J E S 2 J O B L O G

```
18.04.59 JOB  147  IEF677I WARNING MESSAGE(S) FOR JOB VSTESTE4 ISSUED
18.04.59 JOB  147  $HASP373 VSTESTE4 STARTED - INIT  1 - CLASS A - SYS HMVS
18.04.59 JOB  147  IEF403I VSTESTE4 - STARTED - TIME=18.04.59
18.05.00 JOB  147  CCI001C PL1L    /IEMAA   /00:00:00.17/        /00004/1       /VSTESTE4
18.05.00 JOB  147  CCI001C LKED    /IEWL    /00:00:00.04/        /00000/1       /VSTESTE4
18.05.00 JOB  147  CCI001C GO      /PGM=*.DD/00:00:00.01/        /00000/1       /VSTESTE4
18.05.00 JOB  147  IEF404I VSTESTE4 - ENDED - TIME=18.05.00
18.05.00 JOB  147  $HASP395 VSTESTE4 ENDED

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

 07 JUL 20 JOB EXECUTION DATE

       22 CARDS READ

    1,165 SYSOUT PRINT RECORDS

        0 SYSOUT PUNCH RECORDS

     0.00 MINUTES EXECUTION TIME
```

```
 1      //VSTESTE4 JOB 1,'VSAMIOP IVP ESDSUPDT',CLASS=A,MSGCLASS=X,             JOB  147
        //             REGION=4096K
        ***
        ********************************************************************
        *** PL/1 MODULE: ESDSUPDT     VSAM DATASET: VSTESTES.CLUSTER (ESDS)
        ***
        *** SEQUENTIALLY READS/REWRITES ENTRY SEQUENCED 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(ESDSUPDT),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.ESDSF01 DD DSN=PUB001.VSTESTES.CLUSTER,DISP=OLD
```

```
 STMT NO. MESSAGE
-
   19     IEF686I DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
IEF236I ALLOC. FOR VSTESTE4 PL1L PL1F
IEF237I 253  ALLOCATED TO STEPLIB
IEF237I 253  ALLOCATED TO SYS00310
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 VSTESTE4 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.JOB00147.SO0101                        SYSOUT
IEF285I   SYS20189.T180459.RA000.VSTESTE4.LOADSET     PASSED        *------223
IEF285I   VOL SER NOS= MVS380.
IEF285I   SYS20189.T180459.RA000.VSTESTE4.SYSUT3      DELETED       *------303
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T180459.RA000.VSTESTE4.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 20189.1804
IEF374I STEP /PL1L    / STOP  20189.1805 CPU    0MIN 00.17SEC SRB    0MIN 00.04SEC VIRT  4096K SYS   212K
**** JOBCARD READ 20189 18:04:59 ************************************************************************************
*                                      PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                              *
*  STEP NAME  PL1L        USER CORE        4096K   TAPES USED/IO 000/000000000   START   TIME  18:04:59    TCB TIME  00:00:00.17  *
*  PGM  NAME  IEMAA       SYSTEM CORE       212K   DISKS USED/IO 004/000000556   STOP    TIME  18:05:00    SRB TIME  00:00:00.04  *
*   COND CODE    0004     PRIVATE AREA SZ   4096K       ALLOC TIME  18:04:59      ELAPSED TIME                PGM LOAD  18:04:59   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004          3013   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 VSTESTE4 LKED PL1F
IEF237I 253  ALLOCATED TO SYSLIB
IEF237I 253  ALLOCATED TO
IEF237I 253  ALLOCATED TO SYS00312
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 VSTESTE4 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.T180459.RA000.VSTESTE4.GOSET       PASSED        *-------57
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T180459.RA000.VSTESTE4.SYSUT1      DELETED       *--------0
IEF285I   VOL SER NOS= MVS370.
IEF285I   JES2.JOB00147.SO0102                        SYSOUT
```

```
IEF285I   SYS20189.T180459.RA000.VSTESTE4.LOADSET      DELETED       *------224
IEF285I   VOL SER NOS= MVS380.
IEF373I STEP /LKED    / START 20189.1805
IEF374I STEP /LKED    / STOP  20189.1805 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  18:05:00    TCB TIME  00:00:00.04  *
*  PGM  NAME  IEWL        SYSTEM CORE        208K   DISKS USED/IO 004/000000387   STOP    TIME  18:05:00    SRB TIME  00:00:00.01  *
*   COND CODE    0000     PRIVATE AREA SZ   4096K       ALLOC TIME  18:05:00      ELAPSED TIME                PGM LOAD  18:05:00   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004          1978   00:00:00.06            0             0         0              0              0             0  *
********************************************************************************************************************
*  CPU $ (     0.01) + EXCP $ (     0.52) + MEMORY $ (     0.02) = TOTAL $ (     0.55)                                *
********************************************************************************************************************
IEF236I ALLOC. FOR VSTESTE4 GO PL1F
IEF237I 251  ALLOCATED TO PGM=*.DD
IEF237I 253  ALLOCATED TO STEPLIB
IEF237I 253  ALLOCATED TO
IEF237I 253  ALLOCATED TO SYS00314
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO PRINTR
IEF237I JES2 ALLOCATED TO SYSUDUMP
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I 190  ALLOCATED TO ESDSF01
IEF237I 190  ALLOCATED TO SYS00316
IEF142I VSTESTE4 GO PL1F - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS20189.T180459.RA000.VSTESTE4.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.JOB00147.SO0103                         SYSOUT
IEF285I   JES2.JOB00147.SO0104                         SYSOUT
IEF285I   JES2.JOB00147.SO0105                         SYSOUT
IEF285I   JES2.JOB00147.SO0106                         SYSOUT
IEF285I   PUB001.VSTESTES.CLUSTER                      KEPT          *--------2
IEF285I   VOL SER NOS= PUB001.
IEF285I   UCPUB001                                     KEPT          *--------0
IEF285I   VOL SER NOS= PUB001.
IEF373I STEP /GO      / START 20189.1805
IEF374I STEP /GO      / STOP  20189.1805 CPU    0MIN 00.01SEC SRB    0MIN 00.00SEC VIRT    96K SYS   220K
********************************************************************************************************************
*                                        PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                             *
*  STEP NAME  GO          USER CORE           96K   TAPES USED/IO 000/000000000   START   TIME  18:05:00    TCB TIME  00:00:00.01  *
*  PGM  NAME  PGM=*.DD    SYSTEM CORE        220K   DISKS USED/IO 003/000000002   STOP    TIME  18:05:00    SRB TIME  00:00:00.00  *
*   COND CODE    0000     PRIVATE AREA SZ   4096K       ALLOC TIME  18:05:00      ELAPSED TIME                PGM LOAD  18:05:00   *
** 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.T180500.RA000.VSTESTE4.R0000001     KEPT          *--------0
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T180459.RA000.VSTESTE4.GOSET        DELETED
IEF285I   VOL SER NOS= WORK00.
