || ' RECORDS ADDED';                  73
                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                            74
                    COUNTER_EDIT = CHANGE_COUNTER;                                  75
                    PRINT_AREA = COUNTER_EDIT || ' RECORDS CHANGED';                76
                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                            77
                    COUNTER_EDIT = DELETE_COUNTER;                                  78
                    PRINT_AREA = COUNTER_EDIT || ' RECORDS DELETED';                79
                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                            80
                                                                                   81
                    RETURN;                                                         82
                                                                                   83
                 PROCESS_MAINT:                                                     84
                    PROCEDURE;                                                      85
                                                                                   86
                    RECORD_COUNTER = RECORD_COUNTER + 1;                            87
```

```
                                                                                88
55                  IF (MAINT_ACTION = 'A') THEN                                89
56                     CALL PROCESS_ADD;                                        90
57                  ELSE                                                        91
57                     IF (MAINT_ACTION = 'C') THEN                             92
58                        CALL PROCESS_CHANGE;                                  93
59                     ELSE                                                     94
59                        IF (MAINT_ACTION = 'D') THEN                          95
60                           CALL PROCESS_DELETE;                               96
61                        ELSE                                                  97
61                           DO;                                                98
62                              COUNTER_EDIT = RECORD_COUNTER;                  99
63                              PRINT_AREA = COUNTER_EDIT || ': INVALID ACTION: ' ||  100
                                             MAINT_RECORD_SCALAR;               101
64                              WRITE FILE(PRINTR) FROM(PRINT_LINE);            102
65                           END;                                               103
                                                                                104
66                  RETURN;                                                     105
                                                                                106
67               END PROCESS_MAIN;                                              107
                                                                                108
68            PROCESS_ADD:                                                      109
                 PROCEDURE;                                                     110
                                                                                111
69                  VSAM_RECORD_SCALAR = MAINT_IMAGE;                           112
70                  PRINT_AREA = 'KEY: ' || VSAM_RECORD_KEY || '  DATA: ' ||    113
                                 VSAM_RECORD_SCALAR || '  ADDING';              114
71                  WRITE FILE(PRINTR) FROM(PRINT_LINE);                        115
72                  CALL WRITE_KS;                                              116
73                  IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN                117
74                     DO;                                                      118
75                        PRINT_AREA = (105)' ' || 'ADDED OK';                  119
76                        WRITE FILE(PRINTR) FROM(PRINT_LINE);                  120
77                     END;                                                     121
78                  ELSE                                                        122
78                     IF (VSIO_VSAM_FEEDBACK = VSIO_FB_DUPLICATE_RECORD) THEN  123
79                        DO;                                                   124
80                           PRINT_AREA = (105)' ' || 'DUPLICATE RECORD';       125
81                           WRITE FILE(PRINTR) FROM(PRINT_LINE);               126
82                        END;                                                  127
                                                                                128
83                  RETURN;                                                     129
                                                                                130
84                END PROCESS_ADD;                                              131
                                                                                132
85            PROCESS_CHANGE:                                                   133
                 PROCEDURE;                                                     134
```

```
  86             VSAM_RECORD_SCALAR = MAINT_IMAGE;
  87             CALL READ_KS;
  88             IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN
  89                CALL COMPLETE_CHANGE;
  90             ELSE
  90                DO;
  91                   PRINT_AREA = 'READING FOR UPDATE: ' ||
                                    VSAM_RECORD_KEY ||
                                    '  *** NOT FOUND ***';
  92                   WRITE FILE(PRINTR) FROM(PRINT_LINE);
  93                END;

  94             RETURN;

  95          END PROCESS_CHANGE;

  96       COMPLETE_CHANGE:
              PROCEDURE;

  97             PRINT_AREA = 'KEY: ' || VSAM_RECORD_KEY || '  DATA: ' ||
                              VSAM_RECORD_SCALAR || '  BEFORE';
  98             WRITE FILE(PRINTR) FROM(PRINT_LINE);
  99             VSAM_RECORD_SCALAR = MAINT_IMAGE;
 100             CALL REWRITE_KS;
 101             IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN
 102                DO;
 103                   PRINT_AREA = (23)' ' || VSAM_RECORD_SCALAR || '  AFTER';
 104                   WRITE FILE(PRINTR) FROM(PRINT_LINE);
 105                END;

