J E S 2 J O B L O G

```
18.27.21 JOB  162  IEF677I WARNING MESSAGE(S) FOR JOB VSTESTR6 ISSUED
18.27.21 JOB  162  $HASP373 VSTESTR6 STARTED - INIT  1 - CLASS A - SYS HMVS
18.27.21 JOB  162  IEF403I VSTESTR6 - STARTED - TIME=18.27.21
18.27.21 JOB  162  CCI001C PL1L    /IEMAA   /00:00:00.21/        /00004/SYS     /VSTESTR6
18.27.22 JOB  162  CCI001C LKED    /IEWL    /00:00:00.06/        /00000/SYS     /VSTESTR6
18.27.22 JOB  162  CCI001C GO      /PGM=*.DD/00:00:00.01/        /00000/SYS     /VSTESTR6
18.27.22 JOB  162  IEF404I VSTESTR6 - ENDED - TIME=18.27.22
18.27.22 JOB  162  $HASP395 VSTESTR6 ENDED

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

 07 JUL 20 JOB EXECUTION DATE

        33 CARDS READ

     1,621 SYSOUT PRINT RECORDS

         0 SYSOUT PUNCH RECORDS

      0.00 MINUTES EXECUTION TIME
```

```
 1      //VSTESTR6 JOB (SYS),'VSAMIOP IVP RRDSRAND',CLASS=A,MSGCLASS=X,         JOB  162
        //             REGION=4096K
        ***
        ********************************************************************
        *** PL/1 MODULE: RRDSRAND     VSAM DATASET: VSTESTRR.CLUSTER (RRDS)
        ***
        *** RANDOMLY ADDS/DELETES/CHANGES RECORDS IN NUMBERED 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(RRDSRAND),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.RRDSF01 DD DSN=PUB001.VSTESTRR.CLUSTER,DISP=OLD
27      //GO.IMAGES DD  *
```

```
 STMT NO. MESSAGE
-
   19     IEF686I DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
IEF236I ALLOC. FOR VSTESTR6 PL1L PL1F
IEF237I 253  ALLOCATED TO STEPLIB
IEF237I 253  ALLOCATED TO SYS00406
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 VSTESTR6 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.JOB00162.SO0102                        SYSOUT
IEF285I   SYS20189.T182721.RA000.VSTESTR6.LOADSET     PASSED        *------310
IEF285I   VOL SER NOS= MVS380.
IEF285I   SYS20189.T182721.RA000.VSTESTR6.SYSUT3      DELETED       *------475
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T182721.RA000.VSTESTR6.SYSUT1      DELETED       *--------0
IEF285I   VOL SER NOS= MVS370.
IEF285I   SYSC.VSAMIOP.SOURCE                         KEPT          *--------5
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   SYSC.VSAMIOP.MACLIB                         KEPT          *-------27
IEF285I   VOL SER NOS= SYSCPK.
IEF373I STEP /PL1L    / START 20189.1827
IEF374I STEP /PL1L    / STOP  20189.1827 CPU    0MIN 00.21SEC SRB    0MIN 00.05SEC VIRT  4096K SYS   212K
**** JOBCARD READ 20189 18:27:21 ************************************************************************************
*                                      PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                              *
*  STEP NAME  PL1L        USER CORE        4096K   TAPES USED/IO 000/000000000   START   TIME  18:27:21    TCB TIME  00:00:00.21  *
*  PGM  NAME  IEMAA       SYSTEM CORE       212K   DISKS USED/IO 004/000000817   STOP    TIME  18:27:21    SRB TIME  00:00:00.05  *
*   COND CODE    0004     PRIVATE AREA SZ   4096K       ALLOC TIME  18:27:21      ELAPSED TIME                PGM LOAD  18:27:21   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004          4400   00:00:00.30            0             0           0             0              0             0             0  *
************************************************************************************************************************
*  CPU $ (     0.07) + EXCP $ (     1.10) + MEMORY $ (     2.45) = TOTAL $ (     3.62)                                  *
************************************************************************************************************************
IEF236I ALLOC. FOR VSTESTR6 LKED PL1F
IEF237I 253  ALLOCATED TO SYSLIB
IEF237I 253  ALLOCATED TO
IEF237I 253  ALLOCATED TO SYS00408
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 VSTESTR6 LKED PL1F - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYSC.PL1LIB                                 KEPT          *------130
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.T182721.RA000.VSTESTR6.GOSET       PASSED        *-------64
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T182721.RA000.VSTESTR6.SYSUT1      DELETED       *--------6
IEF285I   VOL SER NOS= MVS370.
IEF285I   JES2.JOB00162.SO0103                        SYSOUT
```

```
IEF285I   SYS20189.T182721.RA000.VSTESTR6.LOADSET      DELETED       *------311
IEF285I   VOL SER NOS= MVS380.
IEF373I STEP /LKED    / START 20189.1827
IEF374I STEP /LKED    / STOP  20189.1827 CPU    0MIN 00.06SEC SRB    0MIN 00.02SEC VIRT   260K SYS   168K
********************************************************************************************************************
*                                        PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                             *
*  STEP NAME  LKED        USER CORE          260K   TAPES USED/IO 000/000000000   START   TIME  18:27:21    TCB TIME  00:00:00.06  *
*  PGM  NAME  IEWL        SYSTEM CORE        168K   DISKS USED/IO 004/000000511   STOP    TIME  18:27:22    SRB TIME  00:00:00.02  *
*   COND CODE    0000     PRIVATE AREA SZ   4096K       ALLOC TIME  18:27:21      ELAPSED TIME                PGM LOAD  18:27:21   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004          2606   00:00:00.08            0             0         0             0              0             0             0  *
********************************************************************************************************************
*  CPU $ (     0.02) + EXCP $ (     0.68) + MEMORY $ (     0.04) = TOTAL $ (     0.74)                                 *
********************************************************************************************************************
IEF236I ALLOC. FOR VSTESTR6 GO PL1F
IEF237I 251  ALLOCATED TO PGM=*.DD
IEF237I 253  ALLOCATED TO STEPLIB
IEF237I 253  ALLOCATED TO
IEF237I 253  ALLOCATED TO SYS00410
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO PRINTR
IEF237I JES2 ALLOCATED TO SYSUDUMP
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I 190  ALLOCATED TO RRDSF01
IEF237I 190  ALLOCATED TO SYS00412
IEF237I JES2 ALLOCATED TO IMAGES
IEF142I VSTESTR6 GO PL1F - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS20189.T182721.RA000.VSTESTR6.GOSET        KEPT          *--------0
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYSC.PL1LIB                                  KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   SYSC.PL1LIB                                  KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   UCSYSCPK                                     KEPT          *--------0
IEF285I   VOL SER NOS= SYSCPK.
IEF285I   JES2.JOB00162.SO0104                         SYSOUT
IEF285I   JES2.JOB00162.SO0105                         SYSOUT
IEF285I   JES2.JOB00162.SO0106                         SYSOUT
IEF285I   JES2.JOB00162.SO0107                         SYSOUT
IEF285I   PUB001.VSTESTRR.CLUSTER                      KEPT          *-------15
IEF285I   VOL SER NOS= PUB001.
IEF285I   UCPUB001                                     KEPT          *--------0
IEF285I   VOL SER NOS= PUB001.
IEF285I   JES2.JOB00162.SI0101                         SYSIN
IEF373I STEP /GO      / START 20189.1827
IEF374I STEP /GO      / STOP  20189.1827 CPU    0MIN 00.01SEC SRB    0MIN 00.00SEC VIRT   104K SYS   196K
********************************************************************************************************************
*                                        PRC-CCI  370/148 VS2 R03.8  HMVS  STEP STATISTICS                             *
*  STEP NAME  GO          USER CORE          104K   TAPES USED/IO 000/000000000   START   TIME  18:27:22    TCB TIME  00:00:00.01  *
*  PGM  NAME  PGM=*.DD    SYSTEM CORE        196K   DISKS USED/IO 003/000000015   STOP    TIME  18:27:22    SRB TIME  00:00:00.00  *
*   COND CODE    0000     PRIVATE AREA SZ   4096K       ALLOC TIME  18:27:22      ELAPSED TIME                PGM LOAD  18:27:22   *
** PGNO * NR SRV UNITS * ACTIVE TIME ** PAGES IN *** PAGES OUT ** # SWAPS * PGS SWAP IN * PGS SWAP OUT * VIO PGS IN * VIO PGS OUT **
*   004           105   00:00:00.02            0             0         0             0              0             0             0  *
********************************************************************************************************************
*  CPU $ (     0.00) + EXCP $ (     0.02) + MEMORY $ (     0.00) = TOTAL $ (     0.02)                                 *
********************************************************************************************************************
IEF237I 251  ALLOCATED TO SYS00001
IEF285I   SYS20189.T182722.RA000.VSTESTR6.R0000001     KEPT          *--------0
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20189.T182721.RA000.VSTESTR6.GOSET        DELETED
IEF285I   VOL SER NOS= WORK00.
```

```
IEF375I  JOB /VSTESTR6/ START 20189.1827
IEF376I  JOB /VSTESTR6/ STOP  20189.1827 CPU    0MIN 00.28SEC SRB    0MIN 00.07SEC
```

