J E S 2 J O B L O G

```
18.12.10 JOB  152  IEF677I WARNING MESSAGE(S) FOR JOB VSTESTK4 ISSUED
18.12.10 JOB  152  $HASP373 VSTESTK4 STARTED - INIT  1 - CLASS A - SYS HMVS
18.12.10 JOB  152  IEF403I VSTESTK4 - STARTED - TIME=18.12.10
18.12.10 JOB  152  CCI001C PL1L    /IEMAA   /00:00:00.18/        /00004/SYS     /VSTESTK4
18.12.10 JOB  152  CCI001C LKED    /IEWL    /00:00:00.05/        /00000/SYS     /VSTESTK4
18.12.10 JOB  152  CCI001C GO      /PGM=*.DD/00:00:00.01/        /00000/SYS     /VSTESTK4
18.12.10 JOB  152  IEF404I VSTESTK4 - ENDED - TIME=18.12.10
18.12.10 JOB  152  $HASP395 VSTESTK4 ENDED

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

 07 JUL 20 JOB EXECUTION DATE

       22 CARDS READ

    1,222 SYSOUT PRINT RECORDS

        0 SYSOUT PUNCH RECORDS

     0.00 MINUTES EXECUTION TIME
```

```
 1      //VSTESTK4 JOB (SYS),'VSAMIOP IVP KSDSUPDT',CLASS=A,MSGCLASS=X,         JOB  152
        //             REGION=4096K
        ***
        ********************************************************************
        *** PL/1 MODULE: KSDSUPDT     VSAM DATASET: VSTESTKS.CLUSTER (KSDS)
        ***
        *** SEQUENTIALLY READ RECORDS AND UPDATE OR DELETE SELECTED RECORDS
        ********************************************************************
        ***
 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(KSDSUPDT),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
```

```
 STMT NO. MESSAGE
-
   19     IEF686I DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
IEF236I ALLOC. FOR VSTESTK4 PL1L PL1F
IEF237I 253  ALLOCATED TO STEPLIB
IEF237I 253  ALLOCATED TO SYS00342
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 VSTESTK4 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.JOB00152.SO0101                        SYSOUT
IEF285I   SYS20189.T181210.RA000.VSTESTK4.LOADSET      PASSED        *------232
IEF285I   VOL SER NOS= MVS380.
IEF285I   SYS20189.T181210.RA000.VSTESTK4.SYSUT3       DELETED       *------329
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T181210.RA000.VSTESTK4.SYSUT1       DELETED       *--------0
IEF285I   VOL SER NOS= MVS370.
IEF285I   SYSC.VSAMIOP.SOURCE                         KEPT          *--------3
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 00.18SEC SRB    0MIN 00.04SEC VIRT  4096K SYS   212K
**** JOBCARD READ 20189 18:12:10 ************************************************************************************
*                                      PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                              *
*  STEP NAME  PL1L        USER CORE        4096K   TAPES USED/IO 000/000000000   START   TIME  18:12:10    TCB TIME  00:00:00.18  *
*  PGM  NAME  IEMAA       SYSTEM CORE       212K   DISKS USED/IO 004/000000591   STOP    TIME  18:12:10    SRB TIME  00:00:00.04  *
*   COND CODE    0004     PRIVATE AREA SZ   4096K       ALLOC TIME  18:12:10      ELAPSED TIME                PGM LOAD  18:12:10   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004          3197   00:00:00.25            0             0           0             0              0             0            0  *
************************************************************************************************************************
*  CPU $ (     0.06) + EXCP $ (     0.79) + MEMORY $ (     2.10) = TOTAL $ (     2.95)                                  *
************************************************************************************************************************
IEF236I ALLOC. FOR VSTESTK4 LKED PL1F
IEF237I 253  ALLOCATED TO SYSLIB
IEF237I 253  ALLOCATED TO
IEF237I 253  ALLOCATED TO SYS00344
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 VSTESTK4 LKED PL1F - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYSC.PL1LIB                                 KEPT          *------106
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.T181210.RA000.VSTESTK4.GOSET        PASSED        *-------57
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T181210.RA000.VSTESTK4.SYSUT1       DELETED       *--------0
IEF285I   VOL SER NOS= MVS370.
IEF285I   JES2.JOB00152.SO0102                        SYSOUT
```

```
IEF285I   SYS20189.T181210.RA000.VSTESTK4.LOADSET      DELETED       *------233
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   208K
********************************************************************************************************************
*                                        PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                             *
*  STEP NAME  LKED        USER CORE          260K   TAPES USED/IO 000/000000000   START   TIME  18:12:10    TCB TIME  00:00:00.05  *
*  PGM  NAME  IEWL        SYSTEM CORE        208K   DISKS USED/IO 004/000000396   STOP    TIME  18:12:10    SRB TIME  00:00:00.01  *
*   COND CODE    0000     PRIVATE AREA SZ   4096K       ALLOC TIME  18:12:10      ELAPSED TIME                PGM LOAD  18:12:10   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004          2024   00:00:00.06            0             0         0              0              0             0  *
********************************************************************************************************************
*  CPU $ (     0.01) + EXCP $ (     0.53) + MEMORY $ (     0.03) = TOTAL $ (     0.57)                                *
********************************************************************************************************************
IEF236I ALLOC. FOR VSTESTK4 GO PL1F
IEF237I 251  ALLOCATED TO PGM=*.DD
IEF237I 253  ALLOCATED TO STEPLIB
IEF237I 253  ALLOCATED TO
IEF237I 253  ALLOCATED TO SYS00346
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 SYS00348
IEF142I VSTESTK4 GO PL1F - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS20189.T181210.RA000.VSTESTK4.GOSET        KEPT          *--------0
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYSC.PL1LIB                                  KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   UCSYSCPK                                     KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   JES2.JOB00152.SO0103                         SYSOUT
IEF285I   JES2.JOB00152.SO0104                         SYSOUT
IEF285I   JES2.JOB00152.SO0105                         SYSOUT
IEF285I   JES2.JOB00152.SO0106                         SYSOUT
IEF285I   PUB001.VSTESTKS.CLUSTER                      KEPT          *--------3
IEF285I   VOL SER NOS= PUB001.
IEF285I   UCPUB001                                     KEPT          *--------0
IEF285I   VOL SER NOS= PUB001.
IEF373I STEP /GO      / START 20189.1812
IEF374I STEP /GO      / STOP  20189.1812 CPU    0MIN 00.01SEC SRB    0MIN 00.00SEC VIRT   108K SYS   220K
********************************************************************************************************************
*                                        PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                             *
*  STEP NAME  GO          USER CORE          108K   TAPES USED/IO 000/000000000   START   TIME  18:12:10    TCB TIME  00:00:00.01  *
*  PGM  NAME  PGM=*.DD    SYSTEM CORE        220K   DISKS USED/IO 003/000000003   STOP    TIME  18:12:10    SRB TIME  00:00:00.00  *
*   COND CODE    0000     PRIVATE AREA SZ   4096K       ALLOC TIME  18:12:10      ELAPSED TIME                PGM LOAD  18:12:10   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004            42   00:00:00.02            0             0         0              0              0             0  *
********************************************************************************************************************
*  CPU $ (     0.00) + EXCP $ (     0.00) + MEMORY $ (     0.00) = TOTAL $ (     0.00)                                *
********************************************************************************************************************
IEF237I 251  ALLOCATED TO SYS00001
IEF285I   SYS20189.T181210.RA000.VSTESTK4.R0000001     KEPT          *--------0
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T181210.RA000.VSTESTK4.GOSET        DELETED
IEF285I   VOL SER NOS= WORK00.
IEF375I  JOB /VSTESTK4/ START 20189.1812
IEF376I  JOB /VSTESTK4/ STOP  20189.1812 CPU    0MIN 00.24SEC SRB    0MIN 00.05SEC
```