 106             RETURN;

 107          END COMPLETE_CHANGE;

 108       PROCESS_DELETE:
              PROCEDURE;

 109             VSAM_RECORD_SCALAR = MAINT_IMAGE;
 110             CALL READ_KS;
 111             IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN
 112                CALL COMPLETE_DELETE;
 113             ELSE
 113                DO;
 114                   PRINT_AREA = 'READING FOR DELETE: ' ||
                                    VSAM_RECORD_KEY ||
                                    '  *** NOT FOUND ***';
```

```
115                        WRITE FILE(PRINTR) FROM(PRINT_LINE);                  182
116                     END;                                                     183
                                                                                 184
117                     RETURN;                                                  185
                                                                                 186
118                  END PROCESS_DELETE;                                         187
                                                                                 188
119               COMPLETE_DELETE:                                               189
                     PROCEDURE;                                                  190
                                                                                 191
120                     PRINT_AREA = 'KEY: ' || VSAM_RECORD_KEY || '  DATA: ' || 192
                                    VSAM_RECORD_SCALAR || '  BEFORE';           193
121                     WRITE FILE(PRINTR) FROM(PRINT_LINE);                     194
122                     CALL DELETE_KS;                                          195
123                     IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN             196
124                        DO;                                                   197
125                           PRINT_AREA = (105)' ' || 'DELETED';                198
126                           WRITE FILE(PRINTR) FROM(PRINT_LINE);               199
127                        END;                                                  200
                                                                                 201
128                     RETURN;                                                  202
                                                                                 203
129                  END COMPLETE_DELETE;                                        204
                                                                                 205
130               DELETE_KS:                                                     206
                     PROCEDURE;                                                  207
                                                                                 208
                  /*************************************************************209
                    CALL ROUTINE TO DELETE LAST RECORD READ FROM VSAM DATASET    209
                   *************************************************************/209
                                                                                 211
131                     VSIO_COMMAND = VSIO_DELETE;                              212
132                     CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                      213
                                      VSIO_FILE_BLOCK,                           214
                                      VSAM_RECORD);                              215
133                     IF (VSIO_RETURN_CODE ª= VSIO_RC_SUCCESS) THEN            216
134                        CALL VSIO_ERROR;                                      217
135                     ELSE                                                     218
135                        DELETE_COUNTER = DELETE_COUNTER + 1;                  219
                                                                                 220
136                     RETURN;                                                  221
                                                                                 222
137                  END DELETE_KS;                                              223
                                                                                 224
138               READ_KS:                                                       225
                     PROCEDURE;                                                  226
                                                                                 227
```

```
                  /*************************************************************** 228
                    CALL ROUTINE TO READ NEXT RECORD FROM VSAM DATASET            228
                   ***************************************************************/ 228
                                                                                   230
139                    VSIO_COMMAND = VSIO_READ;                                   231
140                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                         232
                                     VSIO_FILE_BLOCK,                              233
                                     VSAM_RECORD);                                 234
141                    IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN               235
142                       IF (VSIO_VSAM_FEEDBACK ^= VSIO_FB_RECORD_NOT_FOUND) THEN 236
143                          CALL VSIO_ERROR;                                      237
                                                                                   238
144                    RETURN;                                                     239
                                                                                   240
145                 END READ_KS;                                                   241
                                                                                   242
146                REWRITE_KS:                                                     243
                      PROCEDURE;                                                   244
                                                                                   245
                  /*************************************************************** 246
                    CALL ROUTINE TO REWRITE PREVIOUSLY READ RECORD TO VSAM DATASET 246
                   ***************************************************************/ 246
                                                                                   248
147                    VSIO_COMMAND = VSIO_REWRITE;                                249
148                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                         250
                                     VSIO_FILE_BLOCK,                              251
                                     VSAM_RECORD);                                 252
149                    IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN               253
150                       CALL VSIO_ERROR;                                         254
151                    ELSE                                                        255
151                       CHANGE_COUNTER = CHANGE_COUNTER + 1;                     256
                                                                                   257
152                    RETURN;                                                     258
                                                                                   259
153                 END REWRITE_KS;                                                260
                                                                                   261
154                WRITE_KS:                                                       262
                      PROCEDURE;                                                   263
                                                                                   264
                  /*************************************************************** 265
                    CALL ROUTINE TO WRITE RECORD INTO VSAM DATASET                 265
                   ***************************************************************/ 265
                                                                                   267
155                    VSIO_COMMAND = VSIO_WRITE;                                  268
156                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                         269
                                     VSIO_FILE_BLOCK,                              270
                                     VSAM_RECORD);                                 271
```