IEF375I  JOB /VSTESTE4/ START 20189.1804
IEF376I  JOB /VSTESTE4/ STOP  20189.1805 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    /*****************************************************************04660000
  2                                                                      04670000
  3      ESDSUPDT - TESTS THE VSAMIO ROUTINE BY READING RECORDS FROM AN ESDS  04680000
  4                 CLUSTER SEQUENTIALLY AND THEN REWRITING SOME OF THE       04690000
  5                 RECORDS WITH UPDATED CONTENTS.                            04700000
  6                                                                      04710000
  7     *****************************************************************/04720000
  8    ESDSUPD:                                                          04730000
  9       PROCEDURE OPTIONS(MAIN);                                       04740000
 10                                                                      04750000
 11          ON ERROR                                                    04760000
 12             BEGIN;                                                   04770000
 13                ON ERROR SYSTEM;                                      04780000
 14                PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*');     04790000
 15                PUT SKIP DATA;                                        04800000
 16                PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*');     04810000
 17             END;                                                     04820000
 18                                                                      04830000
 19          OPEN                                                        04840000
 20             FILE(PRINTR) LINESIZE(121);                              04850000
 21                                                                      04860000
 22          PRINT_AREA = 'ESDSUPDT: READ/REWRITE ESDS SEQUENTIALLY';    04870000
 23          WRITE FILE(PRINTR) FROM(PRINT_LINE);                        04880000
 24          PRINT_AREA = '----------------------------------------';    04890000
 25          WRITE FILE(PRINTR) FROM(PRINT_LINE);                        04900000
 26          PRINT_AREA = ' ';                                           04910000
 27          WRITE FILE(PRINTR) FROM(PRINT_LINE);                        04920000
 28                                                                      04930000
 29          MORE_RECORDS = YES;                                         04940000
 30                                                                      04950000
 31     /*****************************************************************04960000
 32       ESTABLISH PARAMETERS OF VSAM DATASET AND CALL ROUTINE TO OPEN IT     04970000
 33      *****************************************************************/04980000
 34          VSFB_DDNAME = 'ESDSF01';                                    04990000
 35          VSFB_ORGANIZATION = VSIO_ESDS;                              05000000
 36          VSFB_ACCESS = VSIO_SEQUENTIAL;                              05010000
 37          VSFB_MODE = VSIO_INPUT_OUTPUT;                              05020000
 38          VSFB_RECORD_LENGTH = 80;                                    05030000
 39          VSFB_KEY_POSITION = 0;                                      05040000
 40          VSFB_KEY_LENGTH = 0;                                        05050000
 41          VSIO_COMMAND = VSIO_OPEN;                                   05060000
 42          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                         05070000
 43                        VSIO_FILE_BLOCK,                              05080000
 44                        RECORD_IMAGE);                                05090000
```

MACRO SOURCE2 LISTING

```
 45          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                   05100000
 46             DO;                                                          05110000
 47                CALL VSIO_ERROR;                                          05120000
 48                RETURN;                                                   05130000
 49             END;                                                         05140000
 50                                                                          05150000
 51          DO WHILE(MORE_RECORDS);                                         05160000
 52             CALL READ_ES;                                                05170000
 53             IF (MORE_RECORDS) THEN                                       05180000
 54                IF (RECORD_IDENT_LOW = '02' |                             05190000
 55                    RECORD_IDENT_LOW = '06' |                             05200000
 56                    RECORD_IDENT_LOW = '07') THEN                         05210000
 57                   CALL UPDATE_RECORD;                                    05220000
 58          END;                                                            05230000
 59                                                                          05240000
 60    /*****************************************************************05250000
 61      CALL ROUTINE TO CLOSE VSAM DATASET                                  05260000
 62     *****************************************************************/05270000
 63          VSIO_COMMAND = VSIO_CLOSE;                                      05280000
 64          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                             05290000
 65                        VSIO_FILE_BLOCK,                                  05300000
 66                        RECORD_IMAGE);                                    05310000
 67          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                   05320000
 68             CALL VSIO_ERROR;                                             05330000
 69                                                                          05340000
 70          RETURN;                                                         05350000
 71                                                                          05360000
 72    UPDATE_RECORD:                                                        05370000
 73       PROCEDURE;                                                         05380000
 74                                                                          05390000
 75          COUNTER_EDIT = RECORD_COUNTER;                                  05400000
 76          PRINT_AREA = COUNTER_EDIT || ': ' ||                            05410000
 77                       RECORD_IMAGE_SCALAR || '  BEFORE';                05420000
 78          WRITE FILE(PRINTR) FROM(PRINT_LINE);                            05430000
 79                                                                          05440000
 80          RECORD_IDENT_HI = '99';                                         05450000
 81          CALL REWRITE_ES;                                                05460000
 82                                                                          05470000
 83          IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN                    05480000
 84             DO;                                                          05490000
 85                PRINT_AREA = '            ' ||                           05500000
 86                             RECORD_IMAGE_SCALAR || '  AFTER';           05510000
 87                WRITE FILE(PRINTR) FROM(PRINT_LINE);                      05520000
 88             END;                                                         05530000
 89                                                                          05540000
```

MACRO SOURCE2 LISTING

```
 90          RETURN;                                                  05550000
 91                                                                   05560000
 92       END UPDATE_RECORD;                                          05570000
 93                                                                   05580000
 94    READ_ES:                                                       05590000
 95       PROCEDURE;                                                  05600000
 96                                                                   05610000
 97    /*****************************************************************05620000
 98      CALL ROUTINE TO READ NEXT RECORD FROM VSAM DATASET           05630000
 99     *****************************************************************/05640000
100          VSIO_COMMAND = VSIO_READ;                                05650000