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    /*****************************************************************14950000
  2                                                                      14960000
  3      KSDSUPDT - TESTS THE VSAMIO ROUTINE BY READING RECORDS FROM A KSDS   14970000
  4                 CLUSTER SEQUENTIALLY AND THEN REWRITING SOME OF THE       14980000
  5                 RECORDS WITH UPDATED CONTENTS.                           14990000
  6                                                                      15000000
  7     *****************************************************************/15010000
  8    KSDSUPD:                                                          15020000
  9       PROCEDURE OPTIONS(MAIN);                                       15030000
 10                                                                      15040000
 11          ON ERROR                                                    15050000
 12             BEGIN;                                                   15060000
 13                ON ERROR SYSTEM;                                      15070000
 14                PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*');     15080000
 15                PUT SKIP DATA;                                        15090000
 16                PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*');     15100000
 17             END;                                                     15110000
 18                                                                      15120000
 19          OPEN                                                        15130000
 20             FILE(PRINTR) LINESIZE(133);                              15140000
 21                                                                      15150000
 22          PRINT_AREA = 'KSDSUPDT: READ/REWRITE KSDS SEQUENTIALLY';    15160000
 23          WRITE FILE(PRINTR) FROM(PRINT_LINE);                        15170000
 24          PRINT_AREA = '----------------------------------------';    15180000
 25          WRITE FILE(PRINTR) FROM(PRINT_LINE);                        15190000
 26          PRINT_AREA = ' ';                                           15200000
 27          WRITE FILE(PRINTR) FROM(PRINT_LINE);                        15210000
 28                                                                      15220000
 29          MORE_RECORDS = YES;                                         15230000
 30                                                                      15240000
 31     /*****************************************************************15250000
 32       ESTABLISH PARAMETERS OF VSAM DATASET AND CALL ROUTINE TO OPEN IT     15260000
 33      *****************************************************************/15270000
 34          VSFB_DDNAME = 'KSDSF01';                                    15280000
 35          VSFB_ORGANIZATION = VSIO_KSDS;                              15290000
 36          VSFB_ACCESS = VSIO_SEQUENTIAL;                              15300000
 37          VSFB_MODE = VSIO_INPUT_OUTPUT;                              15310000
 38          VSFB_RECORD_LENGTH = 80;                                    15320000
 39          VSFB_KEY_POSITION = 0;                                      15330000
 40          VSFB_KEY_LENGTH = 10;                                       15340000
 41          VSIO_COMMAND = VSIO_OPEN;                                   15350000
 42          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                         15360000
 43                        VSIO_FILE_BLOCK,                              15370000
 44                        RECORD_IMAGE);                                15380000
```

MACRO SOURCE2 LISTING

```
 45          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                   15390000
 46             DO;                                                          15400000
 47                CALL VSIO_ERROR;                                          15410000
 48                RETURN;                                                   15420000
 49             END;                                                         15430000
 50                                                                          15440000
 51          DO WHILE(MORE_RECORDS);                                         15450000
 52             CALL READ_KS;                                                15460000
 53             IF (MORE_RECORDS) THEN                                       15470000
 54                IF (RECORD_CITY = 'SAN ANTONIO    ' |                     15480000
 55                    RECORD_CITY = 'DETROIT        ') THEN                 15490000
 56                   CALL UPDATE_RECORD;                                    15500000
 57          END;                                                            15510000
 58                                                                          15520000
 59    /*****************************************************************15530000
 60      CALL ROUTINE TO CLOSE VSAM DATASET                                  15540000
 61     *****************************************************************/15550000
 62          VSIO_COMMAND = VSIO_CLOSE;                                      15560000
 63          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                             15570000
 64                        VSIO_FILE_BLOCK,                                  15580000
 65                        RECORD_IMAGE);                                    15590000
 66          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                   15600000
 67             CALL VSIO_ERROR;                                             15610000
 68                                                                          15620000
 69          RETURN;                                                         15630000
 70                                                                          15640000
 71    UPDATE_RECORD:                                                        15650000
 72       PROCEDURE;                                                         15660000
 73                                                                          15670000
 74          COUNTER_EDIT = RECORD_COUNTER;                                  15680000
 75          PRINT_AREA = COUNTER_EDIT || ':  KEY: ' || RECORD_KEY ||        15690000
 76                       '  DATA: ' || RECORD_IMAGE_SCALAR ||'  BEFORE';    15700000
 77          WRITE FILE(PRINTR) FROM(PRINT_LINE);                            15710000
 78                                                                          15720000
 79          IF (RECORD_CITY = 'SAN ANTONIO    ') THEN                       15730000
 80             DO;                                                          15740000
 81                RECORD_CITY = 'AUSTIN';                                   15750000
 82                CALL REWRITE_KS;                                          15760000
 83                IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN              15770000
 84                   DO;                                                    15780000
 85                      PRINT_AREA = (36)' ' ||                             15790000
 86                                   RECORD_IMAGE_SCALAR || '  AFTER';      15800000
 87                      WRITE FILE(PRINTR) FROM(PRINT_LINE);                15810000
 88                   END;                                                   15820000
 89             END;                                                         15830000
```

MACRO SOURCE2 LISTING

```
 90          ELSE                                                    15840000
 91             DO;                                                  15850000
 92                CALL DELETE_KS;                                   15860000
 93                IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN      15870000
 94                   DO;                                            15880000
 95                      PRINT_AREA = (116)' ' || '  DELETED';       15890000
 96                      WRITE FILE(PRINTR) FROM(PRINT_LINE);        15900000
 97                   END;                                           15910000
 98             END;                                                 15920000
 99                                                                  15930000
100          RETURN;                                                 15940000
101                                                                  15950000
102       END UPDATE_RECORD;                                         15960000
103                                                                  15970000
104    DELETE_KS:                                                    15980000
105       PROCEDURE;                                                 15990000
106                                                                  16000000
107    /*****************************************************************16010000
108      CALL ROUTINE TO DELETE LAST RECORD READ FROM VSAM DATASET   16020000
109     *****************************************************************/16030000
110          VSIO_COMMAND = VSIO_DELETE;                             16040000
111          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                     16050000
112                        VSIO_FILE_BLOCK,                          16060000
113                        RECORD_IMAGE);                            16070000
114          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN           16080000
115             CALL VSIO_ERROR;                                     16090000
116                                                                  16100000
117          RETURN;                                                 16110000
118                                                                  16120000
119       END DELETE_KS;                                             16130000
120                                                                  16140000
121    READ_KS:                                                      16150000
122       PROCEDURE;                                                 16160000
123                                                                  16170000
124    /*****************************************************************16180000
125      CALL ROUTINE TO READ NEXT RECORD FROM VSAM DATASET          16190000
126     *****************************************************************/16200000
127          VSIO_COMMAND = VSIO_READ;                               16210000
128          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                     16220000
129                        VSIO_FILE_BLOCK,                          16230000
130                        RECORD_IMAGE);                            16240000
131          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN           16250000
132             IF (VSIO_RETURN_CODE = VSIO_RC_END_OF_FILE) THEN     16260000
133                MORE_RECORDS = NO;                                16270000
134             ELSE                                                 16280000
```

MACRO SOURCE2 LISTING