```
157                  IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                    272
158                     IF (VSIO_VSAM_FEEDBACK ^= VSIO_FB_DUPLICATE_RECORD) THEN      273
159                        CALL VSIO_ERROR;                                           274
160                     ELSE                                                          275
160                        ;                                                          276
161                  ELSE                                                            277
161                     ADD_COUNTER = ADD_COUNTER + 1;                               278
                                                                                     279
162                  RETURN;                                                         280
                                                                                     281
163                END WRITE_KS;                                                     282
                                                                                     283
164             VSIO_ERROR:                                                         284
                   PROCEDURE;                                                        285
165                   PRINT_AREA = 'VSAMIO ERROR OCCURRED DURING ' ||                286
                                 VSIO_COMMAND;                                       287
166                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                          288
167                   PRINT_AREA = 'VSIO_RETURN_CODE = ' ||                         289
                                 VSIO_RETURN_CODE;                                   290
168                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                          291
169                   PRINT_AREA = 'VSIO_VSAM_RETURN_CODE = ' ||                    292
                                 VSIO_VSAM_RETURN_CODE;                              293
170                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                          294
171                   PRINT_AREA = 'VSIO_VSAM_FUNCTION_CODE = ' ||                  295
                                 VSIO_VSAM_FUNCTION_CODE;                            296
172                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                          297
173                   PRINT_AREA = 'VSIO_VSAM_FEEDBACK_CODE = ' ||                  298
                                 VSIO_VSAM_FEEDBACK_CODE;                            299
174                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                          300
175                   PRINT_AREA = ' ';                                             301
                                                                                     302
176                   RETURN;                                                       303
                                                                                     304
177                END VSIO_ERROR;                                                  305
                                                                                     306
178             DECLARE                                                             307
                   IMAGES FILE INPUT RECORD SEQUENTIAL EXTERNAL                      308
                      ENV(F),                                                        309
                   PRINTR FILE OUTPUT RECORD SEQUENTIAL EXTERNAL                     310
                      ENV(F CTLASA);                                                 311
                                                                                     312
179             DECLARE                                                             313
                   ADD_COUNTER                  FIXED BINARY(15,0) INIT(0),       314
                   CHANGE_COUNTER               FIXED BINARY(15,0) INIT(0),       315
                   COUNTER_EDIT                 PICTURE 'ZZ,ZZZ,ZZ9V',            316
                   DELETE_COUNTER               FIXED BINARY(15,0) INIT(0),       317
                   MORE_RECORDS                 BIT(1),                           318
```

```
                        NO                              BIT(1)    INIT('0'B),
                        YES                             BIT(1)    INIT('1'B);

180                  DECLARE
                        1 MAINT_RECORD,
                           2 MAINT_ACTION               CHAR(1),
                           2 MAINT_SKIP                 CHAR(1),
                           2 MAINT_IMAGE                CHAR(78);

181                  DECLARE
                        MAINT_RECORD_SCALAR             DEFINED MAINT_RECORD
                                                        CHAR(80);

182                  DECLARE
                        1 VSAM_RECORD,
                           2 VSAM_RECORD_KEY            CHAR(10),
                           2 VSAM_RECORD_FIELDS         CHAR(70);

183                  DECLARE
                        VSAM_RECORD_SCALAR              DEFINED VSAM_RECORD
                                                        CHAR(80);

184                  DECLARE
                        1 PRINT_LINE,
                           2 CARRIAGE_CONTROL           CHAR(1)   INIT(' '),
                           2 PRINT_AREA                 CHAR(132);

                  /**********************************************************************

                          VV   VV    SSSSS       A      M     M    IIII      OOOOO
                          VV   VV   SS   SS     AAA     MM   MM     II      OO   OO
                          VV   VV   SS         AA AA    MMM MMM     II      OO   OO
                          VV   VV    SSSSS    AA   AA   MMMMMMM     II      OO   OO
                          VV   VV        SS   AA   AA   MM M MM     II      OO   OO
                           VV VV    SS   SS   AAAAAAA   MM   MM     II      OO   OO
                            VVV     SS   SS   AA   AA   MM   MM     II      OO   OO
                             V       SSSSS    AA   AA   MM   MM    IIII      OOOOO

                   **********************************************************************
                    THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS
                    ROUTINE.