101          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                      05660000
102                        VSIO_FILE_BLOCK,                           05670000
103                        RECORD_IMAGE);                             05680000
104          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN            05690000
105             IF (VSIO_RETURN_CODE = VSIO_RC_END_OF_FILE) THEN      05700000
106                MORE_RECORDS = NO;                                 05710000
107             ELSE                                                  05720000
108                CALL VSIO_ERROR;                                   05730000
109          ELSE                                                     05740000
110             RECORD_COUNTER = RECORD_COUNTER + 1;                  05750000
111                                                                   05760000
112          RETURN;                                                  05770000
113                                                                   05780000
114       END READ_ES;                                                05790000
115                                                                   05800000
116    REWRITE_ES:                                                    05810000
117       PROCEDURE;                                                  05820000
118                                                                   05830000
119    /*****************************************************************05840000
120      CALL ROUTINE TO REWRITE PREVIOUSLY READ RECORD TO VSAM DATASET05850000
121     *****************************************************************/05860000
122          VSIO_COMMAND = VSIO_REWRITE;                             05870000
123          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                      05880000
124                        VSIO_FILE_BLOCK,                           05890000
125                        RECORD_IMAGE);                             05900000
126          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN            05910000
127             CALL VSIO_ERROR;                                      05920000
128                                                                   05930000
129          RETURN;                                                  05940000
130                                                                   05950000
131       END REWRITE_ES;                                             05960000
132                                                                   05970000
133    VSIO_ERROR:                                                    05980000
134       PROCEDURE;                                                  05990000
```

MACRO SOURCE2 LISTING

```
135          PRINT_AREA = 'VSAMIO ERROR OCCURRED DURING ' ||                 06000000
136                       VSIO_COMMAND;                                      06010000
137          WRITE FILE(PRINTR) FROM(PRINT_LINE);                            06020000
138          PRINT_AREA = 'VSIO_RETURN_CODE = ' ||                           06030000
139                       VSIO_RETURN_CODE;                                  06040000
140          WRITE FILE(PRINTR) FROM(PRINT_LINE);                            06050000
141          PRINT_AREA = 'VSIO_VSAM_RETURN_CODE = ' ||                      06060000
142                       VSIO_VSAM_RETURN_CODE;                             06070000
143          WRITE FILE(PRINTR) FROM(PRINT_LINE);                            06080000
144          PRINT_AREA = 'VSIO_VSAM_FUNCTION_CODE = ' ||                    06090000
145                       VSIO_VSAM_FUNCTION_CODE;                           06100000
146          WRITE FILE(PRINTR) FROM(PRINT_LINE);                            06110000
147          PRINT_AREA = 'VSIO_VSAM_FEEDBACK_CODE = ' ||                    06120000
148                       VSIO_VSAM_FEEDBACK_CODE;                           06130000
149          WRITE FILE(PRINTR) FROM(PRINT_LINE);                            06140000
150          PRINT_AREA = ' ';                                               06150000
151                                                                          06160000
152          RETURN;                                                         06170000
153                                                                          06180000
154       END VSIO_ERROR;                                                    06190000
155                                                                          06200000
156       DECLARE                                                            06210000
157          PRINTR FILE OUTPUT RECORD SEQUENTIAL EXTERNAL                   06220000
158             ENV(F CTLASA);                                               06230000
159                                                                          06240000
160       DECLARE                                                            06250000
161          COUNTER_EDIT                  PICTURE 'ZZ,ZZZ,ZZ9V',          06260000
162          MORE_RECORDS                  BIT(1),                         06270000
163          NO                            BIT(1)    INIT('0'B),           06280000
164          RECORD_COUNTER                FIXED BINARY(15,0),             06290000
165          YES                           BIT(1)    INIT('1'B);           06300000
166                                                                          06310000
167       DECLARE                                                            06320000
168          1 RECORD_IMAGE,                                                 06330000
169             2 RECORD_IDENT,                                              06340000
170                3 RECORD_IDENT_HI        CHAR(2),                         06350000
171                3 RECORD_IDENT_MID       CHAR(6),                         06360000
172                3 RECORD_IDENT_LOW       CHAR(2),                         06370000
173             2 RECORD_FIELDS             CHAR(70);                        06380000
174                                                                          06390000
175       DECLARE                                                            06400000
176          RECORD_IMAGE_SCALAR           DEFINED RECORD_IMAGE             06410000
177                                        CHAR(80);                        06420000
178                                                                          06430000
179       DECLARE                                                            06440000
```

MACRO SOURCE2 LISTING

```
180          1 PRINT_LINE,                                                 06450000
181             2 CARRIAGE_CONTROL          CHAR(1)   INIT(' '),           06460000
182             2 PRINT_AREA                CHAR(120);                     06470000
183                                                                        06480000
184     %INCLUDE (VSAMIO);                                                 06490000
185     %INCLUDE (VSAMIOFB);                                               06500000
186                                                                        06510000
187        END ESDSUPD;                                                    06520000
188                                                                        06530000
```

INCLUDED TEXT FOLLOWS FROM DD.MEMBER = SYSLIB .VSAMIO

```
189     /*****************************************************************31100000
190                                                                        31110000
191             VV   VV    SSSSS       A      M     M    IIII      OOOOO       31120000
192             VV   VV   SS   SS     AAA     MM   MM     II      OO   OO      31130000
193             VV   VV   SS         AA AA    MMM MMM     II      OO   OO      31140000
194             VV   VV    SSSSS    AA   AA   MMMMMMM     II      OO   OO      31150000
195             VV   VV        SS   AA   AA   MM M MM     II      OO   OO      31160000
196              VV VV    SS   SS   AAAAAAA   MM   MM     II      OO   OO      31170000
197               VVV     SS   SS   AA   AA   MM   MM     II      OO   OO      31180000
198                V       SSSSS    AA   AA   MM   MM    IIII      OOOOO       31190000
199                                                                        31200000
200      ******************************************************************31210000
201      THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS  31220000