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    /*****************************************************************20210000
  2                                                                      20220000
  3      RRDSRAND - TESTS THE VSAMIO ROUTINE BY READING AN RRDS CLUSTER  20230000
  4                 RANDOMLY AND ADDING/UPDATING/DELETING RECORDS.       20240000
  5                                                                      20250000
  6     ******************************************************************/20260000
  7    RRDSRND:                                                          20270000
  8       PROCEDURE OPTIONS(MAIN);                                       20280000
  9                                                                      20290000
 10          ON ERROR                                                   20300000
 11             BEGIN;                                                  20310000
 12                ON ERROR SYSTEM;                                     20320000
 13                PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*');20330000
 14                PUT SKIP DATA;                                       20340000
 15                PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*');20350000
 16             END;                                                    20360000
 17                                                                     20370000
 18          OPEN                                                       20380000
 19             FILE(IMAGES),                                           20390000
 20             FILE(PRINTR) LINESIZE(133);                             20400000
 21                                                                     20410000
 22          ON ENDFILE(IMAGES)                                         20420000
 23             MORE_RECORDS = NO;                                      20430000
 24                                                                     20440000
 25          PRINT_AREA = 'RRDSRAND: READ/UPDATE RRDS DIRECT';          20450000
 26          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       20460000
 27          PRINT_AREA = '-------------------------------';            20470000
 28          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       20480000
 29          PRINT_AREA = ' ';                                          20490000
 30          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       20500000
 31                                                                     20510000
 32          MORE_RECORDS = YES;                                        20520000
 33                                                                     20530000
 34    /*****************************************************************20540000
 35      ESTABLISH PARAMETERS FOR VSAM DATASET AND CALL ROUTINE TO OPEN  20550000
 36     ******************************************************************/20560000
 37          VSFB_DDNAME = 'RRDSF01';                                   20570000
 38          VSFB_ORGANIZATION = VSIO_RRDS;                             20580000
 39          VSFB_ACCESS = VSIO_DIRECT;                                 20590000
 40          VSFB_MODE = VSIO_INPUT_OUTPUT;                             20600000
 41          VSFB_RECORD_LENGTH = 80;                                   20610000
 42          VSFB_KEY_POSITION = 0;                                     20620000
 43          VSFB_KEY_LENGTH = 0;                                       20630000
 44          VSIO_COMMAND = VSIO_OPEN;                                  20640000
```

MACRO SOURCE2 LISTING

```
45          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                          20650000
46                        VSIO_FILE_BLOCK,                               20660000
47                        VSAM_RECORD);                                  20670000
48          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                20680000
49             DO;                                                       20690000
50                CALL VSIO_ERROR;                                       20700000
51                RETURN;                                                20710000
52             END;                                                      20720000
53                                                                       20730000
54          DO WHILE(MORE_RECORDS);                                      20740000
55             READ FILE(IMAGES) INTO(MAINT_RECORD);                     20750000
56             IF (MORE_RECORDS) THEN                                    20760000
57                CALL PROCESS_MAINT;                                    20770000
58          END;                                                         20780000
59                                                                       20790000
60          CLOSE FILE(IMAGES);                                          20800000
61                                                                       20810000
62       /*****************************************************************20820000
63         CALL ROUTINE TO CLOSE VSAM DATASET                            20830000
64        *****************************************************************/20840000
65          VSIO_COMMAND = VSIO_CLOSE;                                   20850000
66          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                          20860000
67                        VSIO_FILE_BLOCK,                               20870000
68                        VSAM_RECORD);                                  20880000
69          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN                20890000
70             CALL VSIO_ERROR;                                          20900000
71                                                                       20910000
72          COUNTER_EDIT = ADD_COUNTER;                                  20920000
73          PRINT_AREA = COUNTER_EDIT || ' RECORDS ADDED';               20930000
74          WRITE FILE(PRINTR) FROM(PRINT_LINE);                         20940000
75          COUNTER_EDIT = CHANGE_COUNTER;                               20950000
76          PRINT_AREA = COUNTER_EDIT || ' RECORDS CHANGED';             20960000
77          WRITE FILE(PRINTR) FROM(PRINT_LINE);                         20970000
78          COUNTER_EDIT = DELETE_COUNTER;                               20980000
79          PRINT_AREA = COUNTER_EDIT || ' RECORDS DELETED';             20990000
80          WRITE FILE(PRINTR) FROM(PRINT_LINE);                         21000000
81                                                                       21010000
82          RETURN;                                                      21020000
83                                                                       21030000
84       PROCESS_MAINT:                                                  21040000
85          PROCEDURE;                                                   21050000
86                                                                       21060000
87             RECORD_COUNTER = RECORD_COUNTER + 1;                      21070000
88                                                                       21080000
89             IF (MAINT_ACTION = 'A') THEN                              21090000
```

MACRO SOURCE2 LISTING

```
 90            CALL PROCESS_ADD;                                        21100000
 91         ELSE                                                        21110000
 92            IF (MAINT_ACTION = 'C') THEN                             21120000
 93               CALL PROCESS_CHANGE;                                  21130000
 94            ELSE                                                     21140000
 95               IF (MAINT_ACTION = 'D') THEN                          21150000
 96                  CALL PROCESS_DELETE;                               21160000
 97               ELSE                                                  21170000
 98                  DO;                                                21180000
 99                     COUNTER_EDIT = RECORD_COUNTER;                  21190000
100                     PRINT_AREA = COUNTER_EDIT || ': INVALID ACTION: ' || 21200000
101                                  MAINT_RECORD_SCALAR;               21210000
102                     WRITE FILE(PRINTR) FROM(PRINT_LINE);            21220000
103                  END;                                               21230000
104                                                                     21240000
105         RETURN;                                                     21250000
106                                                                     21260000
107      END PROCESS_MAINT;                                             21270000
108                                                                     21280000
109   PROCESS_ADD:                                                      21290000
110      PROCEDURE;                                                     21300000
111                                                                     21310000
112         VSAM_RECORD_SCALAR = MAINT_IMAGE;                           21320000
113         VSFB_KEY_LENGTH = VSAM_KEY_RRN;                             21330000
114         RRN_EDIT = VSAM_KEY_RRN;                                    21340000
115         PRINT_AREA = 'RRN: ' || RRN_EDIT || '  DATA: ' ||           21350000
116                      VSAM_RECORD_SCALAR || '  ADDING';              21360000
117         WRITE FILE(PRINTR) FROM(PRINT_LINE);                        21370000
118         CALL WRITE_RR;                                              21380000
119         IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN                21390000
120            DO;                                                      21400000
121               PRINT_AREA = (101)' ' || 'ADDED OK';                  21410000
122               WRITE FILE(PRINTR) FROM(PRINT_LINE);                  21420000
123            END;                                                     21430000
124         ELSE                                                        21440000
125            IF (VSIO_VSAM_FEEDBACK = VSIO_FB_DUPLICATE_RECORD) THEN  21450000
126               DO;                                                   21460000
127                  PRINT_AREA = (101)' ' || '*DUPLICATE RECORD*';     21470000
128                  WRITE FILE(PRINTR) FROM(PRINT_LINE);               21480000
129               END;                                                  21490000
130                                                                     21500000
131         RETURN;                                                     21510000
132                                                                     21520000
133      END PROCESS_ADD;                                               21530000
134                                                                     21540000
```

MACRO SOURCE2 LISTING

```
135    PROCESS_CHANGE:                                                  21550000
136       PROCEDURE;                                                    21560000
137                                                                     21570000
138          VSAM_RECORD_SCALAR = MAINT_IMAGE;                          21580000
139          VSFB_KEY_LENGTH = VSAM_KEY_RRN;                            21590000
140          RRN_EDIT = VSAM_KEY_RRN;                                   21600000
141          CALL READ_RR;                                              21610000
142          IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN               21620000
143             CALL COMPLETE_CHANGE;                                   21630000
144          ELSE                                                       21640000
145             DO;                                                     21650000
146                PRINT_AREA = 'READING FOR UPDATE: ' ||               21660000
147                             RRN_EDIT ||                             21670000
148                             '  *** NOT FOUND ***';                  21680000
149                WRITE FILE(PRINTR) FROM(PRINT_LINE);                 21690000
150             END;                                                    21700000
151                                                                     21710000
152          RETURN;                                                    21720000
153                                                                     21730000
154       END PROCESS_CHANGE;                                           21740000
155                                                                     21750000
156    COMPLETE_CHANGE:                                                 21760000
157       PROCEDURE;                                                    21770000
158                                                                     21780000
159          PRINT_AREA = 'RRN: ' || RRN_EDIT || '  DATA: ' ||         21790000
160                       VSAM_RECORD_SCALAR || '  BEFORE';            21800000
161          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       21810000
162          VSAM_RECORD_SCALAR = MAINT_IMAGE;                          21820000
163          CALL REWRITE_RR;                                           21830000
164          IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN               21840000
165             DO;                                                     21850000
166                PRINT_AREA = (19)' ' || VSAM_RECORD_SCALAR || '  AFTER';  21860000
167                WRITE FILE(PRINTR) FROM(PRINT_LINE);                 21870000
168             END;                                                    21880000
169                                                                     21890000
170          RETURN;                                                    21900000
171                                                                     21910000
172       END COMPLETE_CHANGE;                                          21920000
173                                                                     21930000
174    PROCESS_DELETE:                                                  21940000
175       PROCEDURE;                                                    21950000
176                                                                     21960000
177          VSAM_RECORD_SCALAR = MAINT_IMAGE;                          21970000
178          VSFB_KEY_LENGTH = VSAM_KEY_RRN;                            21980000
179          RRN_EDIT = VSAM_KEY_RRN;                                   21990000
```

MACRO SOURCE2 LISTING