                    THE VSIO_PARAMETER_VALUES SUPPLY THE VALUES USED TO MOVE INTO
                    PARAMETER ENTRIES TO TAILOR THE ROUTINE TO A SPECIFIC DATASET AND
                    TO PROVIDE COMMANDS TO DRIVE THE ROUTINE.
                   **********************************************************************/
```

```
185                     DECLARE                                                  370
                           1 VSIO_PARAMETER_VALUES   STATIC,                     371
                              2 VSIO_OPEN            CHAR(8)   INIT('OPEN    '),  372
                              2 VSIO_CLOSE           CHAR(8)   INIT('CLOSE   '),  373
                              2 VSIO_READ            CHAR(8)   INIT('READ    '),  374
                              2 VSIO_WRITE           CHAR(8)   INIT('WRITE   '),  375
                              2 VSIO_REWRITE         CHAR(8)   INIT('REWRITE '),  376
                              2 VSIO_DELETE          CHAR(8)   INIT('DELETE  '),  377
                              2 VSIO_START_EQUAL     CHAR(8)   INIT('STARTEQ '),  378
                              2 VSIO_START_NOTLESS   CHAR(8)   INIT('STARTGE '),  379
                              2 VSIO_KSDS            CHAR(4)   INIT('KSDS'),      380
                              2 VSIO_ESDS            CHAR(4)   INIT('ESDS'),      381
                              2 VSIO_RRDS            CHAR(4)   INIT('RRDS'),      382
                              2 VSIO_SEQUENTIAL      CHAR(10)  INIT('SEQUENTIAL'),383
                              2 VSIO_DIRECT          CHAR(10)  INIT('DIRECT    '),384
                              2 VSIO_DYNAMIC         CHAR(10)  INIT('DYNAMIC   '),385
                              2 VSIO_INPUT           CHAR(6)   INIT('INPUT '),    386
                              2 VSIO_OUTPUT          CHAR(6)   INIT('OUTPUT'),    387
                              2 VSIO_INPUT_OUTPUT    CHAR(6)   INIT('UPDATE'),    388
                              2 (VSIO_RC_SUCCESS                INIT(0),          389
                                 VSIO_RC_LOGIC_ERROR            INIT(8),          390
                                 VSIO_RC_END_OF_FILE            INIT(9999),       391
                                 VSIO_RC_UNKNOWN_COMMAND        INIT(20),         392
                                 VSIO_RC_DATASET_ALREADY_OPEN   INIT(21),         393
                                 VSIO_RC_DATASET_NOT_OPEN       INIT(22),         394
                                 VSIO_RC_ORGANIZATION_UNKNOWN   INIT(23),         395
                                 VSIO_RC_ACCESS_UNKNOWN         INIT(24),         396
                                 VSIO_RC_ORG_ACCESS_MISMATCH    INIT(25),         397
                                 VSIO_RC_MODE_UNKNOWN           INIT(26),         398
                                 VSIO_RC_MODE_UNSUPPORTED       INIT(27),         399
                                 VSIO_RC_DDNAME_BLANK           INIT(28))         400
                                                   FIXED BINARY(15,0),            401
                             2 (VSIO_FB_DUPLICATE_RECORD        INIT(8),          402
                                 VSIO_FB_KEY_SEQUENCE           INIT(12),         403
                                 VSIO_FB_RECORD_NOT_FOUND       INIT(16),         404
                                 VSIO_FB_NO_MORE_SPACE          INIT(28),         405
                                 VSIO_FB_READ_WITHOUT_START     INIT(88))         406
                                                   FIXED BINARY(15,0),            407
                        /*************************************************************  408
                          THE VSIO_PARAMETER_BLOCK IS THE COMMUNICATION INTERFACE TO THE  408
                          THE ROUTINE.                                                   408
                         *************************************************************/  408
                                                                                  411
                           1 VSIO_PARAMETER_BLOCK    STATIC,                     413
                              2 VSIO_COMMAND         CHAR(8)   INIT(' '),        414
                              2 (VSIO_RETURN_CODE,                               415
                                 VSIO_VSAM_RC,                                   416
```