202      ROUTINE.                                                          31230000
203                                                                        31240000
204      THE VSIO_PARAMETER_VALUES SUPPLY THE VALUES USED TO MOVE INTO     31250000
205      PARAMETER ENTRIES TO TAILOR THE ROUTINE TO A SPECIFIC DATASET AND  31260000
206      TO PROVIDE COMMANDS TO DRIVE THE ROUTINE.                         31270000
207     ******************************************************************/31280000
208                                                                        31290000
209             DECLARE                                                    31300000
210                1 VSIO_PARAMETER_VALUES   STATIC,                       31310000
211                   2 VSIO_OPEN            CHAR(8)   INIT('OPEN    '),   31320000
212                   2 VSIO_CLOSE           CHAR(8)   INIT('CLOSE   '),   31330000
213                   2 VSIO_READ            CHAR(8)   INIT('READ    '),   31340000
214                   2 VSIO_WRITE           CHAR(8)   INIT('WRITE   '),   31350000
215                   2 VSIO_REWRITE         CHAR(8)   INIT('REWRITE '),   31360000
216                   2 VSIO_DELETE          CHAR(8)   INIT('DELETE  '),   31370000
217                   2 VSIO_START_EQUAL     CHAR(8)   INIT('STARTEQ '),   31380000
218                   2 VSIO_START_NOTLESS   CHAR(8)   INIT('STARTGE '),   31390000
219                   2 VSIO_KSDS            CHAR(4)   INIT('KSDS'),       31400000
220                   2 VSIO_ESDS            CHAR(4)   INIT('ESDS'),       31410000
```

MACRO SOURCE2 LISTING

```
221                  2 VSIO_RRDS            CHAR(4)   INIT('RRDS'),          31420000
222                  2 VSIO_SEQUENTIAL      CHAR(10)  INIT('SEQUENTIAL'),    31430000
223                  2 VSIO_DIRECT          CHAR(10)  INIT('DIRECT    '),    31440000
224                  2 VSIO_DYNAMIC         CHAR(10)  INIT('DYNAMIC   '),    31450000
225                  2 VSIO_INPUT           CHAR(6)   INIT('INPUT '),        31460000
226                  2 VSIO_OUTPUT          CHAR(6)   INIT('OUTPUT'),        31470000
227                  2 VSIO_INPUT_OUTPUT    CHAR(6)   INIT('UPDATE'),        31480000
228                  2 (VSIO_RC_SUCCESS                INIT(0),              31490000
229                     VSIO_RC_LOGIC_ERROR            INIT(8),              31500000
230                     VSIO_RC_END_OF_FILE            INIT(9999),           31510000
231                     VSIO_RC_UNKNOWN_COMMAND        INIT(20),             31520000
232                     VSIO_RC_DATASET_ALREADY_OPEN   INIT(21),             31530000
233                     VSIO_RC_DATASET_NOT_OPEN       INIT(22),             31540000
234                     VSIO_RC_ORGANIZATION_UNKNOWN   INIT(23),             31550000
235                     VSIO_RC_ACCESS_UNKNOWN         INIT(24),             31560000
236                     VSIO_RC_ORG_ACCESS_MISMATCH    INIT(25),             31570000
237                     VSIO_RC_MODE_UNKNOWN           INIT(26),             31580000
238                     VSIO_RC_MODE_UNSUPPORTED       INIT(27),             31590000
239                     VSIO_RC_DDNAME_BLANK           INIT(28))             31600000
240                                          FIXED BINARY(15,0),             31610000
241                  2 (VSIO_FB_DUPLICATE_RECORD       INIT(8),              31620000
242                     VSIO_FB_KEY_SEQUENCE           INIT(12),             31630000
243                     VSIO_FB_RECORD_NOT_FOUND       INIT(16),             31640000
244                     VSIO_FB_NO_MORE_SPACE          INIT(28),             31650000
245                     VSIO_FB_READ_WITHOUT_START     INIT(88))             31660000
246                                          FIXED BINARY(15,0),             31670000
247               /*****************************************************************31680000
248                 THE VSIO_PARAMETER_BLOCK IS THE COMMUNICATION INTERFACE TO THE  31690000
249                 THE ROUTINE.                                                    31700000
250                *****************************************************************/31710000
251                                                                                 31720000
252                1 VSIO_PARAMETER_BLOCK    STATIC,                             31730000
253                   2 VSIO_COMMAND         CHAR(8)   INIT(' '),                31740000
254                   2 (VSIO_RETURN_CODE,                                         31750000
255                      VSIO_VSAM_RC,                                             31760000
256                      VSIO_VSAM_FUNCTION,                                       31770000
257                      VSIO_VSAM_FEEDBACK) FIXED BINARY(15,0) INIT(0);           31780000
258                                                                                31790000
259               /*****************************************************************31800000
260                                 END OF VSAMIO COPY BOOK                         31810000
261                *****************************************************************/31820000
```

INCLUDED TEXT FOLLOWS FROM DD.MEMBER = SYSLIB .VSAMIOFB

MACRO SOURCE2 LISTING

```
262    /*****************************************************************00000100
263                                                                       00000200
264       VV   VV   SSSSS      A     M     M  IIII   OOOOO   FFFFFFF  BBBBBB  00000300
265       VV   VV  SS   SS    AAA    MM   MM   II   OO   OO  FF       BB   BB 00000400
266       VV   VV  SS        AA AA   MMM MMM   II   OO   OO  FF       BB   BB 00000500
267       VV   VV   SSSSS   AA   AA  MMMMMMM   II   OO   OO  FFFFF    BBBBBB  00000600
268       VV   VV       SS  AA   AA  MM M MM   II   OO   OO  FF       BB   BB 00000700
269        VV VV   SS   SS  AAAAAAA  MM   MM   II   OO   OO  FF       BB   BB 00000800
270         VVV    SS   SS  AA   AA  MM   MM   II   OO   OO  FF       BB   BB 00000900
271          V      SSSSS   AA   AA  MM   MM  IIII   OOOOO   FF       BBBBBB  00001000
272                                                                       00001100
273     *****************************************************************00001200
274       THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS 00001300
275       ROUTINE, AND ARE USED TO COMMUNICATE CHARACTERISTICS FOR A SINGLE   00001400
276       VSAM DATASET.                                                       00001500
277                                                                           00001600
278       WITH THE 2 EXCEPTIONS FOR RECORD LENGTH (TO ACCOMODATE VARIABLE     00001700
279       LENGTH RECORDS) AND RELATIVE RECORD (TO ACCOMODATE RELATIVE RECORD  00001800
280       DATASETS), THESE DATA NAMES MUST BE POPULATED PRIOR TO CALLING THE  00001900
281       ROUTINE TO OPEN THE DATASET AND MUST NOT THEN BE CHANGED UNTIL THE  00002000
282       DATASET HAS BEEN CLOSED.                                            00002100
283     *****************************************************************/00002200
284                                                                           00002300
285          DECLARE                                                        00002400
286             1 VSIO_FILE_BLOCK            STATIC,                        00002500
287               2 VSFB_DDNAME              CHAR(8)   INIT(' '),           00002600
288               2 VSFB_ORGANIZATION        CHAR(4)   INIT(' '),           00002700
289               2 VSFB_ACCESS              CHAR(10)  INIT(' '),           00002800
290               2 VSFB_MODE                CHAR(6)   INIT(' '),           00002900
291               2 (VSFB_RECORD_LENGTH,                                    00003000
292                  VSFB_KEY_POSITION,                                     00003100
293                  VSFB_KEY_LENGTH)        FIXED BINARY(15,0) INIT(0),    00003200
294               2 VSFB_FILE_STATUS         CHAR(1)   INIT('C'),           00003300
295               2 VSFB_RESERVED            CHAR(161);                     00003400
296                                                                         00003500
297     /*****************************************************************00003600
298                            END OF VSAMIOFB COPY BOOK                    00003700
299      *****************************************************************/00003800
```

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

SOURCE LISTING.