```
180          CALL READ_RR;                                              22000000
181          IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN               22010000
182             CALL COMPLETE_DELETE;                                   22020000
183          ELSE                                                       22030000
184             DO;                                                     22040000
185                PRINT_AREA = 'READING FOR DELETE: ' ||               22050000
186                             RRN_EDIT ||                             22060000
187                             '  *** NOT FOUND ***';                  22070000
188                WRITE FILE(PRINTR) FROM(PRINT_LINE);                 22080000
189             END;                                                    22090000
190                                                                     22100000
191          RETURN;                                                    22110000
192                                                                     22120000
193       END PROCESS_DELETE;                                           22130000
194                                                                     22140000
195    COMPLETE_DELETE:                                                 22150000
196       PROCEDURE;                                                    22160000
197                                                                     22170000
198          PRINT_AREA = 'RRN: ' || RRN_EDIT || '  DATA: ' ||         22180000
199                       VSAM_RECORD_SCALAR || '  BEFORE';            22190000
200          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       22200000
201          CALL DELETE_RR;                                            22210000
202          IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN               22220000
203             DO;                                                     22230000
204                PRINT_AREA = (101)' ' || 'DELETED';                  22240000
205                WRITE FILE(PRINTR) FROM(PRINT_LINE);                 22250000
206             END;                                                    22260000
207                                                                     22270000
208          RETURN;                                                    22280000
209                                                                     22290000
210       END COMPLETE_DELETE;                                          22300000
211                                                                     22310000
212    DELETE_RR:                                                       22320000
213       PROCEDURE;                                                    22330000
214                                                                     22340000
215    /*****************************************************************22350000
216      CALL ROUTINE TO DELETE LAST RECORD READ FROM VSAM DATASET      22360000
217     *****************************************************************/22370000
218          VSIO_COMMAND = VSIO_DELETE;                                22380000
219          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                        22390000
220                        VSIO_FILE_BLOCK,                             22400000
221                        VSAM_RECORD);                                22410000
222          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN              22420000
223             CALL VSIO_ERROR;                                        22430000
224          ELSE                                                       22440000
```

```
225             DELETE_COUNTER = DELETE_COUNTER + 1;                    22450000
226                                                                     22460000
227          RETURN;                                                    22470000
228                                                                     22480000
229       END DELETE_RR;                                                22490000
230                                                                     22500000
231    READ_RR:                                                         22510000
232       PROCEDURE;                                                    22520000
233                                                                     22530000
234    /*****************************************************************22540000
235      CALL ROUTINE TO READ NEXT RECORD FROM VSAM DATASET             22550000
236     *****************************************************************/22560000
237          VSIO_COMMAND = VSIO_READ;                                  22570000
238          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                        22580000
239                        VSIO_FILE_BLOCK,                             22590000
240                        VSAM_RECORD);                                22600000
241          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN              22610000
242             IF (VSIO_VSAM_FEEDBACK ^= VSIO_FB_RECORD_NOT_FOUND) THEN 22620000
243                CALL VSIO_ERROR;                                     22630000
244                                                                     22640000
245          RETURN;                                                    22650000
246                                                                     22660000
247       END READ_RR;                                                  22670000
248                                                                     22680000
249    REWRITE_RR:                                                      22690000
250       PROCEDURE;                                                    22700000
251                                                                     22710000
252    /*****************************************************************22720000
253      CALL ROUTINE TO REWRITE PREVIOUSLY READ RECORD TO VSAM DATASET 22730000
254     *****************************************************************/22740000
255          VSIO_COMMAND = VSIO_REWRITE;                               22750000
256          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                        22760000
257                        VSIO_FILE_BLOCK,                             22770000
258                        VSAM_RECORD);                                22780000
259          IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN              22790000
260             CALL VSIO_ERROR;                                        22800000
261          ELSE                                                       22810000
262             CHANGE_COUNTER = CHANGE_COUNTER + 1;                    22820000
263                                                                     22830000
264          RETURN;                                                    22840000
265                                                                     22850000
266       END REWRITE_RR;                                               22860000
267                                                                     22870000
268    WRITE_RR:                                                        22880000
269       PROCEDURE;                                                    22890000
```

MACRO SOURCE2 LISTING

```
270                                                                   22900000
271    /*****************************************************************22910000
272      CALL ROUTINE TO WRITE RECORD INTO VSAM DATASET                 22920000
273     *****************************************************************/22930000
274          VSIO_COMMAND = VSIO_WRITE;                                 22940000
275          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                        22950000
276                        VSIO_FILE_BLOCK,                             22960000
277                        VSAM_RECORD);                                22970000
278          IF (VSIO_RETURN_CODE ª= VSIO_RC_SUCCESS) THEN              22980000
279             IF (VSIO_VSAM_FEEDBACK ª= VSIO_FB_DUPLICATE_RECORD) THEN 22990000
280                CALL VSIO_ERROR;                                     23000000
281             ELSE                                                    23010000
282                ;                                                    23020000
283          ELSE                                                       23030000
284             ADD_COUNTER = ADD_COUNTER + 1;                          23040000
285                                                                     23050000
286          RETURN;                                                    23060000
287                                                                     23070000
288       END WRITE_RR;                                                 23080000
289                                                                     23090000
290    VSIO_ERROR:                                                      23100000
291       PROCEDURE;                                                    23110000
292          PRINT_AREA = 'VSAMIO ERROR OCCURRED DURING ' ||            23120000
293                       VSIO_COMMAND;                                 23130000
294          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       23140000
295          PRINT_AREA = 'VSIO_RETURN_CODE = ' ||                      23150000
296                       VSIO_RETURN_CODE;                             23160000
297          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       23170000
298          PRINT_AREA = 'VSIO_VSAM_RETURN_CODE = ' ||                 23180000
299                       VSIO_VSAM_RETURN_CODE;                        23190000
300          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       23200000
301          PRINT_AREA = 'VSIO_VSAM_FUNCTION_CODE = ' ||               23210000
302                       VSIO_VSAM_FUNCTION_CODE;                      23220000
303          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       23230000
304          PRINT_AREA = 'VSIO_VSAM_FEEDBACK_CODE = ' ||               23240000
305                       VSIO_VSAM_FEEDBACK_CODE;                      23250000
306          WRITE FILE(PRINTR) FROM(PRINT_LINE);                       23260000
307          PRINT_AREA = ' ';                                          23270000
308                                                                     23280000
309          RETURN;                                                    23290000
310                                                                     23300000
311       END VSIO_ERROR;                                               23310000
312                                                                     23320000
313       DECLARE                                                       23330000
314          IMAGES FILE INPUT RECORD SEQUENTIAL EXTERNAL               23340000
```

MACRO SOURCE2 LISTING

```
315            ENV(F),                                                  23350000
316         PRINTR FILE OUTPUT RECORD SEQUENTIAL EXTERNAL               23360000
317            ENV(F CTLASA);                                           23370000
318                                                                     23380000
319      DECLARE                                                        23390000
320         ADD_COUNTER                  FIXED BINARY(15,0) INIT(0),   23400000
321         CHANGE_COUNTER               FIXED BINARY(15,0) INIT(0),   23410000
322         COUNTER_EDIT                 PICTURE 'ZZ,ZZZ,ZZ9V',        23420000
323         DELETE_COUNTER               FIXED BINARY(15,0) INIT(0),   23430000
324         MORE_RECORDS                 BIT(1),                       23440000
325         NO                           BIT(1)    INIT('0'B),         23450000
326         RRN_EDIT                     PICTURE 'ZZ,ZZ9V',            23460000
327         YES                          BIT(1)    INIT('1'B);         23470000
328                                                                     23480000
329      DECLARE                                                        23490000
330         1 MAINT_RECORD,                                             23500000
331            2 MAINT_ACTION            CHAR(1),                      23510000
332            2 MAINT_SKIP              CHAR(1),                      23520000
333            2 MAINT_IMAGE             CHAR(78);                     23530000
334                                                                     23540000
335      DECLARE                                                        23550000
336         MAINT_RECORD_SCALAR          DEFINED MAINT_RECORD          23560000
337                                      CHAR(80);                     23570000
338                                                                     23580000
339      DECLARE                                                        23590000
340         1 VSAM_RECORD,                                              23600000
341            2 VSAM_RECORD_KEY,                                       23610000
342               3 VSAM_KEY_HI          CHAR(7),                      23620000
343               3 VSAM_KEY_RRN         CHAR(3),                      23630000
344            2 VSAM_RECORD_FIELDS      CHAR(70);                     23640000
345                                                                     23650000
346      DECLARE                                                        23660000
347         VSAM_RECORD_SCALAR           DEFINED VSAM_RECORD           23670000
348                                      CHAR(80);                     23680000
349                                                                     23690000
350      DECLARE                                                        23700000
351         1 PRINT_LINE,                                               23710000
352            2 CARRIAGE_CONTROL        CHAR(1)   INIT(' '),          23720000
353            2 PRINT_AREA              CHAR(132);                    23730000
354                                                                     23740000
355   %INCLUDE (VSAMIO);                                               23750000
356   %INCLUDE (VSAMIOFB);                                             23760000
357                                                                     23770000
358      END RRDSRND;                                                   23780000
```

MACRO SOURCE2 LISTING

INCLUDED TEXT FOLLOWS FROM DD.MEMBER = SYSLIB .VSAMIO