```
                    VSIO_VSAM_FUNCTION,                                          417
                    VSIO_VSAM_FEEDBACK) FIXED BINARY(15,0) INIT(0);              418
                                                                                 419
             /*********************************************************************  420
                              END OF VSAMIO COPY BOOK                             420
              *********************************************************************/  420
             /*********************************************************************  423
                                                                                    423
                VV   VV   SSSSS      A     M     M  IIII   OOOOO   FFFFFFF  BBBBBB    423
                VV   VV  SS   SS    AAA    MM   MM   II   OO   OO  FF       BB   BB   423
                VV   VV  SS        AA AA   MMM MMM   II   OO   OO  FF       BB   BB   423
                VV   VV   SSSSS   AA   AA  MMMMMMM   II   OO   OO  FFFFF    BBBBBB    423
                VV   VV       SS  AA   AA  MM M MM   II   OO   OO  FF       BB   BB   423
                 VV VV   SS   SS  AAAAAAA  MM   MM   II   OO   OO  FF       BB   BB   423
                  VVV    SS   SS  AA   AA  MM   MM   II   OO   OO  FF       BB   BB   423
                   V      SSSSS   AA   AA  MM   MM  IIII   OOOOO   FF       BBBBBB    423
                                                                                    423
              *********************************************************************  423
                THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS   423
                ROUTINE, AND ARE USED TO COMMUNICATE CHARACTERISTICS FOR A SINGLE     423
                VSAM DATASET.                                                         423
                                                                                      423
                WITH THE 2 EXCEPTIONS FOR RECORD LENGTH (TO ACCOMODATE VARIABLE       423
                LENGTH RECORDS) AND RELATIVE RECORD (TO ACCOMODATE RELATIVE RECORD     423
                DATASETS), THESE DATA NAMES MUST BE POPULATED PRIOR TO CALLING THE    423
                ROUTINE TO OPEN THE DATASET AND MUST NOT THEN BE CHANGED UNTIL THE    423
                DATASET HAS BEEN CLOSED.                                              423
                 *********************************************************************/  423
                                                                                      444
186                  DECLARE                                                          446
                        1 VSIO_FILE_BLOCK            STATIC,                         447
                           2 VSFB_DDNAME             CHAR(8)   INIT(' '),            448
                           2 VSFB_ORGANIZATION       CHAR(4)   INIT(' '),            449
                           2 VSFB_ACCESS             CHAR(10)  INIT(' '),            450
                           2 VSFB_MODE               CHAR(6)   INIT(' '),            451
                           2 (VSFB_RECORD_LENGTH,                                    452
                              VSFB_KEY_POSITION,                                     453
                              VSFB_KEY_LENGTH)       FIXED BINARY(15,0) INIT(0),     454
                           2 VSFB_FILE_STATUS        CHAR(1)   INIT('C'),            455
                           2 VSFB_RESERVED           CHAR(161);                      456
                                                                                     457
                    /*********************************************************************  458
                                       END OF VSAMIOFB COPY BOOK                     458
                     *********************************************************************/  458
                                                                                      348
187                    END KSDSRND;                                                   349
```

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
179	*********	ADD_COUNTER	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 43,161,161
184		CARRIAGE_CONTROL	IN PRINT_LINE,AUTOMATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER
179	*********	CHANGE_COUNTER	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 46,151,151