```
                /*****************************************************************    1
                                                                                       1
                  ESDSUPDT - TESTS THE VSAMIO ROUTINE BY READING RECORDS FROM AN ESDS   1
                             CLUSTER SEQUENTIALLY AND THEN REWRITING SOME OF THE        1
                             RECORDS WITH UPDATED CONTENTS.                            1
                                                                                       1
                 *****************************************************************/    1
                                                                                       7
 1               ESDSUPD:                                                              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(121);                              20
                                                                                     21
10                     PRINT_AREA = 'ESDSUPDT: READ/REWRITE ESDS 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 = 'ESDSF01';                                    34
18                     VSFB_ORGANIZATION = VSIO_ESDS;                              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 = 0;                                        40
24                     VSIO_COMMAND = VSIO_OPEN;                                   41
```

```
25                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                        42
                                    VSIO_FILE_BLOCK,                             43
                                    RECORD_IMAGE);                               44
26                    IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN              45
27                       DO;                                                     46
28                          CALL VSIO_ERROR;                                     47
29                          RETURN;                                              48
30                       END;                                                    49
                                                                                 50
31                    DO WHILE(MORE_RECORDS);                                    51
32                       CALL READ_ES;                                           52
33                       IF (MORE_RECORDS) THEN                                  53
34                          IF (RECORD_IDENT_LOW = '02' |                        54
                                RECORD_IDENT_LOW = '06' |                        55
                                RECORD_IDENT_LOW = '07') THEN                    56
35                             CALL UPDATE_RECORD;                               57
36                    END;                                                       58
                                                                                 59
                   /*************************************************************  60
                     CALL ROUTINE TO CLOSE VSAM DATASET                          60
                    *************************************************************/  60
                                                                                 62
37                    VSIO_COMMAND = VSIO_CLOSE;                                 63
38                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                        64
                                    VSIO_FILE_BLOCK,                             65
                                    RECORD_IMAGE);                               66
39                    IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN              67
40                       CALL VSIO_ERROR;                                        68
                                                                                 69
41                    RETURN;                                                    70
                                                                                 71
42                 UPDATE_RECORD:                                                72
                      PROCEDURE;                                                 73
                                                                                 74
43                    COUNTER_EDIT = RECORD_COUNTER;                             75
44                    PRINT_AREA = COUNTER_EDIT || ': ' ||                       76
                                   RECORD_IMAGE_SCALAR || '  BEFORE';            77
45                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                       78
                                                                                 79
46                    RECORD_IDENT_HI = '99';                                    80
47                    CALL REWRITE_ES;                                           81
                                                                                 82
48                    IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN               83
49                       DO;                                                     84
50                          PRINT_AREA = '            ' ||                       85
                                       RECORD_IMAGE_SCALAR || '  AFTER';        86
51                          WRITE FILE(PRINTR) FROM(PRINT_LINE);                 87
```

```
52                      END;                                                  88
                                                                              89
53                    RETURN;                                                 90
                                                                              91
54                  END UPDATE_RECORD;                                        92
                                                                              93
55                READ_ES:                                                  94
                     PROCEDURE;                                             95
                                                                            96
                  /*************************************************************  97
                    CALL ROUTINE TO READ NEXT RECORD FROM VSAM DATASET          97
                   *************************************************************/ 97
                                                                            99
56                    VSIO_COMMAND = VSIO_READ;                            100
57                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                  101
                                    VSIO_FILE_BLOCK,                       102
                                    RECORD_IMAGE);                         103
58                    IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN        104
59                       IF (VSIO_RETURN_CODE = VSIO_RC_END_OF_FILE) THEN  105
60                          MORE_RECORDS = NO;                             106
61                       ELSE                                              107
61                          CALL VSIO_ERROR;                               108
62                    ELSE                                                 109
62                       RECORD_COUNTER = RECORD_COUNTER + 1;              110
                                                                           111
63                    RETURN;                                              112
                                                                           113
64                    END READ_ES;                                         114
                                                                           115
65                REWRITE_ES:                                              116
                     PROCEDURE;                                            117
                                                                           118
                  /*************************************************************  119
                    CALL ROUTINE TO REWRITE PREVIOUSLY READ RECORD TO VSAM DATASET 119
                   *************************************************************/ 119
                                                                           121
66                    VSIO_COMMAND = VSIO_REWRITE;                         122
