J E S 2 J O B L O G

```
18.12.16 JOB  154  IEF677I WARNING MESSAGE(S) FOR JOB VSTESTK6 ISSUED
18.12.16 JOB  154  $HASP373 VSTESTK6 STARTED - INIT  1 - CLASS A - SYS HMVS
18.12.16 JOB  154  IEF403I VSTESTK6 - STARTED - TIME=18.12.16
18.12.17 JOB  154  CCI001C PL1L    /IEMAA   /00:00:00.17/        /00004/SYS     /VSTESTK6
18.12.17 JOB  154  CCI001C LKED    /IEWL    /00:00:00.04/        /00000/SYS     /VSTESTK6
18.12.17 JOB  154  CCI001C GO      /PGM=*.DD/00:00:00.01/        /00000/SYS     /VSTESTK6
18.12.17 JOB  154  IEF404I VSTESTK6 - ENDED - TIME=18.12.17
18.12.17 JOB  154  $HASP395 VSTESTK6 ENDED

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

 07 JUL 20 JOB EXECUTION DATE

       22 CARDS READ

    1,172 SYSOUT PRINT RECORDS

        0 SYSOUT PUNCH RECORDS

     0.00 MINUTES EXECUTION TIME
```

```
 1      //VSTESTK6 JOB (SYS),'VSAMIOP IVP KSDSSSEQ',CLASS=A,MSGCLASS=X,         JOB  154
        //             REGION=4096K
        ***
        ********************************************************************
        *** PL/1 MODULE: KSDSSSEQ     VSAM DATASET: VSTESTKS.CLUSTER (KSDS)
        ***
        *** TESTS START AND READ FUNCTIONS AGAINST KSDS 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(KSDSSSEQ),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 VSTESTK6 PL1L PL1F
IEF237I 253  ALLOCATED TO STEPLIB
IEF237I 253  ALLOCATED TO SYS00358
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 VSTESTK6 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.JOB00154.SO0101                        SYSOUT
IEF285I   SYS20189.T181216.RA000.VSTESTK6.LOADSET      PASSED        *------219
IEF285I   VOL SER NOS= MVS380.
IEF285I   SYS20189.T181216.RA000.VSTESTK6.SYSUT3       DELETED       *------309
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T181216.RA000.VSTESTK6.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 00.17SEC SRB    0MIN 00.04SEC VIRT  4096K SYS   212K
**** JOBCARD READ 20189 18:12:16 ************************************************************************************
*                                      PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                              *
*  STEP NAME  PL1L        USER CORE        4096K   TAPES USED/IO 000/000000000   START   TIME  18:12:16    TCB TIME  00:00:00.17  *
*  PGM  NAME  IEMAA       SYSTEM CORE       212K   DISKS USED/IO 004/000000558   STOP    TIME  18:12:17    SRB TIME  00:00:00.04  *
*   COND CODE    0004     PRIVATE AREA SZ   4096K       ALLOC TIME  18:12:16      ELAPSED TIME                PGM LOAD  18:12:16   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004          3024   00:00:00.24            0             0           0             0              0             0            0  *
************************************************************************************************************************
*  CPU $ (     0.06) + EXCP $ (     0.75) + MEMORY $ (     1.98) = TOTAL $ (     2.79)                                  *
************************************************************************************************************************
IEF236I ALLOC. FOR VSTESTK6 LKED PL1F
IEF237I 253  ALLOCATED TO SYSLIB
IEF237I 253  ALLOCATED TO
IEF237I 253  ALLOCATED TO SYS00360
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 VSTESTK6 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.T181216.RA000.VSTESTK6.GOSET        PASSED        *-------57
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T181216.RA000.VSTESTK6.SYSUT1       DELETED       *--------0
IEF285I   VOL SER NOS= MVS370.
IEF285I   JES2.JOB00154.SO0102                        SYSOUT
```

```
IEF285I   SYS20189.T181216.RA000.VSTESTK6.LOADSET      DELETED       *------220
IEF285I   VOL SER NOS= MVS380.
IEF373I STEP /LKED    / START 20189.1812
IEF374I STEP /LKED    / STOP  20189.1812 CPU    0MIN 00.04SEC 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  18:12:17    TCB TIME  00:00:00.04  *
*  PGM  NAME  IEWL        SYSTEM CORE        208K   DISKS USED/IO 004/000000383   STOP    TIME  18:12:17    SRB TIME  00:00:00.01  *
*   COND CODE    0000     PRIVATE AREA SZ   4096K       ALLOC TIME  18:12:17      ELAPSED TIME                PGM LOAD  18:12:17   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004          1958   00:00:00.06            0             0         0              0              0             0  *
********************************************************************************************************************
*  CPU $ (     0.01) + EXCP $ (     0.51) + MEMORY $ (     0.02) = TOTAL $ (     0.54)                              *
********************************************************************************************************************
IEF236I ALLOC. FOR VSTESTK6 GO PL1F
IEF237I 251  ALLOCATED TO PGM=*.DD
IEF237I 253  ALLOCATED TO STEPLIB
IEF237I 253  ALLOCATED TO
IEF237I 253  ALLOCATED TO SYS00362
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 SYS00364
IEF142I VSTESTK6 GO PL1F - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS20189.T181216.RA000.VSTESTK6.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.JOB00154.SO0103                         SYSOUT
IEF285I   JES2.JOB00154.SO0104                         SYSOUT
IEF285I   JES2.JOB00154.SO0105                         SYSOUT
IEF285I   JES2.JOB00154.SO0106                         SYSOUT
IEF285I   PUB001.VSTESTKS.CLUSTER                      KEPT          *--------2
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   104K SYS   220K
********************************************************************************************************************
*                                        PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                           *
*  STEP NAME  GO          USER CORE          104K   TAPES USED/IO 000/000000000   START   TIME  18:12:17    TCB TIME  00:00:00.01  *
*  PGM  NAME  PGM=*.DD    SYSTEM CORE        220K   DISKS USED/IO 003/000000002   STOP    TIME  18:12:17    SRB TIME  00:00:00.00  *
*   COND CODE    0000     PRIVATE AREA SZ   4096K       ALLOC TIME  18:12:17      ELAPSED TIME                PGM LOAD  18:12:17   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004            35   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.T181217.RA000.VSTESTK6.R0000001     KEPT          *--------0
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T181216.RA000.VSTESTK6.GOSET        DELETED
IEF285I   VOL SER NOS= WORK00.
IEF375I  JOB /VSTESTK6/ START 20189.1812
IEF376I  JOB /VSTESTK6/ STOP  20189.1812 CPU    0MIN 00.22SEC 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    /*****************************************************************13000000
  2                                                                      13010000
  3      KSDSSSEQ - TESTS THE VSAMIO ROUTINE BY USING START AND READ COMMANDS 13020000
  4                 ON AN INDEXED CLUTER TO READ THE FILE IN SKIP-SEQUENTIAL  13030000
  5                 MODE.                                                13040000
  6                                                                      13050000
  7     ******************************************************************/13060000
  8    KSDSSSQ:                                                          13070000
  9       PROCEDURE OPTIONS(MAIN);                                       13080000
 10                                                                      13090000
 11          ON ERROR                                                    13100000
 12             BEGIN;                                                   13110000
 13                ON ERROR SYSTEM;                                      13120000
 14                PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*'); 13130000
 15                PUT SKIP DATA;                                        13140000
 16                PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*'); 13150000
 17             END;                                                     13160000
 18                                                                      13170000
 19          OPEN                                                        13180000
 20             FILE(PRINTR) LINESIZE(133);                              13190000
 21                                                                      13200000
 22          PRINT_AREA = 'KSDSSSEQ: READ KSDS SEQUENTIALLY (W/START)';  13210000
 23          WRITE FILE(PRINTR) FROM(PRINT_LINE);                        13220000
 24          PRINT_AREA = '----------------------------------------';    13230000
 25          WRITE FILE(PRINTR) FROM(PRINT_LINE);                        13240000
 26          PRINT_AREA = ' ';                                           13250000
 27          WRITE FILE(PRINTR) FROM(PRINT_LINE);                        13260000
 28                                                                      13270000
 29    /*****************************************************************13280000
 30      ESTABLISH PARAMETERS OF VSAM DATASET AND CALL ROUTINE TO OPEN IT     13290000
 31     ******************************************************************/13300000
 32          VSFB_DDNAME = 'KSDSF01';                                    13310000
 33          VSFB_ORGANIZATION = VSIO_KSDS;                              13320000
 34          VSFB_ACCESS = VSIO_SEQUENTIAL;                              13330000
 35          VSFB_MODE = VSIO_INPUT;                                     13340000
 36          VSFB_RECORD_LENGTH = 80;                                    13350000
 37          VSFB_KEY_POSITION = 0;                                      13360000
 38          VSFB_KEY_LENGTH = 10;                                       13370000
 39          VSIO_COMMAND = VSIO_OPEN;                                   13380000
 40          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                         13390000
 41                        VSIO_FILE_BLOCK,                              13400000
 42                        RECORD_IMAGE);                                13410000
 43          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN               13420000
 44             DO;                                                      13430000
```