96		COMPLETE_CHANGE	ENTRY,DECIMAL,FLOAT(SINGLE) 89
119		COMPLETE_DELETE	ENTRY,DECIMAL,FLOAT(SINGLE) 112
179		COUNTER_EDIT	AUTOMATIC,UNALIGNED,DECIMAL,PICTURE(ZZ,ZZZ,ZZ9V) 43,44,46,47,49,50,62,63
179	*********	DELETE_COUNTER	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 49,135,135
130		DELETE_KS	ENTRY,DECIMAL,FLOAT(SINGLE) 122
178		IMAGES	FILE,EXTERNAL,INPUT,RECORD,SEQUENTIAL,ENVIRONMENT(F) 9,10,34,38
1	*********	KSDSRND	ENTRY,BINARY,FIXED(15,0)
180		MAINT_ACTION	IN MAINT_RECORD,AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 55,57,59
180		MAINT_IMAGE	IN MAINT_RECORD,AUTOMATIC,UNALIGNED,STRING(78),CHARACTER 69,86,99,109
180		MAINT_RECORD	AUTOMATIC,STRUCTURE 34
181		MAINT_RECORD_SCALAR	AUTOMATIC,DEFINED,UNALIGNED,STRING(80),CHARACTER 63
180		MAINT_SKIP	IN MAINT_RECORD,AUTOMATIC,UNALIGNED,STRING(1),CHARACTER

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
179	MORE_RECORDS	AUTOMATIC,UNALIGNED,STRING(1),BIT 11,18,33,35
179	NO	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 11
184	PRINT_AREA	IN PRINT_LINE,AUTOMATIC,UNALIGNED,STRING(132),CHARACTER 12,14,16,44,47,50,63,70,75,80,91,97,103,114,120,125,165,167,169,171 173,175
184	PRINT_LINE	AUTOMATIC,STRUCTURE 13,15,17,45,48,51,64,71,76,81,92,98,104,115,121,126,166,168,170,172 174
178	PRINTR	FILE,EXTERNAL,OUTPUT,RECORD,SEQUENTIAL,ENVIRONMENT(F CTLASA) 9,13,15,17,45,48,51,64,71,76,81,92,98,104,115,121,126,166,168,170 172,174
68	PROCESS_ADD	ENTRY,DECIMAL,FLOAT(SINGLE) 56
85	PROCESS_CHANGE	ENTRY,DECIMAL,FLOAT(SINGLE) 58
108	PROCESS_DELETE	ENTRY,DECIMAL,FLOAT(SINGLE) 60
53	PROCESS_MAINT	ENTRY,DECIMAL,FLOAT(SINGLE) 36
138	READ_KS	ENTRY,DECIMAL,FLOAT(SINGLE) 87,110
	RECORD_COUNTER	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 54,54,62
146	REWRITE_KS	ENTRY,DECIMAL,FLOAT(SINGLE) 100
	SYSPRINT	FILE,EXTERNAL 5,6,7
182	VSAM_RECORD	AUTOMATIC,STRUCTURE 27,40,132,140,148,156

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
182		VSAM_RECORD_FIELDS	IN VSAM_RECORD,AUTOMATIC,UNALIGNED,STRING(70),CHARACTER
182		VSAM_RECORD_KEY	IN VSAM_RECORD,AUTOMATIC,UNALIGNED,STRING(10),CHARACTER 70,91,97,114,120
183		VSAM_RECORD_SCALAR	AUTOMATIC,DEFINED,UNALIGNED,STRING(80),CHARACTER 69,70,86,97,99,103,109,120
		VSAMIOP	EXTERNAL,ENTRY,DECIMAL,FLOAT(SINGLE) 27,40,132,140,148,156
186		VSFB_ACCESS	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(10),CHARACTER 21
186		VSFB_DDNAME	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(8),CHARACTER 19
186		VSFB_FILE_STATUS	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER
186	*********	VSFB_KEY_LENGTH	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 25
186	*********	VSFB_KEY_POSITION	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 24
186		VSFB_MODE	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(6),CHARACTER 22
186		VSFB_ORGANIZATION	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(4),CHARACTER 20
186	*********	VSFB_RECORD_LENGTH	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 23
186		VSFB_RESERVED	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,STRING(161),CHARACTER
185		VSIO_CLOSE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 39
185		VSIO_COMMAND	IN VSIO_PARAMETER_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 26,39,131,139,147,155,165

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
185		VSIO_DELETE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 131
185		VSIO_DIRECT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER 21
185		VSIO_DYNAMIC	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER
164		VSIO_ERROR	ENTRY,DECIMAL,FLOAT(SINGLE) 30,42,134,143,150,159
185		VSIO_ESDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER
185	*********	VSIO_FB_DUPLICATE_RECORD	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 78,158
185	*********	VSIO_FB_KEY_SEQUENCE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
185	*********	VSIO_FB_NO_MORE_SPACE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
185	*********	VSIO_FB_READ_WITHOUT_START	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
185	*********	VSIO_FB_RECORD_NOT_FOUND	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 142
186		VSIO_FILE_BLOCK	STATIC,STRUCTURE 27,40,132,140,148,156
185		VSIO_INPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER
185		VSIO_INPUT_OUTPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER 22
185		VSIO_KSDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4),