67                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                  123
                                    VSIO_FILE_BLOCK,                       124
                                    RECORD_IMAGE);                         125
68                    IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN        126
69                       CALL VSIO_ERROR;                                  127
                                                                           128
70                    RETURN;                                              129
                                                                           130
71                    END REWRITE_ES;                                      131
                                                                           132
```

```
   72              VSIO_ERROR:                                                  133
                      PROCEDURE;                                                134
   73                    PRINT_AREA = 'VSAMIO ERROR OCCURRED DURING ' ||        135
                                 VSIO_COMMAND;                                  136
   74                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                   137
   75                    PRINT_AREA = 'VSIO_RETURN_CODE = ' ||                  138
                                 VSIO_RETURN_CODE;                              139
   76                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                   140
   77                    PRINT_AREA = 'VSIO_VSAM_RETURN_CODE = ' ||             141
                                 VSIO_VSAM_RETURN_CODE;                         142
   78                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                   143
   79                    PRINT_AREA = 'VSIO_VSAM_FUNCTION_CODE = ' ||           144
                                 VSIO_VSAM_FUNCTION_CODE;                       145
   80                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                   146
   81                    PRINT_AREA = 'VSIO_VSAM_FEEDBACK_CODE = ' ||           147
                                 VSIO_VSAM_FEEDBACK_CODE;                       148
   82                    WRITE FILE(PRINTR) FROM(PRINT_LINE);                   149
   83                    PRINT_AREA = ' ';                                      150
                                                                                151
   84                    RETURN;                                                152
                                                                                153
   85                 END VSIO_ERROR;                                           154
                                                                                155
   86                 DECLARE                                                   156
                         PRINTR FILE OUTPUT RECORD SEQUENTIAL EXTERNAL          157
                            ENV(F CTLASA);                                      158
                                                                                159
   87                 DECLARE                                                   160
                         COUNTER_EDIT                PICTURE 'ZZ,ZZZ,ZZ9V',     161
                         MORE_RECORDS                BIT(1),                    162
                         NO                          BIT(1)    INIT('0'B),      163
                         RECORD_COUNTER              FIXED BINARY(15,0),        164
                         YES                         BIT(1)    INIT('1'B);      165
                                                                                166
   88                 DECLARE                                                   167
                         1 RECORD_IMAGE,                                        168
                            2 RECORD_IDENT,                                     169
                               3 RECORD_IDENT_HI     CHAR(2),                   170
                               3 RECORD_IDENT_MID    CHAR(6),                   171
                               3 RECORD_IDENT_LOW    CHAR(2),                   172
                            2 RECORD_FIELDS          CHAR(70);                  173
                                                                                174
   89                 DECLARE                                                   175
                         RECORD_IMAGE_SCALAR         DEFINED RECORD_IMAGE       176
                                                     CHAR(80);                  177
                                                                                178
   90                 DECLARE                                                   179
```

```
                  1 PRINT_LINE,                                                  180
                    2 CARRIAGE_CONTROL          CHAR(1)   INIT(' '),             181
                    2 PRINT_AREA                CHAR(120);                       182
                                                                                 183
                 /*************************************************************  189
                                                                                 189
                    VV   VV    SSSSS       A      M     M    IIII      OOOOO     189
                    VV   VV   SS   SS     AAA     MM   MM     II      OO   OO    189
                    VV   VV   SS         AA AA    MMM MMM     II      OO   OO    189
                    VV   VV    SSSSS    AA   AA   MMMMMMM     II      OO   OO    189
                    VV   VV        SS   AA   AA   MM M MM     II      OO   OO    189
                     VV VV    SS   SS   AAAAAAA   MM   MM     II      OO   OO    189
                      VVV     SS   SS   AA   AA   MM   MM     II      OO   OO    189
                       V       SSSSS    AA   AA   MM   MM    IIII      OOOOO     189
                                                                                 189
                  ************************************************************** 189
                   THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS  189
                   ROUTINE.                                                      189
                                                                                 189
                   THE VSIO_PARAMETER_VALUES SUPPLY THE VALUES USED TO MOVE INTO 189
                   PARAMETER ENTRIES TO TAILOR THE ROUTINE TO A SPECIFIC DATASET AND  189
                   TO PROVIDE COMMANDS TO DRIVE THE ROUTINE.                     189
                  *************************************************************/ 189
                                                                                 207
91                 DECLARE                                                       209
                     1 VSIO_PARAMETER_VALUES   STATIC,                           210
                       2 VSIO_OPEN             CHAR(8)   INIT('OPEN    '),       211
                       2 VSIO_CLOSE            CHAR(8)   INIT('CLOSE   '),       212
                       2 VSIO_READ             CHAR(8)   INIT('READ    '),       213
                       2 VSIO_WRITE            CHAR(8)   INIT('WRITE   '),       214
                       2 VSIO_REWRITE          CHAR(8)   INIT('REWRITE '),       215
                       2 VSIO_DELETE           CHAR(8)   INIT('DELETE  '),       216
                       2 VSIO_START_EQUAL      CHAR(8)   INIT('STARTEQ '),       217
                       2 VSIO_START_NOTLESS    CHAR(8)   INIT('STARTGE '),       218
                       2 VSIO_KSDS             CHAR(4)   INIT('KSDS'),           219
                       2 VSIO_ESDS             CHAR(4)   INIT('ESDS'),           220
                       2 VSIO_RRDS             CHAR(4)   INIT('RRDS'),           221
                       2 VSIO_SEQUENTIAL       CHAR(10)  INIT('SEQUENTIAL'),     222
                       2 VSIO_DIRECT           CHAR(10)  INIT('DIRECT    '),     223
                       2 VSIO_DYNAMIC          CHAR(10)  INIT('DYNAMIC   '),     224
                       2 VSIO_INPUT            CHAR(6)   INIT('INPUT '),         225
                       2 VSIO_OUTPUT           CHAR(6)   INIT('OUTPUT'),         226