MACRO SOURCE2 LISTING

```
 45                CALL VSIO_ERROR;                                    13440000
 46                RETURN;                                             13450000
 47              END;                                                  13460000
 48                                                                    13470000
 49           RECORD_KEY = '1033846021';                               13480000
 50           VSIO_COMMAND = VSIO_START_EQUAL;                         13490000
 51           PRINT_AREA = 'STARTING KEY EQUAL TO ' || RECORD_KEY;     13500000
 52           WRITE FILE(PRINTR) FROM(PRINT_LINE);                     13510000
 53           CALL PROCESS_BLOCK;                                      13520000
 54                                                                    13530000
 55           RECORD_KEY = '2534789096';                               13540000
 56           VSIO_COMMAND = VSIO_START_EQUAL;                         13550000
 57           PRINT_AREA = 'STARTING KEY EQUAL TO ' || RECORD_KEY;     13560000
 58           WRITE FILE(PRINTR) FROM(PRINT_LINE);                     13570000
 59           CALL PROCESS_BLOCK;                                      13580000
 60                                                                    13590000
 61           RECORD_KEY = '3284189067';                               13600000
 62           VSIO_COMMAND = VSIO_START_NOTLESS;                       13610000
 63           PRINT_AREA = 'STARTING KEY NOT LESS THAN ' || RECORD_KEY;13620000
 64           WRITE FILE(PRINTR) FROM(PRINT_LINE);                     13630000
 65           CALL PROCESS_BLOCK;                                      13640000
 66                                                                    13650000
 67           RECORD_KEY = '3860000000';                               13660000
 68           VSIO_COMMAND = VSIO_START_NOTLESS;                       13670000
 69           PRINT_AREA = 'STARTING KEY NOT LESS THAN ' || RECORD_KEY;13680000
 70           WRITE FILE(PRINTR) FROM(PRINT_LINE);                     13690000
 71           CALL PROCESS_BLOCK;                                      13700000
 72                                                                    13710000
 73      /*****************************************************************13720000
 74        CALL ROUTINE TO CLOSE VSAM DATASET                          13730000
 75       *****************************************************************/13740000
 76           VSIO_COMMAND = VSIO_CLOSE;                               13750000
 77           CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                      13760000
 78                         VSIO_FILE_BLOCK,                           13770000
 79                         RECORD_IMAGE);                             13780000
 80           IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN            13790000
 81              CALL VSIO_ERROR;                                      13800000
 82                                                                    13810000
 83           RETURN;                                                  13820000
 84                                                                    13830000
 85      PROCESS_BLOCK:                                                13840000
 86         PROCEDURE;                                                 13850000
 87                                                                    13860000
 88           CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                      13870000
 89                         VSIO_FILE_BLOCK,                           13880000
```