```
DCL NO.          IDENTIFIER                        ATTRIBUTES AND REFERENCES

                                                   CHARACTER
                                                   20

185              VSIO_OPEN                         IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                   CHARACTER
                                                   26

185              VSIO_OUTPUT                       IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6),
                                                   CHARACTER

185              VSIO_PARAMETER_BLOCK              STATIC,STRUCTURE
                                                   27,40,132,140,148,156

185              VSIO_PARAMETER_VALUES             STATIC,STRUCTURE

185    ********* VSIO_RC_ACCESS_UNKNOWN            IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)

185    ********* VSIO_RC_DATASET_ALREADY_OPEN       IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)

185    ********* VSIO_RC_DATASET_NOT_OPEN           IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)

185    ********* VSIO_RC_DDNAME_BLANK               IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)

185    ********* VSIO_RC_END_OF_FILE                IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)

185    ********* VSIO_RC_LOGIC_ERROR                IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)

185    ********* VSIO_RC_MODE_UNKNOWN               IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)

185    ********* VSIO_RC_MODE_UNSUPPORTED           IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)

185    ********* VSIO_RC_ORG_ACCESS_MISMATCH        IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)

185    ********* VSIO_RC_ORGANIZATION_UNKNOWN       IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)
```

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
185	*********	VSIO_RC_SUCCESS	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 28,41,73,88,101,111,123,133,141,149,157
185	*********	VSIO_RC_UNKNOWN_COMMAND	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
185		VSIO_READ	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 139
185	*********	VSIO_RETURN_CODE	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 28,41,73,88,101,111,123,133,141,149,157,167
185		VSIO_REWRITE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 147
185		VSIO_RRDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER
185		VSIO_SEQUENTIAL	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER
185		VSIO_START_EQUAL	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
185		VSIO_START_NOTLESS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
185	*********	VSIO_VSAM_FEEDBACK	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 78,142,158
		VSIO_VSAM_FEEDBACK_CODE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 173
185	*********	VSIO_VSAM_FUNCTION	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
		VSIO_VSAM_FUNCTION_CODE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 171
185	*********	VSIO_VSAM_RC	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		(15,0)
	VSIO_VSAM_RETURN_CODE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 169
185	VSIO_WRITE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 155
154	WRITE_KS	ENTRY,DECIMAL,FLOAT(SINGLE) 72
179	YES	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 18

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
180	MAINT_RECORD	80
184	PRINT_LINE	133
182	VSAM_RECORD	80
186	VSIO_FILE_BLOCK	196
185	VSIO_PARAMETER_BLOCK	16
185	VSIO_PARAMETER_VALUES	158

STORAGE REQUIREMENTS.

THE STORAGE AREA FOR THE PROCEDURE LABELLED KSDSRND IS 656 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 3 IS 184 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 10 IS 176 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED PROCESS_MAINT IS 276 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED PROCESS_ADD IS 276 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED PROCESS_CHANGE IS 244 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED COMPLETE_CHANGE IS 276 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED PROCESS_DELETE IS 244 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED COMPLETE_DELETE IS 276 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED DELETE_KS IS 176 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED READ_KS IS 176 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED REWRITE_KS IS 176 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED WRITE_KS IS 176 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED VSIO_ERROR IS 256 BYTES LONG.

THE PROGRAM CSECT IS NAMED KSDSRND AND IS 3858 BYTES LONG.

THE STATIC CSECT IS NAMED KSDSRNDA AND IS 8848 BYTES LONG.

STATISTICS MACRO RECORDS = 460,SOURCE RECORDS = 465,PROG TEXT STMNTS = 187,OBJECT BYTES = 3858

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN ON UNIT

OFFSET (HEX) 0000 0050 005C 007A 0094 00B2
STATEMENT NO 3 4 5 6 7 8

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN ON UNIT

OFFSET (HEX) 0000 0048 0052
STATEMENT NO 11 11

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE PROCESS_MAINT

OFFSET (HEX) 0000 0048 0058 0062 0070 007E 008C 009A 00A8 00A8 00C6 00EC 0104 0104 010A
STATEMENT NO 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE PROCESS_ADD

OFFSET (HEX) 0000 0044 0062 0090 00A8 00B2 00BE 00BE 00CE 00E6 00EA 00F6 00F6 0106 011E 011E 0124
STATEMENT NO 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE PROCESS_CHANGE

OFFSET (HEX) 0000 0048 0066 0070 007C 008A 008A 00AA 00C2 00C2 00C8
STATEMENT NO 85 86 87 88 89 90 91 92 93 94 95

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE COMPLETE_CHANGE

OFFSET (HEX) 0000 0048 007E 0096 00AC 00B6 00C2 00C2 00EC 0104 0104 010A
STATEMENT NO 96 97 98 99 100 101 102 103 104 105 106 107

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE PROCESS_DELETE

OFFSET (HEX) 0000 0048 0066 0070 007C 008A 008A 00AA 00C2 00C2 00C8
STATEMENT NO 108 109 110 111 112 113 114 115 116 117 118

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE COMPLETE_DELETE

OFFSET (HEX) 0000 0048 007E 0096 00A0 00AC 00AC 00BC 00D4 00D4 00DA
STATEMENT NO 119 120 121 122 123 124 125 126 127 128 129

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE DELETE_KS

OFFSET (HEX)	0000	0038	003E	005E	006A	0078	0088	008E
STATEMENT NO	130	131	132	133	134	135	136	137

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE READ_KS

OFFSET (HEX)	0000	0034	003A	005A	0066	0072	007C	0082
STATEMENT NO	138	139	140	141	142	143	144	145

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE REWRITE_KS

OFFSET (HEX)	0000	0038	003E	005E	006A	0078	0088	008E
STATEMENT NO	146	147	148	149	150	151	152	153

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE WRITE_KS

OFFSET (HEX)	0000	0038	003E	005E	006A	0076	0084	0088	0098	009E
STATEMENT NO	154	155	156	157	158	159	160	161	162	163

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE VSIO_ERROR

OFFSET (HEX)	0000	0038	0052	006A	00B6	00CE	0104	011C	014E	0166	0198	01B0	01BC	01C2
STATEMENT NO	164	165	166	167	168	169	170	171	172	173	174	175	176	177

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE KSDSRND

OFFSET (HEX)	0000	012C	013A	0144	0152	015E	0176	0182	019A	01A6	01BE	01C4	01CA	01D0	01D6	01DC	01E2	01E8	01EE	01F4	0210
STATEMENT NO	1	2	9	10	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
OFFSET (HEX)	021C	021C	0226	022C	022C	0234	024C	0254	025E	0262	026C	0272	028E	029A	02A4	02BE	02DA	02F2	030C	0328	0340
STATEMENT NO	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
OFFSET (HEX)	035A	0376	038E	0394																	
STATEMENT NO	50	51	52	187																	

COMPILER DIAGNOSTICS.

WARNINGS.

IEM0227I NO FILE/STRING OPTION SPECIFIED IN ONE OR MORE GET/PUT STATEMENTS. SYSIN/SYSPRINT HAS BEEN ASSUMED IN EACH CASE.

IEM0764I ONE OR MORE FIXED BINARY ITEMS OF PRECISION 15 OR LESS HAVE BEEN GIVEN HALFWORD STORAGE. THEY ARE FLAGGED '*********' IN THE XREF/ATR LIST.

IEM1790I DATA CONVERSIONS WILL BE DONE BY SUBROUTINE CALL IN THE FOLLOWING STATEMENTS 62, 169, 171, 173.

END OF DIAGNOSTICS.

AUXILIARY STORAGE WILL NOT BE USED FOR DICTIONARY WHEN SIZE = 138K

COMPILE TIME .00 MINS

ELAPSED TIME .00 MINS