                       2 VSIO_INPUT_OUTPUT     CHAR(6)   INIT('UPDATE'),         227
                       2 (VSIO_RC_SUCCESS                INIT(0),                228
                        VSIO_RC_LOGIC_ERROR              INIT(8),                229
                        VSIO_RC_END_OF_FILE              INIT(9999),             230
                        VSIO_RC_UNKNOWN_COMMAND          INIT(20),               231
```

```
                          VSIO_RC_DATASET_ALREADY_OPEN  INIT(21),                232
                          VSIO_RC_DATASET_NOT_OPEN      INIT(22),                233
                          VSIO_RC_ORGANIZATION_UNKNOWN  INIT(23),                234
                          VSIO_RC_ACCESS_UNKNOWN        INIT(24),                235
                          VSIO_RC_ORG_ACCESS_MISMATCH   INIT(25),                236
                          VSIO_RC_MODE_UNKNOWN          INIT(26),                237
                          VSIO_RC_MODE_UNSUPPORTED      INIT(27),                238
                          VSIO_RC_DDNAME_BLANK          INIT(28))                239
                                                FIXED BINARY(15,0),              240
                       2 (VSIO_FB_DUPLICATE_RECORD      INIT(8),                 241
                          VSIO_FB_KEY_SEQUENCE          INIT(12),                242
                          VSIO_FB_RECORD_NOT_FOUND      INIT(16),                243
                          VSIO_FB_NO_MORE_SPACE         INIT(28),                244
                          VSIO_FB_READ_WITHOUT_START    INIT(88))                245
                                                FIXED BINARY(15,0),              246
                    /*****************************************************************  247
                      THE VSIO_PARAMETER_BLOCK IS THE COMMUNICATION INTERFACE TO THE     247
                      THE ROUTINE.                                                       247
                     *****************************************************************/  247
                                                                                        250
                     1 VSIO_PARAMETER_BLOCK    STATIC,                                  252
                        2 VSIO_COMMAND         CHAR(8)   INIT(' '),                     253
                        2 (VSIO_RETURN_CODE,                                            254
                          VSIO_VSAM_RC,                                                 255
                          VSIO_VSAM_FUNCTION,                                           256
                          VSIO_VSAM_FEEDBACK) FIXED BINARY(15,0) INIT(0);               257
                                                                                        258
                    /*****************************************************************  259
                                        END OF VSAMIO COPY BOOK                         259
                     *****************************************************************/  259
                    /*****************************************************************  262
                                                                                        262
                       VV   VV   SSSSS      A     M     M  IIII   OOOOO   FFFFFFF  BBBBBB    262
                       VV   VV  SS   SS    AAA    MM   MM   II   OO   OO  FF       BB   BB   262
                       VV   VV  SS        AA AA   MMM MMM   II   OO   OO  FF       BB   BB   262
                       VV   VV   SSSSS   AA   AA  MMMMMMM   II   OO   OO  FFFFF    BBBBBB    262
                       VV   VV       SS  AA   AA  MM M MM   II   OO   OO  FF       BB   BB   262
                        VV VV    SS   SS  AAAAAAA  MM   MM   II   OO   OO  FF       BB   BB   262
                         VVV     SS   SS  AA   AA  MM   MM   II   OO   OO  FF       BB   BB   262
                          V       SSSSS   AA   AA  MM   MM  IIII   OOOOO   FF       BBBBBB    262
                                                                                        262
                    *****************************************************************  262
                      THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS   262
                      ROUTINE, AND ARE USED TO COMMUNICATE CHARACTERISTICS FOR A SINGLE     262
                      VSAM DATASET.                                                         262
                                                                                            262
                      WITH THE 2 EXCEPTIONS FOR RECORD LENGTH (TO ACCOMODATE VARIABLE       262
```

```
                  LENGTH RECORDS) AND RELATIVE RECORD (TO ACCOMODATE RELATIVE RECORD    262
                  DATASETS), THESE DATA NAMES MUST BE POPULATED PRIOR TO CALLING THE    262
                  ROUTINE TO OPEN THE DATASET AND MUST NOT THEN BE CHANGED UNTIL THE    262
                  DATASET HAS BEEN CLOSED.                                              262
                ***************************************************************/      262
                                                                                        283
92                 DECLARE                                                              285
                      1 VSIO_FILE_BLOCK            STATIC,                             286
                        2 VSFB_DDNAME              CHAR(8)   INIT(' '),                287
                        2 VSFB_ORGANIZATION        CHAR(4)   INIT(' '),                288
                        2 VSFB_ACCESS              CHAR(10)  INIT(' '),                289
                        2 VSFB_MODE                CHAR(6)   INIT(' '),                290
                        2 (VSFB_RECORD_LENGTH,                                         291
                           VSFB_KEY_POSITION,                                          292
                           VSFB_KEY_LENGTH)        FIXED BINARY(15,0) INIT(0),         293
                        2 VSFB_FILE_STATUS         CHAR(1)   INIT('C'),                294
                        2 VSFB_RESERVED            CHAR(161);                          295
                                                                                       296
               /*****************************************************************     297
                                 END OF VSAMIOFB COPY BOOK                             297
                ***************************************************************/      297
                                                                                       186
93                 END ESDSUPD;                                                        187
                                                                                       188
```

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
90	CARRIAGE_CONTROL	IN PRINT_LINE,AUTOMATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER
87	COUNTER_EDIT	AUTOMATIC,UNALIGNED,DECIMAL,PICTURE(ZZ,ZZZ,ZZ9V) 43,44
1	ESDSUPD	ENTRY,DECIMAL,FLOAT(SINGLE)
87	MORE_RECORDS	AUTOMATIC,UNALIGNED,STRING(1),BIT 16,31,33,60
87	NO	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 60
90	PRINT_AREA	IN PRINT_LINE,AUTOMATIC,UNALIGNED,STRING(120),CHARACTER 10,12,14,44,50,73,75,77,79,81,83
90	PRINT_LINE	AUTOMATIC,STRUCTURE 11,13,15,45,51,74,76,78,80,82
86	PRINTR	FILE,EXTERNAL,OUTPUT,RECORD,SEQUENTIAL,ENVIRONMENT(F CTLASA) 9,11,13,15,45,51,74,76,78,80,82
55	READ_ES	ENTRY,DECIMAL,FLOAT(SINGLE) 32
87 *********	RECORD_COUNTER	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 43,62,62
88	RECORD_FIELDS	IN RECORD_IMAGE,AUTOMATIC,UNALIGNED,STRING(70),CHARACTER
88	RECORD_IDENT	IN RECORD_IMAGE,AUTOMATIC,STRUCTURE
88	RECORD_IDENT_HI	IN RECORD_IDENT IN RECORD_IMAGE,AUTOMATIC,UNALIGNED,STRING(2), CHARACTER 46
88	RECORD_IDENT_LOW	IN RECORD_IDENT IN RECORD_IMAGE,AUTOMATIC,UNALIGNED,STRING(2), CHARACTER 34,34,34
88	RECORD_IDENT_MID	IN RECORD_IDENT IN RECORD_IMAGE,AUTOMATIC,UNALIGNED,STRING(6),

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		CHARACTER
88	RECORD_IMAGE	AUTOMATIC,STRUCTURE 25,38,57,67
89	RECORD_IMAGE_SCALAR	AUTOMATIC,DEFINED,UNALIGNED,STRING(80),CHARACTER 44,50
65	REWRITE_ES	ENTRY,DECIMAL,FLOAT(SINGLE) 47
	SYSPRINT	FILE,EXTERNAL 5,6,7
42	UPDATE_RECORD	ENTRY,DECIMAL,FLOAT(SINGLE) 35
	VSAMIOP	EXTERNAL,ENTRY,DECIMAL,FLOAT(SINGLE) 25,38,57,67