```
359    /*****************************************************************31100000
360                                                                      31110000
361           VV   VV    SSSSS       A      M     M    IIII      OOOOO       31120000
362           VV   VV   SS   SS     AAA     MM   MM     II      OO   OO      31130000
363           VV   VV   SS         AA AA    MMM MMM     II      OO   OO      31140000
364           VV   VV    SSSSS    AA   AA   MMMMMMM     II      OO   OO      31150000
365           VV   VV        SS   AA   AA   MM M MM     II      OO   OO      31160000
366            VV VV    SS   SS   AAAAAAA   MM   MM     II      OO   OO      31170000
367             VVV     SS   SS   AA   AA   MM   MM     II      OO   OO      31180000
368              V       SSSSS    AA   AA   MM   MM    IIII      OOOOO       31190000
369                                                                      31200000
370     ******************************************************************31210000
371      THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS  31220000
372      ROUTINE.                                                        31230000
373                                                                      31240000
374      THE VSIO_PARAMETER_VALUES SUPPLY THE VALUES USED TO MOVE INTO   31250000
375      PARAMETER ENTRIES TO TAILOR THE ROUTINE TO A SPECIFIC DATASET AND    31260000
376      TO PROVIDE COMMANDS TO DRIVE THE ROUTINE.                       31270000
377     ******************************************************************/31280000
378                                                                      31290000
379            DECLARE                                                   31300000
380               1 VSIO_PARAMETER_VALUES   STATIC,                      31310000
381                  2 VSIO_OPEN            CHAR(8)   INIT('OPEN    '),      31320000
382                  2 VSIO_CLOSE           CHAR(8)   INIT('CLOSE   '),      31330000
383                  2 VSIO_READ            CHAR(8)   INIT('READ    '),      31340000
384                  2 VSIO_WRITE           CHAR(8)   INIT('WRITE   '),      31350000
385                  2 VSIO_REWRITE         CHAR(8)   INIT('REWRITE '),      31360000
386                  2 VSIO_DELETE          CHAR(8)   INIT('DELETE  '),      31370000
387                  2 VSIO_START_EQUAL     CHAR(8)   INIT('STARTEQ '),      31380000
388                  2 VSIO_START_NOTLESS   CHAR(8)   INIT('STARTGE '),      31390000
389                  2 VSIO_KSDS            CHAR(4)   INIT('KSDS'),          31400000
390                  2 VSIO_ESDS            CHAR(4)   INIT('ESDS'),          31410000
391                  2 VSIO_RRDS            CHAR(4)   INIT('RRDS'),          31420000
392                  2 VSIO_SEQUENTIAL      CHAR(10)  INIT('SEQUENTIAL'),    31430000
393                  2 VSIO_DIRECT          CHAR(10)  INIT('DIRECT    '),    31440000
394                  2 VSIO_DYNAMIC         CHAR(10)  INIT('DYNAMIC   '),    31450000
395                  2 VSIO_INPUT           CHAR(6)   INIT('INPUT '),        31460000
396                  2 VSIO_OUTPUT          CHAR(6)   INIT('OUTPUT'),        31470000
397                  2 VSIO_INPUT_OUTPUT    CHAR(6)   INIT('UPDATE'),        31480000
398                  2 (VSIO_RC_SUCCESS               INIT(0),               31490000
399                     VSIO_RC_LOGIC_ERROR           INIT(8),               31500000
400                     VSIO_RC_END_OF_FILE           INIT(9999),            31510000
401                     VSIO_RC_UNKNOWN_COMMAND       INIT(20),              31520000
```

MACRO SOURCE2 LISTING

```
                    VSIO_RC_DATASET_ALREADY_OPEN  INIT(21),              31530000
                    VSIO_RC_DATASET_NOT_OPEN      INIT(22),              31540000
                    VSIO_RC_ORGANIZATION_UNKNOWN  INIT(23),              31550000
                    VSIO_RC_ACCESS_UNKNOWN        INIT(24),              31560000
                    VSIO_RC_ORG_ACCESS_MISMATCH   INIT(25),              31570000
                    VSIO_RC_MODE_UNKNOWN          INIT(26),              31580000
                    VSIO_RC_MODE_UNSUPPORTED      INIT(27),              31590000
                    VSIO_RC_DDNAME_BLANK          INIT(28))              31600000
                                        FIXED BINARY(15,0),              31610000
                 2 (VSIO_FB_DUPLICATE_RECORD      INIT(8),               31620000
                    VSIO_FB_KEY_SEQUENCE          INIT(12),              31630000
                    VSIO_FB_RECORD_NOT_FOUND      INIT(16),              31640000
                    VSIO_FB_NO_MORE_SPACE         INIT(28),              31650000
                    VSIO_FB_READ_WITHOUT_START    INIT(88))              31660000
                                        FIXED BINARY(15,0),              31670000
  /*****************************************************************31680000
    THE VSIO_PARAMETER_BLOCK IS THE COMMUNICATION INTERFACE TO THE       31690000
    THE ROUTINE.                                                         31700000
   *****************************************************************/31710000
                                                                         31720000
             1 VSIO_PARAMETER_BLOCK    STATIC,                           31730000
                2 VSIO_COMMAND         CHAR(8)   INIT(' '),              31740000
                2 (VSIO_RETURN_CODE,                                     31750000
                   VSIO_VSAM_RC,                                         31760000
                   VSIO_VSAM_FUNCTION,                                   31770000
                   VSIO_VSAM_FEEDBACK) FIXED BINARY(15,0) INIT(0);       31780000
                                                                         31790000
  /*****************************************************************31800000
                           END OF VSAMIO COPY BOOK                       31810000
   *****************************************************************/31820000
```

INCLUDED TEXT FOLLOWS FROM DD.MEMBER = SYSLIB .VSAMIOFB

```
  /*****************************************************************00000100
                                                                         00000200
     VV   VV   SSSSS      A     M     M  IIII   OOOOO   FFFFFFF  BBBBBB  00000300
     VV   VV  SS   SS    AAA    MM   MM   II   OO   OO  FF       BB   BB 00000400
     VV   VV  SS        AA AA   MMM MMM   II   OO   OO  FF       BB   BB 00000500
     VV   VV   SSSSS   AA   AA  MMMMMMM   II   OO   OO  FFFFF    BBBBBB  00000600
     VV   VV       SS  AA   AA  MM M MM   II   OO   OO  FF       BB   BB 00000700
      VV VV   SS   SS  AAAAAAA  MM   MM   II   OO   OO  FF       BB   BB 00000800
       VVV    SS   SS  AA   AA  MM   MM   II   OO   OO  FF       BB   BB 00000900
        V      SSSSS   AA   AA  MM   MM  IIII   OOOOO   FF       BBBBBB  00001000
                                                                         00001100
```

MACRO SOURCE2 LISTING

```
443     ******************************************************************00001200
444       THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS 00001300
445       ROUTINE, AND ARE USED TO COMMUNICATE CHARACTERISTICS FOR A SINGLE   00001400
446       VSAM DATASET.                                                       00001500
447                                                                           00001600
448       WITH THE 2 EXCEPTIONS FOR RECORD LENGTH (TO ACCOMODATE VARIABLE     00001700
449       LENGTH RECORDS) AND RELATIVE RECORD (TO ACCOMODATE RELATIVE RECORD  00001800
450       DATASETS), THESE DATA NAMES MUST BE POPULATED PRIOR TO CALLING THE  00001900
451       ROUTINE TO OPEN THE DATASET AND MUST NOT THEN BE CHANGED UNTIL THE  00002000
452       DATASET HAS BEEN CLOSED.                                            00002100
453     ******************************************************************/00002200
454                                                                           00002300
455          DECLARE                                                        00002400
456             1 VSIO_FILE_BLOCK            STATIC,                        00002500
457                2 VSFB_DDNAME             CHAR(8)   INIT(' '),           00002600
458                2 VSFB_ORGANIZATION       CHAR(4)   INIT(' '),           00002700
459                2 VSFB_ACCESS             CHAR(10)  INIT(' '),           00002800
460                2 VSFB_MODE               CHAR(6)   INIT(' '),           00002900
461                2 (VSFB_RECORD_LENGTH,                                   00003000
462                   VSFB_KEY_POSITION,                                    00003100
463                   VSFB_KEY_LENGTH)       FIXED BINARY(15,0) INIT(0),    00003200
464                2 VSFB_FILE_STATUS        CHAR(1)   INIT('C'),           00003300
465                2 VSFB_RESERVED           CHAR(161);                     00003400
466                                                                         00003500
467    /*****************************************************************00003600
468                          END OF VSAMIOFB COPY BOOK                     00003700
469     ******************************************************************/00003800
```

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

SOURCE LISTING.