```
135                CALL VSIO_ERROR;                                    16290000
136           ELSE                                                     16300000
137              RECORD_COUNTER = RECORD_COUNTER + 1;                  16310000
138                                                                    16320000
139           RETURN;                                                  16330000
140                                                                    16340000
141         END READ_KS;                                               16350000
142                                                                    16360000
143      REWRITE_KS:                                                   16370000
144         PROCEDURE;                                                 16380000
145                                                                    16390000
146      /*****************************************************************16400000
147        CALL ROUTINE TO REWRITE PREVIOUSLY READ RECORD TO VSAM DATASET  16410000
148       *****************************************************************/16420000
149           VSIO_COMMAND = VSIO_REWRITE;                             16430000
150           CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                      16440000
151                         VSIO_FILE_BLOCK,                           16450000
152                         RECORD_IMAGE);                             16460000
153           IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN            16470000
154              CALL VSIO_ERROR;                                      16480000
155                                                                    16490000
156           RETURN;                                                  16500000
157                                                                    16510000
158         END REWRITE_KS;                                            16520000
159                                                                    16530000
160      VSIO_ERROR:                                                   16540000
161         PROCEDURE;                                                 16550000
162            PRINT_AREA = 'VSAMIO ERROR OCCURRED DURING ' ||         16560000
163                         VSIO_COMMAND;                              16570000
164            WRITE FILE(PRINTR) FROM(PRINT_LINE);                    16580000
165            PRINT_AREA = 'VSIO_RETURN_CODE = ' ||                   16590000
166                         VSIO_RETURN_CODE;                          16600000
167            WRITE FILE(PRINTR) FROM(PRINT_LINE);                    16610000
168            PRINT_AREA = 'VSIO_VSAM_RETURN_CODE = ' ||              16620000
169                         VSIO_VSAM_RETURN_CODE;                     16630000
170            WRITE FILE(PRINTR) FROM(PRINT_LINE);                    16640000
171            PRINT_AREA = 'VSIO_VSAM_FUNCTION_CODE = ' ||            16650000
172                         VSIO_VSAM_FUNCTION_CODE;                   16660000
173            WRITE FILE(PRINTR) FROM(PRINT_LINE);                    16670000
174            PRINT_AREA = 'VSIO_VSAM_FEEDBACK_CODE = ' ||            16680000
175                         VSIO_VSAM_FEEDBACK_CODE;                   16690000
176            WRITE FILE(PRINTR) FROM(PRINT_LINE);                    16700000
177            PRINT_AREA = ' ';                                       16710000
178                                                                    16720000
179            RETURN;                                                 16730000
```

MACRO SOURCE2 LISTING

```
180                                                                   16740000
181       END VSIO_ERROR;                                             16750000
182                                                                   16760000
183       DECLARE                                                     16770000
184          PRINTR FILE OUTPUT RECORD SEQUENTIAL EXTERNAL            16780000
185             ENV(F CTLASA);                                        16790000
186                                                                   16800000
187       DECLARE                                                     16810000
188          COUNTER_EDIT                PICTURE 'ZZ,ZZZ,ZZ9V',     16820000
189          MORE_RECORDS                BIT(1),                     16830000
190          NO                          BIT(1)    INIT('0'B),       16840000
191          RECORD_COUNTER              FIXED BINARY(15,0),         16850000
192          YES                         BIT(1)    INIT('1'B);       16860000
193                                                                   16870000
194       DECLARE                                                     16880000
195          1 RECORD_IMAGE,                                          16890000
196             2 RECORD_KEY             CHAR(10),                   16900000
197             2 RECORD_UNDEF_1         CHAR(49),                   16910000
198             2 RECORD_CITY            CHAR(15),                   16920000
199             2 RECORD_UNDEF_2         CHAR(6);                    16930000
200                                                                   16940000
201       DECLARE                                                     16950000
202          RECORD_IMAGE_SCALAR         DEFINED RECORD_IMAGE        16960000
203                                      CHAR(80);                   16970000
204                                                                   16980000
205       DECLARE                                                     16990000
206          1 PRINT_LINE,                                            17000000
207             2 CARRIAGE_CONTROL       CHAR(1)   INIT(' '),        17010000
208             2 PRINT_AREA             CHAR(132);                  17020000
209                                                                   17030000
210    %INCLUDE (VSAMIO);                                             17040000
211    %INCLUDE (VSAMIOFB);                                           17050000
212                                                                   17060000
213       END KSDSUPD;                                                17070000
```

INCLUDED TEXT FOLLOWS FROM DD.MEMBER = SYSLIB .VSAMIO

```
214    /*****************************************************************31100000
215                                                                   31110000
216             VV   VV    SSSSS       A      M     M    IIII      OOOOO      31120000
217             VV   VV   SS   SS     AAA     MM   MM     II      OO   OO     31130000
218             VV   VV   SS         AA AA    MMM MMM     II      OO   OO     31140000
219             VV   VV    SSSSS    AA   AA   MMMMMMM     II      OO   OO     31150000
220             VV   VV        SS   AA   AA   MM M MM     II      OO   OO     31160000
```

MACRO SOURCE2 LISTING