92	VSFB_ACCESS	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(10),CHARACTER 19
92	VSFB_DDNAME	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(8),CHARACTER 17
92	VSFB_FILE_STATUS	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER
92 *********	VSFB_KEY_LENGTH	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 23
92 *********	VSFB_KEY_POSITION	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 22
92	VSFB_MODE	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(6),CHARACTER 20
92	VSFB_ORGANIZATION	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(4),CHARACTER 18
92 *********	VSFB_RECORD_LENGTH	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 21
92	VSFB_RESERVED	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,STRING(161),CHARACTER

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
91	VSIO_CLOSE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 37
91	VSIO_COMMAND	IN VSIO_PARAMETER_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 24,37,56,66,73
91	VSIO_DELETE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
91	VSIO_DIRECT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER
91	VSIO_DYNAMIC	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER
72	VSIO_ERROR	ENTRY,DECIMAL,FLOAT(SINGLE) 28,40,61,69
91	VSIO_ESDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER 18
91 ********	VSIO_FB_DUPLICATE_RECORD	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91 ********	VSIO_FB_KEY_SEQUENCE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91 ********	VSIO_FB_NO_MORE_SPACE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91 ********	VSIO_FB_READ_WITHOUT_START	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91 ********	VSIO_FB_RECORD_NOT_FOUND	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
92	VSIO_FILE_BLOCK	STATIC,STRUCTURE 25,38,57,67
91	VSIO_INPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
91		VSIO_INPUT_OUTPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER 20
91		VSIO_KSDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER
91		VSIO_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 24
91		VSIO_OUTPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER
91		VSIO_PARAMETER_BLOCK	STATIC,STRUCTURE 25,38,57,67
91		VSIO_PARAMETER_VALUES	STATIC,STRUCTURE
91	*********	VSIO_RC_ACCESS_UNKNOWN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91	*********	VSIO_RC_DATASET_ALREADY_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91	*********	VSIO_RC_DATASET_NOT_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91	*********	VSIO_RC_DDNAME_BLANK	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91	*********	VSIO_RC_END_OF_FILE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 59
91	*********	VSIO_RC_LOGIC_ERROR	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91	*********	VSIO_RC_MODE_UNKNOWN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91	*********	VSIO_RC_MODE_UNSUPPORTED	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
91	*********	VSIO_RC_ORG_ACCESS_MISMATCH	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED

```
DCL NO.         IDENTIFIER                          ATTRIBUTES AND REFERENCES

                                                    (15,0)

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

91    ********* VSIO_RC_SUCCESS                     IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                    (15,0)
                                                    26,39,48,58,68

91    ********* VSIO_RC_UNKNOWN_COMMAND             IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                    (15,0)

91              VSIO_READ                           IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                    CHARACTER
                                                    56

91    ********* VSIO_RETURN_CODE                    IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                    (15,0)
                                                    26,39,48,58,59,68,75

91              VSIO_REWRITE                        IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                    CHARACTER
                                                    66

91              VSIO_RRDS                           IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4),
                                                    CHARACTER

91              VSIO_SEQUENTIAL                     IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10),
                                                    CHARACTER
                                                    19

91              VSIO_START_EQUAL                    IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                    CHARACTER

91              VSIO_START_NOTLESS                  IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                    CHARACTER

91    ********* VSIO_VSAM_FEEDBACK                  IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                    (15,0)

                VSIO_VSAM_FEEDBACK_CODE             AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE)
                                                    81

91    ********* 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) 79
91 *********	VSIO_VSAM_RC	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
	VSIO_VSAM_RETURN_CODE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 77
91	VSIO_WRITE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
87	YES	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 16

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
90	PRINT_LINE	121
88	RECORD_IMAGE	80
92	VSIO_FILE_BLOCK	196
91	VSIO_PARAMETER_BLOCK	16
91	VSIO_PARAMETER_VALUES	158

STORAGE REQUIREMENTS.

THE STORAGE AREA FOR THE PROCEDURE LABELLED ESDSUPD IS 512 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 268 BYTES LONG.

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

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

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

THE PROGRAM CSECT IS NAMED ESDSUPD AND IS 1994 BYTES LONG.

THE STATIC CSECT IS NAMED ESDSUPDA AND IS 6144 BYTES LONG.

STATISTICS MACRO RECORDS = 299,SOURCE RECORDS = 302,PROG TEXT STMNTS = 93,OBJECT BYTES = 1994

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	0092	00AA	00B0	00BA	00C6	00C6	00F0	0108	0108	010E
STATEMENT NO	42	43	44	45	46	47	48	49	50	51	52	53	54

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE READ_ES

OFFSET (HEX)	0000	0034	003A	005A	0066	0072	0080	008E	009E	00A4
STATEMENT NO	55	56	57	58	59	60	61	62	63	64

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE REWRITE_ES

OFFSET (HEX)	0000	0038	003E	005E	006A	0074	007A
STATEMENT NO	65	66	67	68	69	70	71

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	72	73	74	75	76	77	78	79	80	81	82	83	84	85

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE ESDSUPD

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	023A	0244	0248	024E	026A	0276	0280	0286
STATEMENT NO	28	29	30	31	32	33	34	35	36	37	38	39	40	41	93

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 77, 79, 81.

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.
```

```
ESDSUPDT: READ/REWRITE ESDS SEQUENTIALLY
----------------------------------------

         2: 0130758002 VIRGINIA O RENFRO    2111 BRIDGE AVENUE         BRIDGEPORT      CT     BEFORE
            9930758002 VIRGINIA O RENFRO    2111 BRIDGE AVENUE         BRIDGEPORT      CT     AFTER
         6: 0561094006 TAMMY L SCHAEFER     3911 KINGS RIDGE STREET    EUGENE          OR     BEFORE
            9961094006 TAMMY L SCHAEFER     3911 KINGS RIDGE STREET    EUGENE          OR     AFTER
         7: 0593878007 LOUIS W HORTON       9722 EASTHAVEN STREET      DES PLAINES     IL     BEFORE
            9993878007 LOUIS W HORTON       9722 EASTHAVEN STREET      DES PLAINES     IL     AFTER
        52: 4813060102 REBECCA S NORTON     1175 ELLIS COURT           LOUISVILLE      KY     BEFORE
            9913060102 REBECCA S NORTON     1175 ELLIS COURT           LOUISVILLE      KY     AFTER
        56: 5105151106 SHIRLEY P TYLER      5262 BRIDGE CIRCLE         FORT COLLINS    CO     BEFORE
            9905151106 SHIRLEY P TYLER      5262 BRIDGE CIRCLE         FORT COLLINS    CO     AFTER
        57: 5220743107 CHRISTOPHER F MORGAN 9624 BONANZA AVENUE        DOWNEY          CA     BEFORE
            9920743107 CHRISTOPHER F MORGAN 9624 BONANZA AVENUE        DOWNEY          CA     AFTER
```