MACRO SOURCE2 LISTING

```
 90                      RECORD_IMAGE);                                  13890000
 91          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN               13900000
 92             IF (VSIO_VSAM_FEEDBACK = VSIO_FB_RECORD_NOT_FOUND) THEN  13910000
 93                DO;                                                   13920000
 94                   PRINT_AREA = '*** NOT FOUND ***';                  13930000
 95                   WRITE FILE(PRINTR) FROM(PRINT_LINE);               13940000
 96                   RETURN;                                            13950000
 97                END;                                                  13960000
 98             ELSE                                                     13970000
 99                DO;                                                   13980000
100                   CALL VSIO_ERROR;                                   13990000
101                   RETURN;                                            14000000
102                END;                                                  14010000
103                                                                      14020000
104          VSIO_COMMAND = VSIO_READ;                                   14030000
105          MORE_RECORDS = YES;                                         14040000
106          RECORD_COUNTER = 0;                                         14050000
107                                                                      14060000
108          DO WHILE(MORE_RECORDS & RECORD_COUNTER < 5);                14070000
109             CALL READ_KS;                                            14080000
110             IF (MORE_RECORDS) THEN                                   14090000
111                DO;                                                   14100000
112                   PRINT_AREA = 'KEY: ' || RECORD_KEY || '  DATA: ' ||14110000
113                                RECORD_IMAGE_SCALAR;                  14120000
114                   WRITE FILE(PRINTR) FROM(PRINT_LINE);               14130000
115                END;                                                  14140000
116             END;                                                     14150000
117                                                                      14160000
118          RETURN;                                                     14170000
119                                                                      14180000
120       END PROCESS_BLOCK;                                             14190000
121                                                                      14200000
122    READ_KS:                                                          14210000
123       PROCEDURE;                                                     14220000
124                                                                      14230000
125    /*****************************************************************14240000
126      CALL ROUTINE TO READ NEXT RECORD FROM VSAM DATASET              14250000
127     *****************************************************************/14260000
128          VSIO_COMMAND = VSIO_READ;                                   14270000
129          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                         14280000
130                        VSIO_FILE_BLOCK,                              14290000
131                        RECORD_IMAGE);                                14300000
132          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN               14310000
133             IF (VSIO_RETURN_CODE = VSIO_RC_END_OF_FILE) THEN         14320000
134                MORE_RECORDS = NO;                                    14330000
```

```
135             ELSE                                                    14340000
136                CALL VSIO_ERROR;                                     14350000
137          ELSE                                                       14360000
138             RECORD_COUNTER = RECORD_COUNTER + 1;                    14370000
139                                                                     14380000
140          RETURN;                                                    14390000
141                                                                     14400000
142       END READ_KS;                                                  14410000
143                                                                     14420000
144    VSIO_ERROR:                                                      14430000
145       PROCEDURE;                                                    14440000
146          PRINT_AREA = 'VSAMIO ERROR OCCURRED DURING ' ||            14450000
147                       VSIO_COMMAND;                                 14460000
148          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       14470000
149          PRINT_AREA = 'VSIO_RETURN_CODE = ' ||                      14480000
150                       VSIO_RETURN_CODE;                             14490000
151          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       14500000
152          PRINT_AREA = 'VSIO_VSAM_RETURN_CODE = ' ||                 14510000
153                       VSIO_VSAM_RETURN_CODE;                        14520000
154          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       14530000
155          PRINT_AREA = 'VSIO_VSAM_FUNCTION_CODE = ' ||               14540000
156                       VSIO_VSAM_FUNCTION_CODE;                      14550000
157          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       14560000
158          PRINT_AREA = 'VSIO_VSAM_FEEDBACK_CODE = ' ||               14570000
159                       VSIO_VSAM_FEEDBACK_CODE;                      14580000
160          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       14590000
161          PRINT_AREA = ' ';                                          14600000
162                                                                     14610000
163          RETURN;                                                    14620000
164                                                                     14630000
165       END VSIO_ERROR;                                               14640000
166                                                                     14650000
167       DECLARE                                                       14660000
168          PRINTR FILE OUTPUT RECORD SEQUENTIAL EXTERNAL              14670000
169             ENV(F CTLASA);                                          14680000
170                                                                     14690000
171       DECLARE                                                       14700000
172          MORE_RECORDS                  BIT(1),                    14710000
173          NO                            BIT(1)    INIT('0'B),      14720000
174          RECORD_COUNTER                FIXED BINARY(15,0) INIT(0), 14730000
175          YES                           BIT(1)    INIT('1'B);      14740000
176                                                                     14750000
177       DECLARE                                                       14760000
178          1 RECORD_IMAGE,                                            14770000
179             2 RECORD_KEY               CHAR(10),                  14780000
```

MACRO SOURCE2 LISTING

```
180             2 RECORD_FIELDS             CHAR(70);                       14790000
181                                                                         14800000
182        DECLARE                                                          14810000
183           RECORD_IMAGE_SCALAR           DEFINED RECORD_IMAGE            14820000
184                                         CHAR(80);                       14830000
185                                                                         14840000
186        DECLARE                                                          14850000
187           1 PRINT_LINE,                                                 14860000
188             2 CARRIAGE_CONTROL          CHAR(1)   INIT(' '),            14870000
189             2 PRINT_AREA                CHAR(132);                      14880000
190                                                                         14890000
191     %INCLUDE (VSAMIO);                                                  14900000
192     %INCLUDE (VSAMIOFB);                                                14910000
193                                                                         14920000
194        END KSDSSSQ;                                                     14930000
```

INCLUDED TEXT FOLLOWS FROM DD.MEMBER = SYSLIB .VSAMIO

```
195     /*****************************************************************31100000
196                                                                         31110000
197             VV   VV    SSSSS       A      M     M    IIII      OOOOO       31120000
198             VV   VV   SS   SS     AAA     MM   MM     II      OO   OO      31130000
199             VV   VV   SS         AA AA    MMM MMM     II      OO   OO      31140000
200             VV   VV    SSSSS    AA   AA   MMMMMMM     II      OO   OO      31150000
201              VV VV        SS   AA   AA   MM M MM     II      OO   OO      31160000
202              VV VV    SS   SS   AAAAAAA   MM   MM     II      OO   OO      31170000
203               VVV     SS   SS   AA   AA   MM   MM     II      OO   OO      31180000
204                V       SSSSS    AA   AA   MM   MM    IIII      OOOOO       31190000
205                                                                         31200000
206     *****************************************************************31210000
207      THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS  31220000
208      ROUTINE.                                                           31230000
209                                                                         31240000
210      THE VSIO_PARAMETER_VALUES SUPPLY THE VALUES USED TO MOVE INTO      31250000
211      PARAMETER ENTRIES TO TAILOR THE ROUTINE TO A SPECIFIC DATASET AND  31260000
212      TO PROVIDE COMMANDS TO DRIVE THE ROUTINE.                          31270000
213     *****************************************************************/31280000
214                                                                         31290000
215             DECLARE                                                     31300000
216                1 VSIO_PARAMETER_VALUES   STATIC,                        31310000
217                   2 VSIO_OPEN            CHAR(8)   INIT('OPEN    '),    31320000
218                   2 VSIO_CLOSE           CHAR(8)   INIT('CLOSE   '),    31330000
219                   2 VSIO_READ            CHAR(8)   INIT('READ    '),    31340000
220                   2 VSIO_WRITE           CHAR(8)   INIT('WRITE   '),    31350000
```