```
                /*****************************************************************    1
                                                                                       1
                  RRDSRAND - TESTS THE VSAMIO ROUTINE BY READING AN RRDS CLUSTER        1
                             RANDOMLY AND ADDING/UPDATING/DELETING RECORDS.             1
                                                                                       1
                 *****************************************************************/   1
                                                                                       6
 1               RRDSRND:                                                              7
                    PROCEDURE OPTIONS(MAIN);                                           8
                                                                                       9
 2                     ON ERROR                                                     10
 3                        BEGIN;                                                    11
 4                           ON ERROR SYSTEM;                                       12
 5                           PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*'); 13
 6                           PUT SKIP DATA;                                         14
 7                           PUT SKIP(3) LIST((54)'*' || ' DEBUG AID ' || (54)'*'); 15
 8                        END;                                                      16
                                                                                     17
 9                     OPEN                                                        18
                          FILE(IMAGES),                                            19
                          FILE(PRINTR) LINESIZE(133);                              20
                                                                                   21
10                     ON ENDFILE(IMAGES)                                          22
11                        MORE_RECORDS = NO;                                       23
                                                                                   24
12                     PRINT_AREA = 'RRDSRAND: READ/UPDATE RRDS DIRECT';           25
13                     WRITE FILE(PRINTR) FROM(PRINT_LINE);                        26
14                     PRINT_AREA = '---------------------------------';           27
15                     WRITE FILE(PRINTR) FROM(PRINT_LINE);                        28
16                     PRINT_AREA = ' ';                                           29
17                     WRITE FILE(PRINTR) FROM(PRINT_LINE);                        30
                                                                                   31
18                     MORE_RECORDS = YES;                                         32
                                                                                   33
                  /*****************************************************************  34
                    ESTABLISH PARAMETERS FOR VSAM DATASET AND CALL ROUTINE TO OPEN      34
                   *****************************************************************/  34
                                                                                       36
19                     VSFB_DDNAME = 'RRDSF01';                                    37
20                     VSFB_ORGANIZATION = VSIO_RRDS;                              38
21                     VSFB_ACCESS = VSIO_DIRECT;                                  39
22                     VSFB_MODE = VSIO_INPUT_OUTPUT;                              40
23                     VSFB_RECORD_LENGTH = 80;                                    41
```

```
24          VSFB_KEY_POSITION = 0;
25          VSFB_KEY_LENGTH = 0;
26          VSIO_COMMAND = VSIO_OPEN;
27          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,
                          VSIO_FILE_BLOCK,
                          VSAM_RECORD);
28          IF (VSIO_RETURN_CODE ª= VSIO_RC_SUCCESS) THEN
29             DO;
30                CALL VSIO_ERROR;
31                RETURN;
32             END;

33          DO WHILE(MORE_RECORDS);
34             READ FILE(IMAGES) INTO(MAINT_RECORD);
35             IF (MORE_RECORDS) THEN
36                CALL PROCESS_MAINT;
37          END;

38          CLOSE FILE(IMAGES);

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

39          VSIO_COMMAND = VSIO_CLOSE;
40          CALL VSAMIOP (VSIO_PARAMETER_BLOCK,
                          VSIO_FILE_BLOCK,
                          VSAM_RECORD);
41          IF (VSIO_RETURN_CODE ª= VSIO_RC_SUCCESS) THEN
42             CALL VSIO_ERROR;

43          COUNTER_EDIT = ADD_COUNTER;
44          PRINT_AREA = COUNTER_EDIT || ' RECORDS ADDED';
45          WRITE FILE(PRINTR) FROM(PRINT_LINE);
46          COUNTER_EDIT = CHANGE_COUNTER;
47          PRINT_AREA = COUNTER_EDIT || ' RECORDS CHANGED';
48          WRITE FILE(PRINTR) FROM(PRINT_LINE);
49          COUNTER_EDIT = DELETE_COUNTER;
50          PRINT_AREA = COUNTER_EDIT || ' RECORDS DELETED';
51          WRITE FILE(PRINTR) FROM(PRINT_LINE);

52          RETURN;

53       PROCESS_MAINT:
            PROCEDURE;

54          RECORD_COUNTER = RECORD_COUNTER + 1;
```

```
55          IF (MAINT_ACTION = 'A') THEN
56             CALL PROCESS_ADD;
57          ELSE
57             IF (MAINT_ACTION = 'C') THEN
58                CALL PROCESS_CHANGE;
59             ELSE
59                IF (MAINT_ACTION = 'D') THEN
60                   CALL PROCESS_DELETE;
61                ELSE
61                   DO;
62                      COUNTER_EDIT = RECORD_COUNTER;
63                      PRINT_AREA = COUNTER_EDIT || ': INVALID ACTION: ' ||
                                     MAINT_RECORD_SCALAR;
64                      WRITE FILE(PRINTR) FROM(PRINT_LINE);
65                   END;

66          RETURN;

67       END PROCESS_MAINT;

68    PROCESS_ADD:
         PROCEDURE;

69          VSAM_RECORD_SCALAR = MAINT_IMAGE;
70          VSFB_KEY_LENGTH = VSAM_KEY_RRN;
71          RRN_EDIT = VSAM_KEY_RRN;
72          PRINT_AREA = 'RRN: ' || RRN_EDIT || '  DATA: ' ||
                         VSAM_RECORD_SCALAR || '  ADDING';
73          WRITE FILE(PRINTR) FROM(PRINT_LINE);
74          CALL WRITE_RR;
75          IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN
76             DO;
77                PRINT_AREA = (101)' ' || 'ADDED OK';
78                WRITE FILE(PRINTR) FROM(PRINT_LINE);
79             END;
80          ELSE
80             IF (VSIO_VSAM_FEEDBACK = VSIO_FB_DUPLICATE_RECORD) THEN
81                DO;
82                   PRINT_AREA = (101)' ' || '*DUPLICATE RECORD*';
83                   WRITE FILE(PRINTR) FROM(PRINT_LINE);
84                END;

85          RETURN;

86       END PROCESS_ADD;
```

```
 87             PROCESS_CHANGE:                                                  135
                   PROCEDURE;                                                    136
                                                                                 137
 88                   VSAM_RECORD_SCALAR = MAINT_IMAGE;                         138
 89                   VSFB_KEY_LENGTH = VSAM_KEY_RRN;                           139
 90                   RRN_EDIT = VSAM_KEY_RRN;                                  140
 91                   CALL READ_RR;                                             141
 92                   IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN              142
 93                      CALL COMPLETE_CHANGE;                                  143
 94                   ELSE                                                      144
 94                      DO;                                                    145
 95                         PRINT_AREA = 'READING FOR UPDATE: ' ||              146
                                         RRN_EDIT ||                             147
                                         '  *** NOT FOUND ***';                  148
 96                         WRITE FILE(PRINTR) FROM(PRINT_LINE);                149
 97                      END;                                                   150
                                                                                 151
 98                   RETURN;                                                   152
                                                                                 153
 99                END PROCESS_CHANGE;                                          154
                                                                                 155
100             COMPLETE_CHANGE:                                                156
                   PROCEDURE;                                                    157
                                                                                 158
101                   PRINT_AREA = 'RRN: ' || RRN_EDIT || '  DATA: ' ||         159
                                   VSAM_RECORD_SCALAR || '  BEFORE';            160
102                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                      161
103                   VSAM_RECORD_SCALAR = MAINT_IMAGE;                         162
104                   CALL REWRITE_RR;                                          163
105                   IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN              164
106                      DO;                                                    165
107                         PRINT_AREA = (19)' ' || VSAM_RECORD_SCALAR || '  AFTER';  166
108                         WRITE FILE(PRINTR) FROM(PRINT_LINE);                167
109                      END;                                                   168
                                                                                 169
110                   RETURN;                                                   170
                                                                                 171
111                END COMPLETE_CHANGE;                                         172
                                                                                 173
112             PROCESS_DELETE:                                                 174
                   PROCEDURE;                                                    175
                                                                                 176
113                   VSAM_RECORD_SCALAR = MAINT_IMAGE;                         177
114                   VSFB_KEY_LENGTH = VSAM_KEY_RRN;                           178
115                   RRN_EDIT = VSAM_KEY_RRN;                                  179
116                   CALL READ_RR;                                             180
117                   IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN              181
```