```
221             VV VV    SS   SS   AAAAAAA   MM   MM     II      OO   OO      31170000
222              VVV     SS   SS   AA   AA   MM   MM     II      OO   OO      31180000
223               V       SSSSS    AA   AA   MM   MM    IIII      OOOOO       31190000
224                                                                            31200000
225     ***************************************************************      31210000
226      THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS   31220000
227      ROUTINE.                                                              31230000
228                                                                            31240000
229      THE VSIO_PARAMETER_VALUES SUPPLY THE VALUES USED TO MOVE INTO         31250000
230      PARAMETER ENTRIES TO TAILOR THE ROUTINE TO A SPECIFIC DATASET AND     31260000
231      TO PROVIDE COMMANDS TO DRIVE THE ROUTINE.                             31270000
232     ***************************************************************      /31280000
233                                                                            31290000
234             DECLARE                                                        31300000
235                1 VSIO_PARAMETER_VALUES   STATIC,                           31310000
236                   2 VSIO_OPEN            CHAR(8)   INIT('OPEN    '),      31320000
237                   2 VSIO_CLOSE           CHAR(8)   INIT('CLOSE   '),      31330000
238                   2 VSIO_READ            CHAR(8)   INIT('READ    '),      31340000
239                   2 VSIO_WRITE           CHAR(8)   INIT('WRITE   '),      31350000
240                   2 VSIO_REWRITE         CHAR(8)   INIT('REWRITE '),      31360000
241                   2 VSIO_DELETE          CHAR(8)   INIT('DELETE  '),      31370000
242                   2 VSIO_START_EQUAL     CHAR(8)   INIT('STARTEQ '),      31380000
243                   2 VSIO_START_NOTLESS   CHAR(8)   INIT('STARTGE '),      31390000
244                   2 VSIO_KSDS            CHAR(4)   INIT('KSDS'),          31400000
245                   2 VSIO_ESDS            CHAR(4)   INIT('ESDS'),          31410000
246                   2 VSIO_RRDS            CHAR(4)   INIT('RRDS'),          31420000
247                   2 VSIO_SEQUENTIAL      CHAR(10)  INIT('SEQUENTIAL'),    31430000
248                   2 VSIO_DIRECT          CHAR(10)  INIT('DIRECT    '),    31440000
249                   2 VSIO_DYNAMIC         CHAR(10)  INIT('DYNAMIC   '),    31450000
250                   2 VSIO_INPUT           CHAR(6)   INIT('INPUT '),        31460000
251                   2 VSIO_OUTPUT          CHAR(6)   INIT('OUTPUT'),        31470000
252                   2 VSIO_INPUT_OUTPUT    CHAR(6)   INIT('UPDATE'),        31480000
253                   2 (VSIO_RC_SUCCESS               INIT(0),               31490000
254                      VSIO_RC_LOGIC_ERROR           INIT(8),               31500000
255                      VSIO_RC_END_OF_FILE           INIT(9999),            31510000
256                      VSIO_RC_UNKNOWN_COMMAND       INIT(20),              31520000
257                      VSIO_RC_DATASET_ALREADY_OPEN  INIT(21),              31530000
258                      VSIO_RC_DATASET_NOT_OPEN      INIT(22),              31540000
259                      VSIO_RC_ORGANIZATION_UNKNOWN  INIT(23),              31550000
260                      VSIO_RC_ACCESS_UNKNOWN        INIT(24),              31560000
261                      VSIO_RC_ORG_ACCESS_MISMATCH   INIT(25),              31570000
262                      VSIO_RC_MODE_UNKNOWN          INIT(26),              31580000
263                      VSIO_RC_MODE_UNSUPPORTED      INIT(27),              31590000
264                      VSIO_RC_DDNAME_BLANK          INIT(28))              31600000
265                                  FIXED BINARY(15,0),                       31610000
```

MACRO SOURCE2 LISTING

```
266                  2 (VSIO_FB_DUPLICATE_RECORD      INIT(8),               31620000
267                     VSIO_FB_KEY_SEQUENCE          INIT(12),              31630000
268                     VSIO_FB_RECORD_NOT_FOUND      INIT(16),              31640000
269                     VSIO_FB_NO_MORE_SPACE         INIT(28),              31650000
270                     VSIO_FB_READ_WITHOUT_START    INIT(88))              31660000
271                                    FIXED BINARY(15,0),                   31670000
272     /*************************************************************31680000
273       THE VSIO_PARAMETER_BLOCK IS THE COMMUNICATION INTERFACE TO THE      31690000
274       THE ROUTINE.                                                        31700000
275      *************************************************************/31710000
276                                                                           31720000
277                1 VSIO_PARAMETER_BLOCK    STATIC,                          31730000
278                   2 VSIO_COMMAND         CHAR(8)   INIT(' '),             31740000
279                   2 (VSIO_RETURN_CODE,                                    31750000
280                      VSIO_VSAM_RC,                                        31760000
281                      VSIO_VSAM_FUNCTION,                                  31770000
282                      VSIO_VSAM_FEEDBACK) FIXED BINARY(15,0) INIT(0);      31780000
283                                                                           31790000
284     /*************************************************************31800000
285                             END OF VSAMIO COPY BOOK                       31810000
286      *************************************************************/31820000
```

INCLUDED TEXT FOLLOWS FROM DD.MEMBER = SYSLIB .VSAMIOFB

```
287     /*************************************************************00000100
288                                                                   00000200
289        VV   VV   SSSSS      A     M     M  IIII   OOOOO   FFFFFFF  BBBBBB  00000300
290        VV   VV  SS   SS    AAA    MM   MM   II   OO   OO  FF       BB   BB 00000400
291        VV   VV  SS        AA AA   MMM MMM   II   OO   OO  FF       BB   BB 00000500
292        VV   VV   SSSSS   AA   AA  MMMMMMM   II   OO   OO  FFFFF    BBBBBB  00000600
293        VV   VV        SS  AA   AA  MM M MM   II   OO   OO  FF       BB   BB 00000700
294         VV VV    SS   SS  AAAAAAA  MM   MM   II   OO   OO  FF       BB   BB 00000800
295          VVV     SS   SS  AA   AA  MM   MM   II   OO   OO  FF       BB   BB 00000900
296           V       SSSSS   AA   AA  MM   MM  IIII   OOOOO   FF       BBBBBB  00001000
297                                                                   00001100
298      *************************************************************00001200
299        THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS 00001300
300        ROUTINE, AND ARE USED TO COMMUNICATE CHARACTERISTICS FOR A SINGLE   00001400
301        VSAM DATASET.                                                      00001500
302                                                                           00001600
303        WITH THE 2 EXCEPTIONS FOR RECORD LENGTH (TO ACCOMODATE VARIABLE    00001700
304        LENGTH RECORDS) AND RELATIVE RECORD (TO ACCOMODATE RELATIVE RECORD  00001800
305        DATASETS), THESE DATA NAMES MUST BE POPULATED PRIOR TO CALLING THE  00001900
306        ROUTINE TO OPEN THE DATASET AND MUST NOT THEN BE CHANGED UNTIL THE  00002000
```

MACRO SOURCE2 LISTING

```
307       DATASET HAS BEEN CLOSED.                                      00002100
308     ***************************************************************/00002200
309                                                                     00002300
310          DECLARE                                                    00002400
311             1 VSIO_FILE_BLOCK            STATIC,                    00002500
312                2 VSFB_DDNAME             CHAR(8)   INIT(' '),       00002600
313                2 VSFB_ORGANIZATION       CHAR(4)   INIT(' '),       00002700
314                2 VSFB_ACCESS             CHAR(10)  INIT(' '),       00002800
315                2 VSFB_MODE               CHAR(6)   INIT(' '),       00002900
316                2 (VSFB_RECORD_LENGTH,                               00003000
317                   VSFB_KEY_POSITION,                                00003100
318                   VSFB_KEY_LENGTH)       FIXED BINARY(15,0) INIT(0),00003200
319                2 VSFB_FILE_STATUS        CHAR(1)   INIT('C'),       00003300
320                2 VSFB_RESERVED           CHAR(161);                 00003400
321                                                                     00003500
322     /***************************************************************00003600
323                           END OF VSAMIOFB COPY BOOK                 00003700
324      ***************************************************************/00003800
```

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

SOURCE LISTING.