MACRO SOURCE2 LISTING

```
221                  2 VSIO_REWRITE         CHAR(8)   INIT('REWRITE '),      31360000
222                  2 VSIO_DELETE          CHAR(8)   INIT('DELETE  '),      31370000
223                  2 VSIO_START_EQUAL     CHAR(8)   INIT('STARTEQ '),      31380000
224                  2 VSIO_START_NOTLESS   CHAR(8)   INIT('STARTGE '),      31390000
225                  2 VSIO_KSDS            CHAR(4)   INIT('KSDS'),          31400000
226                  2 VSIO_ESDS            CHAR(4)   INIT('ESDS'),          31410000
227                  2 VSIO_RRDS            CHAR(4)   INIT('RRDS'),          31420000
228                  2 VSIO_SEQUENTIAL      CHAR(10)  INIT('SEQUENTIAL'),    31430000
229                  2 VSIO_DIRECT          CHAR(10)  INIT('DIRECT    '),    31440000
230                  2 VSIO_DYNAMIC         CHAR(10)  INIT('DYNAMIC   '),    31450000
231                  2 VSIO_INPUT           CHAR(6)   INIT('INPUT '),        31460000
232                  2 VSIO_OUTPUT          CHAR(6)   INIT('OUTPUT'),        31470000
233                  2 VSIO_INPUT_OUTPUT    CHAR(6)   INIT('UPDATE'),        31480000
234                  2 (VSIO_RC_SUCCESS                INIT(0),               31490000
235                     VSIO_RC_LOGIC_ERROR            INIT(8),               31500000
236                     VSIO_RC_END_OF_FILE            INIT(9999),            31510000
237                     VSIO_RC_UNKNOWN_COMMAND        INIT(20),              31520000
238                     VSIO_RC_DATASET_ALREADY_OPEN   INIT(21),              31530000
239                     VSIO_RC_DATASET_NOT_OPEN       INIT(22),              31540000
240                     VSIO_RC_ORGANIZATION_UNKNOWN   INIT(23),              31550000
241                     VSIO_RC_ACCESS_UNKNOWN         INIT(24),              31560000
242                     VSIO_RC_ORG_ACCESS_MISMATCH    INIT(25),              31570000
243                     VSIO_RC_MODE_UNKNOWN           INIT(26),              31580000
244                     VSIO_RC_MODE_UNSUPPORTED       INIT(27),              31590000
245                     VSIO_RC_DDNAME_BLANK           INIT(28))              31600000
246                                    FIXED BINARY(15,0),                    31610000
247                  2 (VSIO_FB_DUPLICATE_RECORD       INIT(8),               31620000
248                     VSIO_FB_KEY_SEQUENCE           INIT(12),              31630000
249                     VSIO_FB_RECORD_NOT_FOUND       INIT(16),              31640000
250                     VSIO_FB_NO_MORE_SPACE          INIT(28),              31650000
251                     VSIO_FB_READ_WITHOUT_START     INIT(88))              31660000
252                                    FIXED BINARY(15,0),                    31670000
253               /*****************************************************************31680000
254                 THE VSIO_PARAMETER_BLOCK IS THE COMMUNICATION INTERFACE TO THE  31690000
255                 THE ROUTINE.                                                    31700000
256                *****************************************************************/31710000
257                                                                               31720000
258               1 VSIO_PARAMETER_BLOCK    STATIC,                             31730000
259                  2 VSIO_COMMAND         CHAR(8)   INIT(' '),                31740000
260                  2 (VSIO_RETURN_CODE,                                       31750000
261                     VSIO_VSAM_RC,                                           31760000
262                     VSIO_VSAM_FUNCTION,                                     31770000
263                     VSIO_VSAM_FEEDBACK) FIXED BINARY(15,0) INIT(0);         31780000
264                                                                             31790000
265               /*****************************************************************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
      ROUTINE TO OPEN THE DATASET AND MUST NOT THEN BE CHANGED UNTIL THE  00002000
      DATASET HAS BEEN CLOSED.                                            00002100
    *********************************************************************/00002200
                                                                          00002300
         DECLARE                                                          00002400
            1 VSIO_FILE_BLOCK            STATIC,                          00002500
               2 VSFB_DDNAME             CHAR(8)   INIT(' '),             00002600
               2 VSFB_ORGANIZATION       CHAR(4)   INIT(' '),             00002700
               2 VSFB_ACCESS             CHAR(10)  INIT(' '),             00002800
               2 VSFB_MODE               CHAR(6)   INIT(' '),             00002900
               2 (VSFB_RECORD_LENGTH,                                     00003000
                  VSFB_KEY_POSITION,                                      00003100
                  VSFB_KEY_LENGTH)       FIXED BINARY(15,0) INIT(0),      00003200
               2 VSFB_FILE_STATUS        CHAR(1)   INIT('C'),             00003300
               2 VSFB_RESERVED           CHAR(161);                       00003400
                                                                          00003500
   /*********************************************************************00003600
                         END OF VSAMIOFB COPY BOOK                        00003700
    *********************************************************************/00003800
```