```
118                      CALL COMPLETE_DELETE;                                  182
119                   ELSE                                                      183
119                      DO;                                                    184
120                         PRINT_AREA = 'READING FOR DELETE: ' ||              185
                                         RRN_EDIT ||                           186
                                         '  *** NOT FOUND ***';                187
121                         WRITE FILE(PRINTR) FROM(PRINT_LINE);                188
122                      END;                                                   189
                                                                                190
123                   RETURN;                                                   191
                                                                                192
124                END PROCESS_DELETE;                                          193
                                                                                194
125             COMPLETE_DELETE:                                                195
                   PROCEDURE;                                                   196
                                                                                197
126                   PRINT_AREA = 'RRN: ' || RRN_EDIT || '  DATA: ' ||         198
                                   VSAM_RECORD_SCALAR || '  BEFORE';           199
127                   WRITE FILE(PRINTR) FROM(PRINT_LINE);                      200
128                   CALL DELETE_RR;                                           201
129                   IF (VSIO_RETURN_CODE = VSIO_RC_SUCCESS) THEN              202
130                      DO;                                                    203
131                         PRINT_AREA = (101)' ' || 'DELETED';                 204
132                         WRITE FILE(PRINTR) FROM(PRINT_LINE);                205
133                      END;                                                   206
                                                                                207
134                   RETURN;                                                   208
                                                                                209
135                END COMPLETE_DELETE;                                         210
                                                                                211
136             DELETE_RR:                                                      212
                   PROCEDURE;                                                   213
                                                                                214
                /*************************************************************** 215
                  CALL ROUTINE TO DELETE LAST RECORD READ FROM VSAM DATASET     215
                 ***************************************************************/ 215
                                                                                217
137                   VSIO_COMMAND = VSIO_DELETE;                               218
138                   CALL VSAMIOP (VSIO_PARAMETER_BLOCK,                       219
                                    VSIO_FILE_BLOCK,                            220
                                    VSAM_RECORD);                               221
139                   IF (VSIO_RETURN_CODE ª= VSIO_RC_SUCCESS) THEN             222
140                      CALL VSIO_ERROR;                                       223
141                   ELSE                                                      224
141                      DELETE_COUNTER = DELETE_COUNTER + 1;                   225
                                                                                226
142                   RETURN;                                                   227
```

```
                                                                          228
143                  END DELETE_RR;                                       229
                                                                          230
144               READ_RR:                                               231
                     PROCEDURE;                                           232
                                                                          233
                  /*************************************************************  234
                    CALL ROUTINE TO READ NEXT RECORD FROM VSAM DATASET    234
                   *************************************************************/  234
                                                                          236
145                     VSIO_COMMAND = VSIO_READ;                         237
146                     CALL VSAMIOP (VSIO_PARAMETER_BLOCK,               238
                                      VSIO_FILE_BLOCK,                    239
                                      VSAM_RECORD);                       240
147                     IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN     241
148                        IF (VSIO_VSAM_FEEDBACK ^= VSIO_FB_RECORD_NOT_FOUND) THEN  242
149                           CALL VSIO_ERROR;                            243
                                                                          244
150                     RETURN;                                           245
                                                                          246
151                  END READ_RR;                                         247
                                                                          248
152               REWRITE_RR:                                            249
                     PROCEDURE;                                           250
                                                                          251
                  /*************************************************************  252
                    CALL ROUTINE TO REWRITE PREVIOUSLY READ RECORD TO VSAM DATASET  252
                   *************************************************************/  252
                                                                          254
153                     VSIO_COMMAND = VSIO_REWRITE;                      255
154                     CALL VSAMIOP (VSIO_PARAMETER_BLOCK,               256
                                      VSIO_FILE_BLOCK,                    257
                                      VSAM_RECORD);                       258
155                     IF (VSIO_RETURN_CODE ^= VSIO_RC_SUCCESS) THEN     259
156                        CALL VSIO_ERROR;                               260
157                     ELSE                                              261
157                        CHANGE_COUNTER = CHANGE_COUNTER + 1;           262
                                                                          263
158                     RETURN;                                           264
                                                                          265
159                  END REWRITE_RR;                                      266
                                                                          267
160               WRITE_RR:                                              268
                     PROCEDURE;                                           269
                                                                          270
                  /*************************************************************  271
                    CALL ROUTINE TO WRITE RECORD INTO VSAM DATASET        271
```

```
                   *********************************************************************/
  161                    VSIO_COMMAND = VSIO_WRITE;
  162                    CALL VSAMIOP (VSIO_PARAMETER_BLOCK,
                                      VSIO_FILE_BLOCK,
                                      VSAM_RECORD);
  163                    IF (VSIO_RETURN_CODE ª= VSIO_RC_SUCCESS) THEN
  164                       IF (VSIO_VSAM_FEEDBACK ª= VSIO_FB_DUPLICATE_RECORD) THEN
  165                          CALL VSIO_ERROR;
  166                       ELSE
  166                          ;
  167                    ELSE
  167                       ADD_COUNTER = ADD_COUNTER + 1;

  168                    RETURN;

  169                 END WRITE_RR;

  170              VSIO_ERROR:
                      PROCEDURE;
  171                    PRINT_AREA = 'VSAMIO ERROR OCCURRED DURING ' ||
                                      VSIO_COMMAND;
  172                    WRITE FILE(PRINTR) FROM(PRINT_LINE);
  173                    PRINT_AREA = 'VSIO_RETURN_CODE = ' ||
                                      VSIO_RETURN_CODE;
  174                    WRITE FILE(PRINTR) FROM(PRINT_LINE);
  175                    PRINT_AREA = 'VSIO_VSAM_RETURN_CODE = ' ||
                                      VSIO_VSAM_RETURN_CODE;
  176                    WRITE FILE(PRINTR) FROM(PRINT_LINE);
  177                    PRINT_AREA = 'VSIO_VSAM_FUNCTION_CODE = ' ||
                                      VSIO_VSAM_FUNCTION_CODE;
  178                    WRITE FILE(PRINTR) FROM(PRINT_LINE);
  179                    PRINT_AREA = 'VSIO_VSAM_FEEDBACK_CODE = ' ||
                                      VSIO_VSAM_FEEDBACK_CODE;
  180                    WRITE FILE(PRINTR) FROM(PRINT_LINE);
  181                    PRINT_AREA = ' ';

  182                    RETURN;

  183                 END VSIO_ERROR;

  184                 DECLARE
                         IMAGES FILE INPUT RECORD SEQUENTIAL EXTERNAL
                            ENV(F),
                         PRINTR FILE OUTPUT RECORD SEQUENTIAL EXTERNAL
                            ENV(F CTLASA);
```