```
                /*****************************************************************    1
                                                                                       1
                  KSDSUPDT - TESTS THE VSAMIO ROUTINE BY READING RECORDS FROM A KSDS   1
                             CLUSTER SEQUENTIALLY AND THEN REWRITING SOME OF THE       1
                             RECORDS WITH UPDATED CONTENTS.                            1
                                                                                       1
                 *****************************************************************/    1
                                                                                       7
 1               KSDSUPD:                                                              8
                    PROCEDURE OPTIONS(MAIN);                                           9
                                                                                      10
 2                     ON ERROR                                                     11
 3                        BEGIN;                                                    12
 4                           ON ERROR SYSTEM;                                       13
 5                           PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*'); 14
 6                           PUT SKIP DATA;                                         15
 7                           PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*'); 16
 8                        END;                                                      17
                                                                                      18
 9                     OPEN                                                         19
                          FILE(PRINTR) LINESIZE(133);                               20
                                                                                    21
10                     PRINT_AREA = 'KSDSUPDT: READ/REWRITE KSDS SEQUENTIALLY';     22
11                     WRITE FILE(PRINTR) FROM(PRINT_LINE);                         23
12                     PRINT_AREA = '---------------------------------------';      24
13                     WRITE FILE(PRINTR) FROM(PRINT_LINE);                         25
14                     PRINT_AREA = ' ';                                            26
15                     WRITE FILE(PRINTR) FROM(PRINT_LINE);                         27
                                                                                    28
16                     MORE_RECORDS = YES;                                          29
                                                                                    30
                  /*****************************************************************   31
                    ESTABLISH PARAMETERS OF VSAM DATASET AND CALL ROUTINE TO OPEN IT     31
                   *****************************************************************/   31
                                                                                      33
17                     VSFB_DDNAME = 'KSDSF01';                                     34
18                     VSFB_ORGANIZATION = VSIO_KSDS;                               35
19                     VSFB_ACCESS = VSIO_SEQUENTIAL;                               36
20                     VSFB_MODE = VSIO_INPUT_OUTPUT;                               37
21                     VSFB_RECORD_LENGTH = 80;                                     38
22                     VSFB_KEY_POSITION = 0;                                       39
23                     VSFB_KEY_LENGTH = 10;                                        40
24                     VSIO_COMMAND = VSIO_OPEN;                                    41
```

```
25                   CALL VSAMIOP (VSIO_PARAMETER_BLOCK,
                                   VSIO_FILE_BLOCK,
                                   RECORD_IMAGE);
26                   IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN
27                      DO;
28                         CALL VSIO_ERROR;
29                         RETURN;
30                      END;

31                   DO WHILE(MORE_RECORDS);
32                      CALL READ_KS;
33                      IF (MORE_RECORDS) THEN
34                         IF (RECORD_CITY = 'SAN ANTONIO    ' |
                               RECORD_CITY = 'DETROIT        ') THEN
35                            CALL UPDATE_RECORD;
36                   END;

                  /*****************************************************************
                    CALL ROUTINE TO CLOSE VSAM DATASET
                   *****************************************************************/

37                   VSIO_COMMAND = VSIO_CLOSE;
38                   CALL VSAMIOP (VSIO_PARAMETER_BLOCK,
                                   VSIO_FILE_BLOCK,
                                   RECORD_IMAGE);
39                   IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN
40                      CALL VSIO_ERROR;

41                   RETURN;

42                UPDATE_RECORD:
                     PROCEDURE;

43                   COUNTER_EDIT = RECORD_COUNTER;
44                   PRINT_AREA = COUNTER_EDIT || ':  KEY: ' || RECORD_KEY ||
                                  '  DATA: ' || RECORD_IMAGE_SCALAR || '  BEFORE';
45                   WRITE FILE(PRINTR) FROM(PRINT_LINE);

46                   IF (RECORD_CITY = 'SAN ANTONIO    ') THEN
47                      DO;
48                         RECORD_CITY = 'AUSTIN';
49                         CALL REWRITE_KS;
50                         IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN
51                            DO;
52                               PRINT_AREA = (36)' ' ||
                                              RECORD_IMAGE_SCALAR || '  AFTER';
53                               WRITE FILE(PRINTR) FROM(PRINT_LINE);
```

```
54                            END;                                                  88
55                         END;                                                     89
56                      ELSE                                                        90
56                         DO;                                                      91
57                            CALL DELETE_KS;                                       92
58                            IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN          93
59                               DO;                                                94
60                                  PRINT_AREA = (116)' ' || '  DELETED';           95
61                                  WRITE FILE(PRINTR) FROM(PRINT_LINE);            96
62                               END;                                               97
63                         END;                                                     98
                                                                                    99
64                      RETURN;                                                    100
                                                                                   101
65                   END UPDATE_RECORD;                                            102
                                                                                   103
66                DELETE_KS:                                                       104
                     PROCEDURE;                                                    105
                                                                                   106
                  /*****************************************************************  107
                    CALL ROUTINE TO DELETE LAST RECORD READ FROM VSAM DATASET      107
                   *****************************************************************/  107
                                                                                   109
67                       VSIO_COMMAND = VSIO_DELETE;                               110
68                       CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                       111
                                       VSIO_FILE_BLOCK,                            112
                                       RECORD_IMAGE);                              113
69                       IF (VSIO_RETURN_CODE ª= VSIO_RC_SUCCESS) THEN             114
70                          CALL VSIO_ERROR;                                       115
                                                                                   116
71                       RETURN;                                                   117
                                                                                   118
72                    END DELETE_KS;                                               119
                                                                                   120
73                READ_KS:                                                         121
                     PROCEDURE;                                                    122
                                                                                   123
                  /*****************************************************************  124
                    CALL ROUTINE TO READ NEXT RECORD FROM VSAM DATASET             124
                   *****************************************************************/  124
                                                                                   126
74                       VSIO_COMMAND = VSIO_READ;                                 127
75                       CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                       128
                                       VSIO_FILE_BLOCK,                            129
                                       RECORD_IMAGE);                              130
76                       IF (VSIO_RETURN_CODE ª= VSIO_RC_SUCCESS) THEN             131
77                          IF (VSIO_RETURN_CODE = VSIO_RC_END_OF_FILE) THEN       132
```