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

SOURCE LISTING.

```
                /*****************************************************************    1
                                                                                       1
                  KSDSSSEQ - TESTS THE VSAMIO ROUTINE BY USING START AND READ COMMANDS 1
                             ON AN INDEXED CLUTER TO READ THE FILE IN SKIP-SEQUENTIAL  1
                             MODE.                                                     1
                                                                                       1
                 *****************************************************************/    1
                                                                                       7
 1               KSDSSSQ:                                                              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 = 'KSDSSSEQ: READ KSDS SEQUENTIALLY (W/START)';     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
                   /*****************************************************************29
                     ESTABLISH PARAMETERS OF VSAM DATASET AND CALL ROUTINE TO OPEN IT  29
                    *****************************************************************/ 29
                                                                                      31
16                    VSFB_DDNAME = 'KSDSF01';                                       32
17                    VSFB_ORGANIZATION = VSIO_KSDS;                                 33
18                    VSFB_ACCESS = VSIO_SEQUENTIAL;                                 34
19                    VSFB_MODE = VSIO_INPUT;                                        35
20                    VSFB_RECORD_LENGTH = 80;                                       36
21                    VSFB_KEY_POSITION = 0;                                         37
22                    VSFB_KEY_LENGTH = 10;                                          38
23                    VSIO_COMMAND = VSIO_OPEN;                                      39
24                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                            40
                                    VSIO_FILE_BLOCK,                                 41
```

```
                         RECORD_IMAGE);
25                IF (VSIO_RETURN_CODE ª= VSIO_RC_SUCCESS) THEN
26                   DO;
27                      CALL VSIO_ERROR;
28                      RETURN;
29                   END;

30                RECORD_KEY = '1033846021';
31                VSIO_COMMAND = VSIO_START_EQUAL;
32                PRINT_AREA = 'STARTING KEY EQUAL TO ' || RECORD_KEY;
33                WRITE FILE(PRINTR) FROM(PRINT_LINE);
34                CALL PROCESS_BLOCK;

35                RECORD_KEY = '2534789096';
36                VSIO_COMMAND = VSIO_START_EQUAL;
37                PRINT_AREA = 'STARTING KEY EQUAL TO ' || RECORD_KEY;
38                WRITE FILE(PRINTR) FROM(PRINT_LINE);
39                CALL PROCESS_BLOCK;

40                RECORD_KEY = '3284189067';
41                VSIO_COMMAND = VSIO_START_NOTLESS;
42                PRINT_AREA = 'STARTING KEY NOT LESS THAN ' || RECORD_KEY;
43                WRITE FILE(PRINTR) FROM(PRINT_LINE);
44                CALL PROCESS_BLOCK;

45                RECORD_KEY = '3860000000';
46                VSIO_COMMAND = VSIO_START_NOTLESS;
47                PRINT_AREA = 'STARTING KEY NOT LESS THAN ' || RECORD_KEY;
48                WRITE FILE(PRINTR) FROM(PRINT_LINE);
49                CALL PROCESS_BLOCK;

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

50                VSIO_COMMAND = VSIO_CLOSE;
51                CALL VSAMIOP (VSIO_PARAMETER_BLOCK,
                         VSIO_FILE_BLOCK,
                         RECORD_IMAGE);
52                IF (VSIO_RETURN_CODE ª= VSIO_RC_SUCCESS) THEN
53                   CALL VSIO_ERROR;

54                RETURN;

55            PROCESS_BLOCK:
                 PROCEDURE;
```

```
56                   CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                          88
                                   VSIO_FILE_BLOCK,                             89
                                   RECORD_IMAGE);                               90
57                   IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                91
58                      IF (VSIO_VSAM_FEEDBACK = VSIO_FB_RECORD_NOT_FOUND) THEN   92
59                         DO;                                                    93
60                            PRINT_AREA = '*** NOT FOUND ***';                   94
61                            WRITE FILE(PRINTR) FROM(PRINT_LINE);                95
62                            RETURN;                                             96
63                         END;                                                   97
64                      ELSE                                                      98
64                         DO;                                                    99
65                            CALL VSIO_ERROR;                                   100
66                            RETURN;                                            101
67                         END;                                                  102
                                                                                 103
68                   VSIO_COMMAND = VSIO_READ;                                   104
69                   MORE_RECORDS = YES;                                         105
70                   RECORD_COUNTER = 0;                                         106
                                                                                 107
71                   DO WHILE(MORE_RECORDS & RECORD_COUNTER < 5);                108
72                      CALL READ_KS;                                            109
73                      IF (MORE_RECORDS) THEN                                   110
74                         DO;                                                   111
75                            PRINT_AREA = 'KEY: ' || RECORD_KEY || '  DATA: ' ||  112
                                           RECORD_IMAGE_SCALAR;                  113
76                            WRITE FILE(PRINTR) FROM(PRINT_LINE);               114
77                         END;                                                  115
78                      END;                                                     116
                                                                                 117
79                      RETURN;                                                  118
                                                                                 119
80                   END PROCESS_BLOCK;                                          120
                                                                                 121
81                READ_KS:                                                      122
                     PROCEDURE;                                                  123
                                                                                 124
                  /*****************************************************************  125
                    CALL ROUTINE TO READ NEXT RECORD FROM VSAM DATASET           125
                   *****************************************************************/  125
                                                                                 127
82                      VSIO_COMMAND = VSIO_READ;                                128
83                      CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                      129
                                      VSIO_FILE_BLOCK,                           130
                                      RECORD_IMAGE);                             131
84                      IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN            132
85                         IF (VSIO_RETURN_CODE = VSIO_RC_END_OF_FILE) THEN      133
```

```
 86                        MORE_RECORDS = NO;                                  134
 87                     ELSE                                                   135
 87                        CALL VSIO_ERROR;                                    136
 88                  ELSE                                                      137
 88                     RECORD_COUNTER = RECORD_COUNTER + 1;                   138
                                                                               139
 89                  RETURN;                                                   140
                                                                               141
 90               END READ_KS;                                                 142
                                                                               143
 91            VSIO_ERROR:                                                     144
                  PROCEDURE;                                                   145
 92                  PRINT_AREA = 'VSAMIO ERROR OCCURRED DURING ' ||          146
                                  VSIO_COMMAND;                                147
 93                  WRITE FILE(PRINTR) FROM(PRINT_LINE);                     148
 94                  PRINT_AREA = 'VSIO_RETURN_CODE = ' ||                    149
                                  VSIO_RETURN_CODE;                            150
 95                  WRITE FILE(PRINTR) FROM(PRINT_LINE);                     151
 96                  PRINT_AREA = 'VSIO_VSAM_RETURN_CODE = ' ||               152
                                  VSIO_VSAM_RETURN_CODE;                       153
 97                  WRITE FILE(PRINTR) FROM(PRINT_LINE);                     154
 98                  PRINT_AREA = 'VSIO_VSAM_FUNCTION_CODE = ' ||             155
                                  VSIO_VSAM_FUNCTION_CODE;                     156
 99                  WRITE FILE(PRINTR) FROM(PRINT_LINE);                     157
100                  PRINT_AREA = 'VSIO_VSAM_FEEDBACK_CODE = ' ||             158
                                  VSIO_VSAM_FEEDBACK_CODE;                     159
101                  WRITE FILE(PRINTR) FROM(PRINT_LINE);                     160
102                  PRINT_AREA = ' ';                                        161
                                                                              162
103                  RETURN;                                                  163
                                                                              164
104               END VSIO_ERROR;                                             165
                                                                              166
105               DECLARE                                                     167
                     PRINTR FILE OUTPUT RECORD SEQUENTIAL EXTERNAL            168
                        ENV(F CTLASA);                                        169
                                                                              170
106               DECLARE                                                     171
                     MORE_RECORDS                BIT(1),                     172
                     NO                          BIT(1)    INIT('0'B),       173
                     RECORD_COUNTER              FIXED BINARY(15,0) INIT(0), 174
                     YES                         BIT(1)    INIT('1'B);       175
                                                                              176
107               DECLARE                                                     177