```
185                DECLARE                                                    319
                      ADD_COUNTER                FIXED BINARY(15,0) INIT(0),  320
                      CHANGE_COUNTER             FIXED BINARY(15,0) INIT(0),  321
                      COUNTER_EDIT               PICTURE 'ZZ,ZZZ,ZZ9V',       322
                      DELETE_COUNTER             FIXED BINARY(15,0) INIT(0),  323
                      MORE_RECORDS               BIT(1),                      324
                      NO                         BIT(1)    INIT('0'B),        325
                      RRN_EDIT                   PICTURE 'ZZ,ZZ9V',           326
                      YES                        BIT(1)    INIT('1'B);        327
                                                                              328
186                DECLARE                                                    329
                      1 MAINT_RECORD,                                         330
                         2 MAINT_ACTION          CHAR(1),                     331
                         2 MAINT_SKIP            CHAR(1),                     332
                         2 MAINT_IMAGE           CHAR(78);                    333
                                                                              334
187                DECLARE                                                    335
                      MAINT_RECORD_SCALAR        DEFINED MAINT_RECORD         336
                                                 CHAR(80);                    337
                                                                              338
188                DECLARE                                                    339
                      1 VSAM_RECORD,                                          340
                         2 VSAM_RECORD_KEY,                                   341
                            3 VSAM_KEY_HI        CHAR(7),                     342
                            3 VSAM_KEY_RRN       CHAR(3),                     343
                         2 VSAM_RECORD_FIELDS    CHAR(70);                    344
                                                                              345
189                DECLARE                                                    346
                      VSAM_RECORD_SCALAR         DEFINED VSAM_RECORD          347
                                                 CHAR(80);                    348
                                                                              349
190                DECLARE                                                    350
                      1 PRINT_LINE,                                           351
                         2 CARRIAGE_CONTROL      CHAR(1)   INIT(' '),         352
                         2 PRINT_AREA            CHAR(132);                   353
                                                                              354
                  /*************************************************************359
                                                                              359
                        VV   VV    SSSSS       A      M     M    IIII      00000  359
                        VV   VV   SS   SS     AAA     MM   MM     II      00   00 359
                        VV   VV   SS         AA AA    MMM MMM     II      00   00 359
                        VV   VV    SSSSS    AA   AA   MMMMMMM     II      00   00 359
                        VV   VV        SS   AA   AA   MM M MM     II      00   00 359
                         VV VV    SS   SS   AAAAAAA   MM   MM     II      00   00 359
                          VVV     SS   SS   AA   AA   MM   MM     II      00   00 359
                           V       SSSSS    AA   AA   MM   MM    IIII      00000  359
                                                                              359
```

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

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

191               DECLARE
                     1 VSIO_PARAMETER_VALUES   STATIC,
                        2 VSIO_OPEN            CHAR(8)   INIT('OPEN    '),
                        2 VSIO_CLOSE           CHAR(8)   INIT('CLOSE   '),
                        2 VSIO_READ            CHAR(8)   INIT('READ    '),
                        2 VSIO_WRITE           CHAR(8)   INIT('WRITE   '),
                        2 VSIO_REWRITE         CHAR(8)   INIT('REWRITE '),
                        2 VSIO_DELETE          CHAR(8)   INIT('DELETE  '),
                        2 VSIO_START_EQUAL     CHAR(8)   INIT('STARTEQ '),
                        2 VSIO_START_NOTLESS   CHAR(8)   INIT('STARTGE '),
                        2 VSIO_KSDS            CHAR(4)   INIT('KSDS'),
                        2 VSIO_ESDS            CHAR(4)   INIT('ESDS'),
                        2 VSIO_RRDS            CHAR(4)   INIT('RRDS'),
                        2 VSIO_SEQUENTIAL      CHAR(10)  INIT('SEQUENTIAL'),
                        2 VSIO_DIRECT          CHAR(10)  INIT('DIRECT    '),
                        2 VSIO_DYNAMIC         CHAR(10)  INIT('DYNAMIC   '),
                        2 VSIO_INPUT           CHAR(6)   INIT('INPUT '),
                        2 VSIO_OUTPUT          CHAR(6)   INIT('OUTPUT'),
                        2 VSIO_INPUT_OUTPUT    CHAR(6)   INIT('UPDATE'),
                        2 (VSIO_RC_SUCCESS               INIT(0),
                           VSIO_RC_LOGIC_ERROR           INIT(8),
                           VSIO_RC_END_OF_FILE           INIT(9999),
                           VSIO_RC_UNKNOWN_COMMAND       INIT(20),
                           VSIO_RC_DATASET_ALREADY_OPEN  INIT(21),
                           VSIO_RC_DATASET_NOT_OPEN      INIT(22),
                           VSIO_RC_ORGANIZATION_UNKNOWN  INIT(23),
                           VSIO_RC_ACCESS_UNKNOWN        INIT(24),
                           VSIO_RC_ORG_ACCESS_MISMATCH   INIT(25),
                           VSIO_RC_MODE_UNKNOWN          INIT(26),
                           VSIO_RC_MODE_UNSUPPORTED      INIT(27),
                           VSIO_RC_DDNAME_BLANK          INIT(28))
                                               FIXED BINARY(15,0),
                        2 (VSIO_FB_DUPLICATE_RECORD      INIT(8),
                           VSIO_FB_KEY_SEQUENCE          INIT(12),
                           VSIO_FB_RECORD_NOT_FOUND      INIT(16),
                           VSIO_FB_NO_MORE_SPACE         INIT(28),
                           VSIO_FB_READ_WITHOUT_START    INIT(88))
                                               FIXED BINARY(15,0),
```

```
                    /*********************************************************************  417
                      THE VSIO_PARAMETER_BLOCK IS THE COMMUNICATION INTERFACE TO THE          417
                      THE ROUTINE.                                                            417
                     ********************************************************************/  417
                                                                                              420
                               1 VSIO_PARAMETER_BLOCK    STATIC,                              422
                                  2 VSIO_COMMAND         CHAR(8)   INIT(' '),                 423
                                  2 (VSIO_RETURN_CODE,                                        424
                                     VSIO_VSAM_RC,                                            425
                                     VSIO_VSAM_FUNCTION,                                      426
                                     VSIO_VSAM_FEEDBACK) FIXED BINARY(15,0) INIT(0);          427
                                                                                              428
                    /*********************************************************************  429
                                              END OF VSAMIO COPY BOOK                         429
                     ********************************************************************/  429
                    /*********************************************************************  432
                                                                                              432
                       VV   VV   SSSSS      A     M     M  IIII   OOOOO   FFFFFFF  BBBBBB     432
                       VV   VV  SS   SS    AAA    MM   MM   II   OO   OO  FF       BB   BB    432
                       VV   VV  SS        AA AA   MMM MMM   II   OO   OO  FF       BB   BB    432
                       VV   VV   SSSSS   AA   AA  MMMMMMM   II   OO   OO  FFFFF    BBBBBB     432
                       VV   VV       SS  AA   AA  MM M MM   II   OO   OO  FF       BB   BB    432
                        VV VV   SS   SS  AAAAAAA  MM   MM   II   OO   OO  FF       BB   BB    432
                         VVV    SS   SS  AA   AA  MM   MM   II   OO   OO  FF       BB   BB    432
                          V      SSSSS   AA   AA  MM   MM  IIII   OOOOO   FF       BBBBBB     432
                                                                                              432
                     *********************************************************************  432
                       THESE PARAMETERS ARE USED TO INTERFACE WITH THE VSAM DATASET ACCESS    432
                       ROUTINE, AND ARE USED TO COMMUNICATE CHARACTERISTICS FOR A SINGLE      432
                       VSAM DATASET.                                                          432
                                                                                              432
                       WITH THE 2 EXCEPTIONS FOR RECORD LENGTH (TO ACCOMODATE VARIABLE        432
                       LENGTH RECORDS) AND RELATIVE RECORD (TO ACCOMODATE RELATIVE RECORD     432
                       DATASETS), THESE DATA NAMES MUST BE POPULATED PRIOR TO CALLING THE     432
                       ROUTINE TO OPEN THE DATASET AND MUST NOT THEN BE CHANGED UNTIL THE     432
                       DATASET HAS BEEN CLOSED.                                               432
                     ********************************************************************/  432
                                                                                              453
   192                    DECLARE                                                             455
                             1 VSIO_FILE_BLOCK           STATIC,                              456
                                2 VSFB_DDNAME            CHAR(8)   INIT(' '),                 457
                                2 VSFB_ORGANIZATION      CHAR(4)   INIT(' '),                 458
                                2 VSFB_ACCESS            CHAR(10)  INIT(' '),                 459
                                2 VSFB_MODE              CHAR(6)   INIT(' '),                 460
                                2 (VSFB_RECORD_LENGTH,                                        461
                                   VSFB_KEY_POSITION,                                         462
                                   VSFB_KEY_LENGTH)      FIXED BINARY(15,0) INIT(0),          463
```

```
                        2 VSFB_FILE_STATUS        CHAR(1)   INIT('C'),              464
                        2 VSFB_RESERVED           CHAR(161);                        465
                                                                                    466
                     /*****************************************************************  467
                                          END OF VSAMIOFB COPY BOOK                   467
                      ****************************************************************/  467
                                                                                    357
193                    END RRDSRND;                                                 358
```

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
185 *********	ADD_COUNTER	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 43,167,167
190	CARRIAGE_CONTROL	IN PRINT_LINE,AUTOMATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER
185 *********	CHANGE_COUNTER	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 46,157,157
100	COMPLETE_CHANGE	ENTRY,DECIMAL,FLOAT(SINGLE) 93
125	COMPLETE_DELETE	ENTRY,DECIMAL,FLOAT(SINGLE) 118
185	COUNTER_EDIT	AUTOMATIC,UNALIGNED,DECIMAL,PICTURE(ZZ,ZZZ,ZZ9V) 43,44,46,47,49,50,62,63
185 *********	DELETE_COUNTER	AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 49,141,141
136	DELETE_RR	ENTRY,DECIMAL,FLOAT(SINGLE) 128
184	IMAGES	FILE,EXTERNAL,INPUT,RECORD,SEQUENTIAL,ENVIRONMENT(F) 9,10,34,38
186	MAINT_ACTION	IN MAINT_RECORD,AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 55,57,59
186	MAINT_IMAGE	IN MAINT_RECORD,AUTOMATIC,UNALIGNED,STRING(78),CHARACTER 69,88,103,113
186	MAINT_RECORD	AUTOMATIC,STRUCTURE 34
187	MAINT_RECORD_SCALAR	AUTOMATIC,DEFINED,UNALIGNED,STRING(80),CHARACTER 63
186	MAINT_SKIP	IN MAINT_RECORD,AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
185	MORE_RECORDS	AUTOMATIC,UNALIGNED,STRING(1),BIT