```
78                         MORE_RECORDS = NO;                                  133
79                      ELSE                                                   134
79                         CALL VSIO_ERROR;                                    135
80                   ELSE                                                      136
80                      RECORD_COUNTER = RECORD_COUNTER + 1;                   137
                                                                               138
81                      RETURN;                                                139
                                                                               140
82                   END READ_KS;                                              141
                                                                               142
83              REWRITE_KS:                                                    143
                   PROCEDURE;                                                  144
                                                                               145
                /*****************************************************************  146
                  CALL ROUTINE TO REWRITE PREVIOUSLY READ RECORD TO VSAM DATASET  146
                 *****************************************************************/  146
                                                                               148
84                      VSIO_COMMAND = VSIO_REWRITE;                           149
85                      CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                    150
                                      VSIO_FILE_BLOCK,                         151
                                      RECORD_IMAGE);                           152
86                      IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN          153
87                         CALL VSIO_ERROR;                                    154
                                                                               155
88                      RETURN;                                                156
                                                                               157
89                   END REWRITE_KS;                                           158
                                                                               159
 90              VSIO_ERROR:                                                   160
                   PROCEDURE;                                                  161
 91                   PRINT_AREA = 'VSAMIO ERROR OCCURRED DURING ' ||          162
                                   VSIO_COMMAND;                               163
 92                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                     164
 93                   PRINT_AREA = 'VSIO_RETURN_CODE = ' ||                    165
                                   VSIO_RETURN_CODE;                           166
 94                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                     167
 95                   PRINT_AREA = 'VSIO_VSAM_RETURN_CODE = ' ||               168
                                   VSIO_VSAM_RETURN_CODE;                      169
 96                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                     170
 97                   PRINT_AREA = 'VSIO_VSAM_FUNCTION_CODE = ' ||             171
                                   VSIO_VSAM_FUNCTION_CODE;                    172
 98                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                     173
 99                   PRINT_AREA = 'VSIO_VSAM_FEEDBACK_CODE = ' ||             174
                                   VSIO_VSAM_FEEDBACK_CODE;                    175
100                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                     176
101                   PRINT_AREA = ' ';                                        177
                                                                               178
```

```
102                  RETURN;                                                    179
                                                                                180
103               END VSIO_ERROR;                                               181
                                                                                182
104               DECLARE                                                       183
                     PRINTR FILE OUTPUT RECORD SEQUENTIAL EXTERNAL              184
                        ENV(F CTLASA);                                          185
                                                                                186
105               DECLARE                                                       187
                     COUNTER_EDIT                PICTURE 'ZZ,ZZZ,ZZ9V',     188
                     MORE_RECORDS                BIT(1),                     189
                     NO                          BIT(1)    INIT('0'B),       190
                     RECORD_COUNTER              FIXED BINARY(15,0),         191
                     YES                         BIT(1)    INIT('1'B);       192
                                                                               193
106               DECLARE                                                      194
                     1 RECORD_IMAGE,                                          195
                        2 RECORD_KEY             CHAR(10),                    196
                        2 RECORD_UNDEF_1         CHAR(49),                    197
                        2 RECORD_CITY            CHAR(15),                    198
                        2 RECORD_UNDEF_2         CHAR(6);                     199
                                                                              200
107               DECLARE                                                     201
                     RECORD_IMAGE_SCALAR         DEFINED RECORD_IMAGE         202
                                                 CHAR(80);                    203
                                                                              204
108               DECLARE                                                     205
                     1 PRINT_LINE,                                            206
                        2 CARRIAGE_CONTROL       CHAR(1)   INIT(' '),         207
                        2 PRINT_AREA             CHAR(132);                   208
                                                                              209
               /*************************************************************** 214
                                                                               214
                     VV   VV    SSSSS       A      M     M    IIII      OOOOO   214
                     VV   VV   SS   SS     AAA     MM   MM     II      OO   OO  214
                     VV   VV   SS         AA AA    MMM MMM     II      OO   OO  214
                     VV   VV    SSSSS    AA   AA   MMMMMMM     II      OO   OO  214
                     VV   VV        SS   AA   AA   MM M MM     II      OO   OO  214
                      VV VV    SS   SS   AAAAAAA   MM   MM     II      OO   OO  214
                       VVV     SS   SS   AA   AA   MM   MM     II      OO   OO  214
                        V       SSSSS    AA   AA   MM   MM    IIII      OOOOO   214
                                                                               214
                *************************************************************** 214
                 THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS 214
                 ROUTINE.                                                      214
                                                                               214
                 THE VSIO_PARAMETER_VALUES SUPPLY THE VALUES USED TO MOVE INTO 214
```

```
                  PARAMETER ENTRIES TO TAILOR THE ROUTINE TO A SPECIFIC DATASET AND      214
                  TO PROVIDE COMMANDS TO DRIVE THE ROUTINE.                              214
                 ***************************************************************/      214
                                                                                        232
109                       DECLARE                                                       234
                             1 VSIO_PARAMETER_VALUES   STATIC,                          235
                                2 VSIO_OPEN            CHAR(8)   INIT('OPEN    '),     236
                                2 VSIO_CLOSE           CHAR(8)   INIT('CLOSE   '),     237
                                2 VSIO_READ            CHAR(8)   INIT('READ    '),     238
                                2 VSIO_WRITE           CHAR(8)   INIT('WRITE   '),     239
                                2 VSIO_REWRITE         CHAR(8)   INIT('REWRITE '),     240
                                2 VSIO_DELETE          CHAR(8)   INIT('DELETE  '),     241
                                2 VSIO_START_EQUAL     CHAR(8)   INIT('STARTEQ '),     242
                                2 VSIO_START_NOTLESS   CHAR(8)   INIT('STARTGE '),     243
                                2 VSIO_KSDS            CHAR(4)   INIT('KSDS'),         244
                                2 VSIO_ESDS            CHAR(4)   INIT('ESDS'),         245
                                2 VSIO_RRDS            CHAR(4)   INIT('RRDS'),         246
                                2 VSIO_SEQUENTIAL      CHAR(10)  INIT('SEQUENTIAL'),   247
                                2 VSIO_DIRECT          CHAR(10)  INIT('DIRECT    '),   248
                                2 VSIO_DYNAMIC         CHAR(10)  INIT('DYNAMIC   '),   249
                                2 VSIO_INPUT           CHAR(6)   INIT('INPUT '),       250
                                2 VSIO_OUTPUT          CHAR(6)   INIT('OUTPUT'),       251
                                2 VSIO_INPUT_OUTPUT    CHAR(6)   INIT('UPDATE'),       252
                                2 (VSIO_RC_SUCCESS               INIT(0),              253
                                   VSIO_RC_LOGIC_ERROR           INIT(8),              254
                                   VSIO_RC_END_OF_FILE           INIT(9999),           255
                                   VSIO_RC_UNKNOWN_COMMAND       INIT(20),             256
                                   VSIO_RC_DATASET_ALREADY_OPEN  INIT(21),             257
                                   VSIO_RC_DATASET_NOT_OPEN      INIT(22),             258
                                   VSIO_RC_ORGANIZATION_UNKNOWN  INIT(23),             259
                                   VSIO_RC_ACCESS_UNKNOWN        INIT(24),             260
                                   VSIO_RC_ORG_ACCESS_MISMATCH   INIT(25),             261
                                   VSIO_RC_MODE_UNKNOWN          INIT(26),             262
                                   VSIO_RC_MODE_UNSUPPORTED      INIT(27),             263
                                   VSIO_RC_DDNAME_BLANK          INIT(28))             264
                                                    FIXED BINARY(15,0),                265
                                2 (VSIO_FB_DUPLICATE_RECORD      INIT(8),              266
                                   VSIO_FB_KEY_SEQUENCE          INIT(12),             267
                                   VSIO_FB_RECORD_NOT_FOUND      INIT(16),             268
                                   VSIO_FB_NO_MORE_SPACE         INIT(28),             269
                                   VSIO_FB_READ_WITHOUT_START    INIT(88))             270
                                                    FIXED BINARY(15,0),                271
                          /*************************************************************  272
                            THE VSIO_PARAMETER_BLOCK IS THE COMMUNICATION INTERFACE TO THE  272
                            THE ROUTINE.                                                  272
                           *************************************************************/  272
                                                                                        275
```