                     1 RECORD_IMAGE,                                          178
                        2 RECORD_KEY             CHAR(10),                    179
                        2 RECORD_FIELDS          CHAR(70);                    180
```

```
                                                                                           181
108                  DECLARE                                                               182
                        RECORD_IMAGE_SCALAR             DEFINED RECORD_IMAGE               183
                                                        CHAR(80);                          184
                                                                                           185
109                  DECLARE                                                               186
                        1 PRINT_LINE,                                                      187
                           2 CARRIAGE_CONTROL           CHAR(1)   INIT(' '),               188
                           2 PRINT_AREA                 CHAR(132);                         189
                                                                                           190
                  /*********************************************************************  195
                                                                                           195
                          VV   VV    SSSSS       A      M     M    IIII      OOOOO         195
                          VV   VV   SS   SS     AAA     MM   MM     II      OO   OO        195
                          VV   VV   SS         AA AA    MMM MMM     II      OO   OO        195
                          VV   VV    SSSSS    AA    AA  MMMMMMM     II      OO   OO        195
                          VV   VV        SS   AA    AA  MM M MM     II      OO   OO        195
                           VV VV    SS   SS   AAAAAAA   MM   MM     II      OO   OO        195
                            VVV     SS   SS   AA    AA  MM   MM     II      OO   OO        195
                             V       SSSSS    AA    AA  MM   MM    IIII      OOOOO         195
                                                                                           195
                   **********************************************************************  195
                    THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS    195
                    ROUTINE.                                                               195
                                                                                           195
                    THE VSIO_PARAMETER_VALUES SUPPLY THE VALUES USED TO MOVE INTO          195
                    PARAMETER ENTRIES TO TAILOR THE ROUTINE TO A SPECIFIC DATASET AND      195
                    TO PROVIDE COMMANDS TO DRIVE THE ROUTINE.                              195
                   *********************************************************************/  195
                                                                                           213
110                        DECLARE                                                         215
                              1 VSIO_PARAMETER_VALUES   STATIC,                            216
                                 2 VSIO_OPEN            CHAR(8)   INIT('OPEN    '),        217
                                 2 VSIO_CLOSE           CHAR(8)   INIT('CLOSE   '),        218
                                 2 VSIO_READ            CHAR(8)   INIT('READ    '),        219
                                 2 VSIO_WRITE           CHAR(8)   INIT('WRITE   '),        220
                                 2 VSIO_REWRITE         CHAR(8)   INIT('REWRITE '),        221
                                 2 VSIO_DELETE          CHAR(8)   INIT('DELETE  '),        222
                                 2 VSIO_START_EQUAL     CHAR(8)   INIT('STARTEQ '),        223
                                 2 VSIO_START_NOTLESS   CHAR(8)   INIT('STARTGE '),        224
                                 2 VSIO_KSDS            CHAR(4)   INIT('KSDS'),            225
                                 2 VSIO_ESDS            CHAR(4)   INIT('ESDS'),            226
                                 2 VSIO_RRDS            CHAR(4)   INIT('RRDS'),            227
                                 2 VSIO_SEQUENTIAL      CHAR(10)  INIT('SEQUENTIAL'),      228
                                 2 VSIO_DIRECT          CHAR(10)  INIT('DIRECT    '),      229
                                 2 VSIO_DYNAMIC         CHAR(10)  INIT('DYNAMIC   '),      230
                                 2 VSIO_INPUT           CHAR(6)   INIT('INPUT '),          231
```

```
                        2 VSIO_OUTPUT          CHAR(6)   INIT('OUTPUT'),          232
                        2 VSIO_INPUT_OUTPUT    CHAR(6)   INIT('UPDATE'),          233
                        2 (VSIO_RC_SUCCESS                INIT(0),                 234
                           VSIO_RC_LOGIC_ERROR            INIT(8),                 235
                           VSIO_RC_END_OF_FILE            INIT(9999),              236
                           VSIO_RC_UNKNOWN_COMMAND        INIT(20),                237
                           VSIO_RC_DATASET_ALREADY_OPEN   INIT(21),                238
                           VSIO_RC_DATASET_NOT_OPEN       INIT(22),                239
                           VSIO_RC_ORGANIZATION_UNKNOWN   INIT(23),                240
                           VSIO_RC_ACCESS_UNKNOWN         INIT(24),                241
                           VSIO_RC_ORG_ACCESS_MISMATCH    INIT(25),                242
                           VSIO_RC_MODE_UNKNOWN           INIT(26),                243
                           VSIO_RC_MODE_UNSUPPORTED       INIT(27),                244
                           VSIO_RC_DDNAME_BLANK           INIT(28))                245
                                                  FIXED BINARY(15,0),               246
                        2 (VSIO_FB_DUPLICATE_RECORD       INIT(8),                 247
                           VSIO_FB_KEY_SEQUENCE           INIT(12),                248
                           VSIO_FB_RECORD_NOT_FOUND       INIT(16),                249
                           VSIO_FB_NO_MORE_SPACE          INIT(28),                250
                           VSIO_FB_READ_WITHOUT_START     INIT(88))                251
                                                  FIXED BINARY(15,0),               252
                     /*************************************************************  253
                       THE VSIO_PARAMETER_BLOCK IS THE COMMUNICATION INTERFACE TO THE  253
                       THE ROUTINE.                                                   253
                      *************************************************************/  253
                                                                                     256
                          1 VSIO_PARAMETER_BLOCK    STATIC,                        258
                          2 VSIO_COMMAND            CHAR(8)   INIT(' '),           259
                          2 (VSIO_RETURN_CODE,                                     260
                             VSIO_VSAM_RC,                                         261
                             VSIO_VSAM_FUNCTION,                                   262
                             VSIO_VSAM_FEEDBACK) FIXED BINARY(15,0) INIT(0);       263
                                                                                   264
                  /*****************************************************************  265
                                      END OF VSAMIO COPY BOOK                         265
                   *****************************************************************/  265
                  /*****************************************************************  268
                                                                                      268
                     VV    VV    SSSSS       A      M      M  IIII    OOOOO    FFFFFFF  BBBBBB     268
                     VV    VV   SS    SS    AAA     MM    MM   II    OO   OO  FF       BB    BB   268
                     VV    VV   SS         AA AA    MMM  MMM   II    OO   OO  FF       BB    BB   268
                     VV    VV    SSSSS    AA   AA   MMMMMMMM   II    OO   OO  FFFFF    BBBBBB     268
                     VV    VV        SS   AA   AA   MM M MM   II    OO   OO  FF       BB    BB   268
                      VV VV    SS    SS   AAAAAAA  MM   MM   II    OO   OO  FF       BB    BB   268
                       VVV     SS    SS   AA   AA  MM   MM   II    OO   OO  FF       BB    BB   268
                        V       SSSSS    AA   AA  MM   MM  IIII    OOOOO   FF       BBBBBB     268
                                                                                      268
```

```
                 ********************************************************************  268
                   THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS   268
                   ROUTINE, AND ARE USED TO COMMUNICATE CHARACTERISTICS FOR A SINGLE     268
                   VSAM DATASET.                                                         268
                                                                                         268
                   WITH THE 2 EXCEPTIONS FOR RECORD LENGTH (TO ACCOMODATE VARIABLE       268
                   LENGTH RECORDS) AND RELATIVE RECORD (TO ACCOMODATE RELATIVE RECORD    268
                   DATASETS), THESE DATA NAMES MUST BE POPULATED PRIOR TO CALLING THE    268
                   ROUTINE TO OPEN THE DATASET AND MUST NOT THEN BE CHANGED UNTIL THE    268
                   DATASET HAS BEEN CLOSED.                                              268
                 ********************************************************************/ 268
                                                                                         289
 111                  DECLARE                                                            291
                         1 VSIO_FILE_BLOCK            STATIC,                            292
                            2 VSFB_DDNAME             CHAR(8)   INIT(' '),               293
                            2 VSFB_ORGANIZATION       CHAR(4)   INIT(' '),               294
                            2 VSFB_ACCESS             CHAR(10)  INIT(' '),               295
                            2 VSFB_MODE               CHAR(6)   INIT(' '),               296
                            2 (VSFB_RECORD_LENGTH,                                       297
                               VSFB_KEY_POSITION,                                        298
                               VSFB_KEY_LENGTH)       FIXED BINARY(15,0) INIT(0),        299
                            2 VSFB_FILE_STATUS        CHAR(1)   INIT('C'),               300
                            2 VSFB_RESERVED           CHAR(161);                         301
                                                                                         302
                /*********************************************************************  303
                                        END OF VSAMIOFB COPY BOOK                        303
                 ********************************************************************/ 303
                                                                                         193
 112               END KSDSSSQ;                                                          194
```