```
 F64-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED   NONE
          DEFAULT OPTION(S) USED -  SIZE=(231424,55296)
****GO        DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET
AUTHORIZATION CODE IS         0.
```

```
KSDSRAND: READ/UPDATE KSDS DIRECT
---------------------------------

KEY: 9532957501  DATA: 9532957501 JOHN J GLASSMAN      7663 SEASIDE AVENUE        NEWPORT BEACH   CA     ADDING
                                                                                                         ADDED OK
KEY: 1964475502  DATA: 1964475502 CAITLIN V BROCKTON   9540 PARKER COURT          KANSAS CITY     MO     ADDING
                                                                                                         ADDED OK
KEY: 9441505503  DATA: 9441505503 SUE P MOORE          640 JACKSON STREET         LOS ANGELES     CA     ADDING
                                                                                                         ADDED OK
KEY: 0045557001  DATA: 0045557001 LARRY E BENSON      4778 DESERT STREET         LA HABRA        CA     ADDING
                                                                                                         DUPLICATE RECORD
READING FOR UPDATE: 5500563505  *** NOT FOUND ***
READING FOR UPDATE: 9966129028  *** NOT FOUND ***
KEY: 9441505503  DATA: 9441505503 SUE P MOORE          640 JACKSON STREET         LOS ANGELES     CA     BEFORE
                       9441505503 CLARA B ALEXANDER    8427 PECAN VALLEY STREET   BOISE           ID     AFTER
READING FOR DELETE: 3849060508  *** NOT FOUND ***
KEY: 3129003066  DATA: 3129003066 BILL W BECK          1798 SEABREEZE AVENUE      BOSTON          MA     BEFORE
                                                                                                         DELETED

        3 RECORDS ADDED
        1 RECORDS CHANGED
        1 RECORDS DELETED
```