```
                  1 VSIO_PARAMETER_BLOCK    STATIC,                                 277
                    2 VSIO_COMMAND          CHAR(8)   INIT(' '),                    278
                    2 (VSIO_RETURN_CODE,                                            279
                       VSIO_VSAM_RC,                                                280
                       VSIO_VSAM_FUNCTION,                                          281
                       VSIO_VSAM_FEEDBACK) FIXED BINARY(15,0) INIT(0);              282
                                                                                    283
             /*****************************************************************    284
                              END OF VSAMIO COPY BOOK                               284
              *****************************************************************/    284
             /*****************************************************************    287
                                                                                    287
               VV   VV   SSSSS      A     M     M  IIII   OOOOO   FFFFFFF  BBBBBB    287
               VV   VV  SS   SS    AAA    MM   MM   II   OO   OO  FF       BB   BB   287
               VV   VV  SS        AA AA   MMM MMM   II   OO   OO  FF       BB   BB   287
               VV   VV   SSSSS   AA   AA  MMMMMMM   II   OO   OO  FFFFF    BBBBBB    287
               VV   VV       SS  AA   AA  MM M MM   II   OO   OO  FF       BB   BB   287
                VV VV   SS   SS  AAAAAAA  MM   MM   II   OO   OO  FF       BB   BB   287
                 VVV    SS   SS  AA   AA  MM   MM   II   OO   OO  FF       BB   BB   287
                  V      SSSSS   AA   AA  MM   MM  IIII   OOOOO   FF       BBBBBB    287
                                                                                    287
              *****************************************************************    287
               THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS   287
               ROUTINE, AND ARE USED TO COMMUNICATE CHARACTERISTICS FOR A SINGLE     287
               VSAM DATASET.                                                        287
                                                                                    287
               WITH THE 2 EXCEPTIONS FOR RECORD LENGTH (TO ACCOMODATE VARIABLE       287
               LENGTH RECORDS) AND RELATIVE RECORD (TO ACCOMODATE RELATIVE RECORD    287
               DATASETS), THESE DATA NAMES MUST BE POPULATED PRIOR TO CALLING THE    287
               ROUTINE TO OPEN THE DATASET AND MUST NOT THEN BE CHANGED UNTIL THE    287
               DATASET HAS BEEN CLOSED.                                             287
             *****************************************************************/    287
                                                                                    308
110                 DECLARE                                                         310
                      1 VSIO_FILE_BLOCK           STATIC,                           311
                        2 VSFB_DDNAME             CHAR(8)   INIT(' '),              312
                        2 VSFB_ORGANIZATION       CHAR(4)   INIT(' '),              313
                        2 VSFB_ACCESS             CHAR(10)  INIT(' '),              314
                        2 VSFB_MODE               CHAR(6)   INIT(' '),              315
                        2 (VSFB_RECORD_LENGTH,                                      316
                           VSFB_KEY_POSITION,                                       317
                           VSFB_KEY_LENGTH)       FIXED BINARY(15,0) INIT(0),       318
                        2 VSFB_FILE_STATUS        CHAR(1)   INIT('C'),              319
                        2 VSFB_RESERVED           CHAR(161);                        320
                                                                                    321
             /*****************************************************************    322
                              END OF VSAMIOFB COPY BOOK                             322
```

```
                 **********/  322
                               212
111                END KSDSUPD;  213
```

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
108		CARRIAGE_CONTROL	IN PRINT_LINE,AUTOMATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER
105		COUNTER_EDIT	AUTOMATIC,UNALIGNED,DECIMAL,PICTURE(ZZ,ZZZ,ZZ9V) 43,44
66		DELETE_KS	ENTRY,DECIMAL,FLOAT(SINGLE) 57
1	*********	KSDSUPD	ENTRY,BINARY,FIXED(15,0)
105		MORE_RECORDS	AUTOMATIC,UNALIGNED,STRING(1),BIT 16,31,33,78
105		NO	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 78
108		PRINT_AREA	IN PRINT_LINE,AUTOMATIC,UNALIGNED,STRING(132),CHARACTER 10,12,14,44,52,60,91,93,95,97,99,101
108		PRINT_LINE	AUTOMATIC,STRUCTURE 11,13,15,45,53,61,92,94,96,98,100
104		PRINTR	FILE,EXTERNAL,OUTPUT,RECORD,SEQUENTIAL,ENVIRONMENT(F CTLASA) 9,11,13,15,45,53,61,92,94,96,98,100
73		READ_KS	ENTRY,DECIMAL,FLOAT(SINGLE) 32
106		RECORD_CITY	IN RECORD_IMAGE,AUTOMATIC,UNALIGNED,STRING(15),CHARACTER 34,34,46,48
105	*********	RECORD_COUNTER	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 43,80,80
106		RECORD_IMAGE	AUTOMATIC,STRUCTURE 25,38,68,75,85
107		RECORD_IMAGE_SCALAR	AUTOMATIC,DEFINED,UNALIGNED,STRING(80),CHARACTER 44,52
106		RECORD_KEY	IN RECORD_IMAGE,AUTOMATIC,UNALIGNED,STRING(10),CHARACTER

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
			44
106		RECORD_UNDEF_1	IN RECORD_IMAGE,AUTOMATIC,UNALIGNED,STRING(49),CHARACTER
106		RECORD_UNDEF_2	IN RECORD_IMAGE,AUTOMATIC,UNALIGNED,STRING(6),CHARACTER
83		REWRITE_KS	ENTRY,DECIMAL,FLOAT(SINGLE)
			49
		SYSPRINT	FILE,EXTERNAL
			5,6,7
42		UPDATE_RECORD	ENTRY,DECIMAL,FLOAT(SINGLE)
			35
		VSAMIOP	EXTERNAL,ENTRY,DECIMAL,FLOAT(SINGLE)
			25,38,68,75,85
110		VSFB_ACCESS	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(10),CHARACTER
			19
110		VSFB_DDNAME	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(8),CHARACTER
			17
110		VSFB_FILE_STATUS	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER
110	*********	VSFB_KEY_LENGTH	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 23
110	*********	VSFB_KEY_POSITION	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 22
110		VSFB_MODE	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(6),CHARACTER 20
110		VSFB_ORGANIZATION	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(4),CHARACTER 18
110	*********	VSFB_RECORD_LENGTH	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 21
110		VSFB_RESERVED	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,STRING(161),CHARACTER
109		VSIO_CLOSE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		37
109	VSIO_COMMAND	IN VSIO_PARAMETER_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
		24,37,67,74,84,91
109	VSIO_DELETE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
		67
109	VSIO_DIRECT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER
109	VSIO_DYNAMIC	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER
90	VSIO_ERROR	ENTRY,DECIMAL,FLOAT(SINGLE)
		28,40,70,79,87
109	VSIO_ESDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER
109 *********	VSIO_FB_DUPLICATE_RECORD	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
109 *********	VSIO_FB_KEY_SEQUENCE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
109 *********	VSIO_FB_NO_MORE_SPACE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
109 *********	VSIO_FB_READ_WITHOUT_START	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
109 *********	VSIO_FB_RECORD_NOT_FOUND	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	VSIO_FILE_BLOCK	STATIC,STRUCTURE
		25,38,68,75,85
109	VSIO_INPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER
109	VSIO_INPUT_OUTPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
			20
109		VSIO_KSDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER
			18
109		VSIO_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
			24
109		VSIO_OUTPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER
109		VSIO_PARAMETER_BLOCK	STATIC,STRUCTURE
			25,38,68,75,85
109		VSIO_PARAMETER_VALUES	STATIC,STRUCTURE
109	*********	VSIO_RC_ACCESS_UNKNOWN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
			(15,0)
109	*********	VSIO_RC_DATASET_ALREADY_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
			(15,0)
109	*********	VSIO_RC_DATASET_NOT_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
			(15,0)
109	*********	VSIO_RC_DDNAME_BLANK	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
			(15,0)
109	*********	VSIO_RC_END_OF_FILE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
			(15,0)
			77
109	*********	VSIO_RC_LOGIC_ERROR	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
			(15,0)
109	*********	VSIO_RC_MODE_UNKNOWN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
			(15,0)
109	*********	VSIO_RC_MODE_UNSUPPORTED	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
			(15,0)
109	*********	VSIO_RC_ORG_ACCESS_MISMATCH	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
			(15,0)

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
109	*********	VSIO_RC_ORGANIZATION_UNKNOWN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
109	*********	VSIO_RC_SUCCESS	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 26,39,50,58,69,76,86
109	*********	VSIO_RC_UNKNOWN_COMMAND	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
109		VSIO_READ	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 74
109	*********	VSIO_RETURN_CODE	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 26,39,50,58,69,76,77,86,93
109		VSIO_REWRITE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 84
109		VSIO_RRDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER
109		VSIO_SEQUENTIAL	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER 19
109		VSIO_START_EQUAL	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
109		VSIO_START_NOTLESS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
109	*********	VSIO_VSAM_FEEDBACK	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
		VSIO_VSAM_FEEDBACK_CODE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 99
109	*********	VSIO_VSAM_FUNCTION	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)