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
109		CARRIAGE_CONTROL	IN PRINT_LINE,AUTOMATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER
1	*********	KSDSSSQ	ENTRY,BINARY,FIXED(15,0)
106		MORE_RECORDS	AUTOMATIC,UNALIGNED,STRING(1),BIT 69,71,73,86
106		NO	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 86
109		PRINT_AREA	IN PRINT_LINE,AUTOMATIC,UNALIGNED,STRING(132),CHARACTER 10,12,14,32,37,42,47,60,75,92,94,96,98,100,102
109		PRINT_LINE	AUTOMATIC,STRUCTURE 11,13,15,33,38,43,48,61,76,93,95,97,99,101
105		PRINTR	FILE,EXTERNAL,OUTPUT,RECORD,SEQUENTIAL,ENVIRONMENT(F CTLASA) 9,11,13,15,33,38,43,48,61,76,93,95,97,99,101
55		PROCESS_BLOCK	ENTRY,DECIMAL,FLOAT(SINGLE) 34,39,44,49
81		READ_KS	ENTRY,DECIMAL,FLOAT(SINGLE) 72
106	*********	RECORD_COUNTER	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 70,71,88,88
107		RECORD_FIELDS	IN RECORD_IMAGE,AUTOMATIC,UNALIGNED,STRING(70),CHARACTER
107		RECORD_IMAGE	AUTOMATIC,STRUCTURE 24,51,56,83
108		RECORD_IMAGE_SCALAR	AUTOMATIC,DEFINED,UNALIGNED,STRING(80),CHARACTER 75
107		RECORD_KEY	IN RECORD_IMAGE,AUTOMATIC,UNALIGNED,STRING(10),CHARACTER 30,32,35,37,40,42,45,47,75
		SYSPRINT	FILE,EXTERNAL 5,6,7

```
DCL NO.          IDENTIFIER                          ATTRIBUTES AND REFERENCES

                 VSAMIOP                             EXTERNAL,ENTRY,DECIMAL,FLOAT(SINGLE)
                                                     24,51,56,83
111              VSFB_ACCESS                         IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(10),CHARACTER
                                                     18
111              VSFB_DDNAME                         IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(8),CHARACTER
                                                     16
111              VSFB_FILE_STATUS                    IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER
111    ********* VSFB_KEY_LENGTH                     IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0)
                                                     22
111    ********* VSFB_KEY_POSITION                   IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0)
                                                     21
111              VSFB_MODE                           IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(6),CHARACTER
                                                     19
111              VSFB_ORGANIZATION                   IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(4),CHARACTER
                                                     17
111    ********* VSFB_RECORD_LENGTH                  IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0)
                                                     20
111              VSFB_RESERVED                       IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,STRING(161),CHARACTER
110              VSIO_CLOSE                          IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                     CHARACTER
                                                     50
110              VSIO_COMMAND                        IN VSIO_PARAMETER_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                     CHARACTER
                                                     23,31,36,41,46,50,68,82,92
110              VSIO_DELETE                         IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                     CHARACTER
110              VSIO_DIRECT                         IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10),
                                                     CHARACTER
110              VSIO_DYNAMIC                        IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10),
                                                     CHARACTER
```

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
91		VSIO_ERROR	ENTRY,DECIMAL,FLOAT(SINGLE) 27,53,65,87
110		VSIO_ESDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER
110	*********	VSIO_FB_DUPLICATE_RECORD	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_FB_KEY_SEQUENCE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_FB_NO_MORE_SPACE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_FB_READ_WITHOUT_START	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_FB_RECORD_NOT_FOUND	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 58
111		VSIO_FILE_BLOCK	STATIC,STRUCTURE 24,51,56,83
110		VSIO_INPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER 19
110		VSIO_INPUT_OUTPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER
110		VSIO_KSDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER 17
110		VSIO_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 23
110		VSIO_OUTPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER
110		VSIO_PARAMETER_BLOCK	STATIC,STRUCTURE

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
			24,51,56,83
110		VSIO_PARAMETER_VALUES	STATIC,STRUCTURE
110	*********	VSIO_RC_ACCESS_UNKNOWN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_RC_DATASET_ALREADY_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_RC_DATASET_NOT_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_RC_DDNAME_BLANK	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_RC_END_OF_FILE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 85
110	*********	VSIO_RC_LOGIC_ERROR	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_RC_MODE_UNKNOWN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_RC_MODE_UNSUPPORTED	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_RC_ORG_ACCESS_MISMATCH	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_RC_ORGANIZATION_UNKNOWN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110	*********	VSIO_RC_SUCCESS	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 25,52,57,84
110	*********	VSIO_RC_UNKNOWN_COMMAND	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
110		VSIO_READ	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 68,82

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
110	*********	VSIO_RETURN_CODE	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 25,52,57,84,85,94
110		VSIO_REWRITE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
110		VSIO_RRDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER
110		VSIO_SEQUENTIAL	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER 18
110		VSIO_START_EQUAL	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 31,36
110		VSIO_START_NOTLESS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 41,46
110	*********	VSIO_VSAM_FEEDBACK	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 58
		VSIO_VSAM_FEEDBACK_CODE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 100
110	*********	VSIO_VSAM_FUNCTION	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
		VSIO_VSAM_FUNCTION_CODE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 98
110	*********	VSIO_VSAM_RC	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
		VSIO_VSAM_RETURN_CODE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 96
110		VSIO_WRITE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
106	YES	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 69

AGGREGATE LENGTH TABLE

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

STORAGE REQUIREMENTS.

THE STORAGE AREA FOR THE PROCEDURE LABELLED KSDSSSQ IS 504 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 PROCESS_BLOCK IS 268 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 VSIO_ERROR IS 256 BYTES LONG.

THE PROGRAM CSECT IS NAMED KSDSSSQ AND IS 2094 BYTES LONG.

THE STATIC CSECT IS NAMED KSDSSSQA AND IS 5920 BYTES LONG.

STATISTICS MACRO RECORDS = 305,SOURCE RECORDS = 307,PROG TEXT STMNTS = 112,OBJECT BYTES = 2094

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 PROCESS_BLOCK

OFFSET (HEX)	0000	0048	0068	0074	0080	0080	0090	00A8	00AE	00B2	00B2	00BC	00C2	00C6	00CC	00D6	00DC	010C	0116	0122	0122
STATEMENT NO	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75

OFFSET (HEX)	014E	0166	0166	016A	0170
STATEMENT NO	76	77	78	79	80

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE READ_KS

OFFSET (HEX)	0000	0034	003A	005A	0066	0072	0080	008E	009E	00A4
STATEMENT NO	81	82	83	84	85	86	87	88	89	90

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	91	92	93	94	95	96	97	98	99	100	101	102	103	104

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE KSDSSSQ

OFFSET (HEX)	0000	00CC	00DA	00E4	00F0	0108	0114	012C	0138	0150	0156	015C	0162	0168	016E	0174	017A	0180	019C	01A8	01A8
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)	01B2	01B8	01B8	01BE	01C4	01DA	01F2	01FC	0202	0208	021E	0236	0240	0246	024C	0262	027A	0284	028A	0290	02A6
STATEMENT NO	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48

OFFSET (HEX)	02BE	02C8	02CE	02EA	02F6	0300	0306
STATEMENT NO	49	50	51	52	53	54	112

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 96, 98, 100.

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.
```

```
KSDSSSEQ: READ KSDS SEQUENTIALLY (W/START)
------------------------------------------

STARTING KEY EQUAL TO 1033846021
KEY: 1033846021  DATA: 1033846021 CHERYL I TUCKER      660 SHORE ROAD             LOUISVILLE      KY
KEY: 1098019022  DATA: 1098019022 HANNAH F QUIMBY      6151 MAIN COURT            PHOENIX         AZ
KEY: 1168050023  DATA: 1168050023 TAMMY J FRANKLIN     5226 ROSA LINDA ROAD       LOUISVILLE      KY
KEY: 1371074024  DATA: 1371074024 TAMMY M HARMON       243 KINGS RIDGE STREET     GREENVILLE      SC
KEY: 1442146025  DATA: 1442146025 LUCY H ERICKSON      6207 RIDGEWAY AVENUE       KANSAS CITY     MO
STARTING KEY EQUAL TO 2534789096
*** NOT FOUND ***
STARTING KEY NOT LESS THAN 3284189067
KEY: 3284189067  DATA: 3284189067 RITA N RENFRO        7881 DAISY COURT           SIOUX FALLS     SD
KEY: 3489628068  DATA: 3489628068 ARNOLD R ELISON      465 MAIN AVENUE            TULSA           OK
KEY: 3775212069  DATA: 3775212069 HANNAH J GLOVER      3791 SEABREEZE STREET      NASHAU          NH
KEY: 3786163070  DATA: 3786163070 REBECCA D DOREN      8730 STRAWBERRY ROAD       SPRING          TX
KEY: 3851331081  DATA: 3851331081 CLIFF J DRAKE        169 SHORE AVENUE           CHICAGO         IL
STARTING KEY NOT LESS THAN 3860000000
KEY: 3912384082  DATA: 3912384082 CRAIG O LABROIE      8021 MILL MOUNTAIN PLACE   MURFREESBORO    TN
KEY: 4077702083  DATA: 4077702083 ROLAND P RODGERS     827 MEADOW STREET          TULSA           OK
KEY: 4093285084  DATA: 4093285084 JACK J SCHWAB        250 BUCKLEY PLACE          CHICAGO         IL
KEY: 4176588085  DATA: 4176588085 BRENDA O MCKAY       315 A & M PLACE            LA HABRA        CA
KEY: 4197550086  DATA: 4197550086 TROY S POWERS        1707 BRIDGE STREET         ENGLEWOOD       CO
```