```
DCL NO.          IDENTIFIER                        ATTRIBUTES AND REFERENCES

                 VSIO_VSAM_FUNCTION_CODE           AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE)
                                                   97
109    *********  VSIO_VSAM_RC                      IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                   (15,0)
                 VSIO_VSAM_RETURN_CODE             AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE)
                                                   95
109              VSIO_WRITE                        IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                   CHARACTER
105              YES                               AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT
                                                   16
```

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
108	PRINT_LINE	133
106	RECORD_IMAGE	80
110	VSIO_FILE_BLOCK	196
109	VSIO_PARAMETER_BLOCK	16
109	VSIO_PARAMETER_VALUES	158

STORAGE REQUIREMENTS.

THE STORAGE AREA FOR THE PROCEDURE LABELLED KSDSUPD IS 528 BYTES LONG.

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

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED UPDATE_RECORD IS 292 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 VSIO_ERROR IS 256 BYTES LONG.

THE PROGRAM CSECT IS NAMED KSDSUPD AND IS 2182 BYTES LONG.

THE STATIC CSECT IS NAMED KSDSUPDA AND IS 6544 BYTES LONG.

STATISTICS MACRO RECORDS = 324,SOURCE RECORDS = 328,PROG TEXT STMNTS = 111,OBJECT BYTES = 2182

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 PROCEDURE UPDATE_RECORD

OFFSET (HEX)	0000	0048	0066	009E	00B6	00C0	00C0	00CC	00D6	00E2	00E2	010C	0124	0124	0128	0128	0132	013E	013E	0148	0160
STATEMENT NO	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62

OFFSET (HEX)	0160	0160	0166																	
STATEMENT NO	63	64	65																	

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE DELETE_KS

OFFSET (HEX)	0000	0038	003E	005E	006A	0074	007A													
STATEMENT NO	66	67	68	69	70	71	72													

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE READ_KS

OFFSET (HEX)	0000	0034	003A	005A	0066	0072	0080	008E	009E	00A4										
STATEMENT NO	73	74	75	76	77	78	79	80	81	82										

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE REWRITE_KS

OFFSET (HEX)	0000	0038	003E	005E	006A	0074	007A													
STATEMENT NO	83	84	85	86	87	88	89													

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	90	91	92	93	94	95	96	97	98	99	100	101	102	103						

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE KSDSUPD

OFFSET (HEX)	0000	00D8	00E6	00F0	00FC	0114	0120	0138	0144	015C	0162	0168	016E	0174	017A	0180	0186	018C	0192	01AE	01BA
STATEMENT NO	1	2	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27

OFFSET (HEX)	01BA	01C4	01CA	01CA	01D2	01DC	01E4	021C	0226	022A	0230	024C	0258	0262	0268					
STATEMENT NO	28	29	30	31	32	33	34	35	36	37	38	39	40	41	111					

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 95, 97, 99.

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.
```

```
KSDSUPDT: READ/REWRITE KSDS SEQUENTIALLY
----------------------------------------

        1:  KEY: 0045557001  DATA: 0045557001 LISA L FONTAINE      714 SUNSET PLACE           SAN ANTONIO     TX     BEFORE
                                   0045557001 LISA L FONTAINE      714 SUNSET PLACE           AUSTIN          TX     AFTER
       18:  KEY: 1766129028  DATA: 1766129028 JOYCE R NORTON       907 BRANDYWINE CIRCLE      SAN ANTONIO     TX     BEFORE
                                   1766129028 JOYCE R NORTON       907 BRANDYWINE CIRCLE      AUSTIN          TX     AFTER
       29:  KEY: 2522284049  DATA: 2522284049 WANDA J TUCKER       7807 WINDTREE ROAD         SAN ANTONIO     TX     BEFORE
                                   2522284049 WANDA J TUCKER       7807 WINDTREE ROAD         AUSTIN          TX     AFTER
       47:  KEY: 4318291087  DATA: 4318291087 SAMANTHA M HALL     8063 EDMUND AVENUE         SAN ANTONIO     TX     BEFORE
                                   4318291087 SAMANTHA M HALL     8063 EDMUND AVENUE         AUSTIN          TX     AFTER
       73:  KEY: 6426347143  DATA: 6426347143 PATTY C ELISON       5454 LINCOLN AVENUE        DETROIT         MI     BEFORE
                                                                                                                     DELETED
```