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		11,18,33,35
185	NO	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 11
190	PRINT_AREA	IN PRINT_LINE,AUTOMATIC,UNALIGNED,STRING(132),CHARACTER 12,14,16,44,47,50,63,72,77,82,95,101,107,120,126,131,171,173,175,177 179,181
190	PRINT_LINE	AUTOMATIC,STRUCTURE 13,15,17,45,48,51,64,73,78,83,96,102,108,121,127,132,172,174,176,178 180
184	PRINTR	FILE,EXTERNAL,OUTPUT,RECORD,SEQUENTIAL,ENVIRONMENT(F CTLASA) 9,13,15,17,45,48,51,64,73,78,83,96,102,108,121,127,132,172,174,176 178,180
68	PROCESS_ADD	ENTRY,DECIMAL,FLOAT(SINGLE) 56
87	PROCESS_CHANGE	ENTRY,DECIMAL,FLOAT(SINGLE) 58
112	PROCESS_DELETE	ENTRY,DECIMAL,FLOAT(SINGLE) 60
53	PROCESS_MAINT	ENTRY,DECIMAL,FLOAT(SINGLE) 36
144	READ_RR	ENTRY,DECIMAL,FLOAT(SINGLE) 91,116
	RECORD_COUNTER	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 54,54,62
152	REWRITE_RR	ENTRY,DECIMAL,FLOAT(SINGLE) 104
1	RRDSRND	ENTRY,DECIMAL,FLOAT(SINGLE)
185	RRN_EDIT	AUTOMATIC,UNALIGNED,DECIMAL,PICTURE(ZZ,ZZ9V) 71,72,90,95,101,115,120,126
	SYSPRINT	FILE,EXTERNAL 5,6,7

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
188		VSAM_KEY_HI	IN VSAM_RECORD_KEY IN VSAM_RECORD,AUTOMATIC,UNALIGNED,STRING(7), CHARACTER
188		VSAM_KEY_RRN	IN VSAM_RECORD_KEY IN VSAM_RECORD,AUTOMATIC,UNALIGNED,STRING(3), CHARACTER 70,71,89,90,114,115
188		VSAM_RECORD	AUTOMATIC,STRUCTURE 27,40,138,146,154,162
188		VSAM_RECORD_FIELDS	IN VSAM_RECORD,AUTOMATIC,UNALIGNED,STRING(70),CHARACTER
188		VSAM_RECORD_KEY	IN VSAM_RECORD,AUTOMATIC,STRUCTURE
189		VSAM_RECORD_SCALAR	AUTOMATIC,DEFINED,UNALIGNED,STRING(80),CHARACTER 69,72,88,101,103,107,113,126
		VSAMIOP	EXTERNAL,ENTRY,DECIMAL,FLOAT(SINGLE) 27,40,138,146,154,162
192		VSFB_ACCESS	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(10),CHARACTER 21
192		VSFB_DDNAME	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(8),CHARACTER 19
192		VSFB_FILE_STATUS	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(1),CHARACTER
192	*********	VSFB_KEY_LENGTH	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 25,70,89,114
192	*********	VSFB_KEY_POSITION	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 24
192		VSFB_MODE	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(6),CHARACTER 22
192		VSFB_ORGANIZATION	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(4),CHARACTER 20
192	*********	VSFB_RECORD_LENGTH	IN VSIO_FILE_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0) 23
192		VSFB_RESERVED	IN VSIO_FILE_BLOCK,STATIC,UNALIGNED,STRING(161),CHARACTER

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
191	VSIO_CLOSE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 39
191	VSIO_COMMAND	IN VSIO_PARAMETER_BLOCK,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 26,39,137,145,153,161,171
191	VSIO_DELETE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 137
191	VSIO_DIRECT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER 21
191	VSIO_DYNAMIC	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(10), CHARACTER
170	VSIO_ERROR	ENTRY,DECIMAL,FLOAT(SINGLE) 30,42,140,149,156,165
191	VSIO_ESDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER
191 *********	VSIO_FB_DUPLICATE_RECORD	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 80,164
191 *********	VSIO_FB_KEY_SEQUENCE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
191 *********	VSIO_FB_NO_MORE_SPACE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
191 *********	VSIO_FB_READ_WITHOUT_START	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
191 *********	VSIO_FB_RECORD_NOT_FOUND	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0) 148
192	VSIO_FILE_BLOCK	STATIC,STRUCTURE 27,40,138,146,154,162

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
191	VSIO_INPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER
191	VSIO_INPUT_OUTPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER 22
191	VSIO_KSDS	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4), CHARACTER
191	VSIO_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER 26
191	VSIO_OUTPUT	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(6), CHARACTER
191	VSIO_PARAMETER_BLOCK	STATIC,STRUCTURE 27,40,138,146,154,162
191	VSIO_PARAMETER_VALUES	STATIC,STRUCTURE
191 *********	VSIO_RC_ACCESS_UNKNOWN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
191 *********	VSIO_RC_DATASET_ALREADY_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
191 *********	VSIO_RC_DATASET_NOT_OPEN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
191 *********	VSIO_RC_DDNAME_BLANK	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
191 *********	VSIO_RC_END_OF_FILE	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
191 *********	VSIO_RC_LOGIC_ERROR	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
191 *********	VSIO_RC_MODE_UNKNOWN	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
191 *********	VSIO_RC_MODE_UNSUPPORTED	IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED

```
DCL NO.         IDENTIFIER                          ATTRIBUTES AND REFERENCES

                                                    (15,0)

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

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

191   ********* VSIO_RC_SUCCESS                      IN VSIO_PARAMETER_VALUES,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                    (15,0)
                                                    28,41,75,92,105,117,129,139,147,155,163

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

191             VSIO_READ                           IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                    CHARACTER
                                                    145

191   ********* VSIO_RETURN_CODE                     IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                    (15,0)
                                                    28,41,75,92,105,117,129,139,147,155,163,173

191             VSIO_REWRITE                        IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8),
                                                    CHARACTER
                                                    153

191             VSIO_RRDS                           IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(4),
                                                    CHARACTER
                                                    20

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

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

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

191   ********* VSIO_VSAM_FEEDBACK                   IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED
                                                    (15,0)
                                                    80,148,164

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

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		179
191 *********	VSIO_VSAM_FUNCTION	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
	VSIO_VSAM_FUNCTION_CODE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 177
191 *********	VSIO_VSAM_RC	IN VSIO_PARAMETER_BLOCK,STATIC,ALIGNED,INITIAL,BINARY,FIXED (15,0)
	VSIO_VSAM_RETURN_CODE	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 175
191	VSIO_WRITE	IN VSIO_PARAMETER_VALUES,STATIC,UNALIGNED,INITIAL,STRING(8), CHARACTER
		161
160	WRITE_RR	ENTRY,DECIMAL,FLOAT(SINGLE)
		74
185	YES	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT
		18

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
186	MAINT_RECORD	80
190	PRINT_LINE	133
188	VSAM_RECORD	80
192	VSIO_FILE_BLOCK	196
191	VSIO_PARAMETER_BLOCK	16
191	VSIO_PARAMETER_VALUES	158

STORAGE REQUIREMENTS.

THE STORAGE AREA FOR THE PROCEDURE LABELLED RRDSRND IS 672 BYTES LONG.

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

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

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

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED PROCESS_ADD IS 284 BYTES LONG.

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

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED COMPLETE_CHANGE IS 284 BYTES LONG.

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

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED COMPLETE_DELETE IS 284 BYTES LONG.

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

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

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

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

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

THE PROGRAM CSECT IS NAMED RRDSRND AND IS 4054 BYTES LONG.

THE STATIC CSECT IS NAMED RRDSRNDA AND IS 8976 BYTES LONG.

STATISTICS   MACRO RECORDS =   469,SOURCE RECORDS =   474,PROG TEXT STMNTS =   193,OBJECT BYTES =   4054

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN ON UNIT

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

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN ON UNIT

OFFSET (HEX) 0000 0048 0052
STATEMENT NO 11 11

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE PROCESS_MAINT

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

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE PROCESS_ADD

OFFSET (HEX) 0000 0044 0062 007C 0092 00C6 00DE 00E8 00F4 00F4 0104 011C 0120 012C 012C 013C 0154 0154 015A
STATEMENT NO 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE PROCESS_CHANGE

OFFSET (HEX) 0000 0048 0066 0080 0096 00A0 00AC 00BA 00BA 00E6 00FE 00FE 0104
STATEMENT NO 87 88 89 90 91 92 93 94 95 96 97 98 99

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE COMPLETE_CHANGE

OFFSET (HEX) 0000 0048 0084 009C 00B2 00BC 00C8 00C8 00F2 010A 010A 0110
STATEMENT NO 100 101 102 103 104 105 106 107 108 109 110 111

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE PROCESS_DELETE

OFFSET (HEX) 0000 0048 0066 0080 0096 00A0 00AC 00BA 00BA 00E6 00FE 00FE 0104
STATEMENT NO 112 113 114 115 116 117 118 119 120 121 122 123 124

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE COMPLETE_DELETE

OFFSET (HEX) 0000 0048 0084 009C 00A6 00B2 00B2 00C2 00DA 00DA 00E0
STATEMENT NO 125 126 127 128 129 130 131 132 133 134 135

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE DELETE_RR

```
OFFSET (HEX)  0000 0038 003E 005E 006A 0078 0088 008E
STATEMENT NO   136  137  138  139  140  141  142  143
```

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE READ_RR

```
OFFSET (HEX)  0000 0034 003A 005A 0066 0072 007C 0082
STATEMENT NO   144  145  146  147  148  149  150  151
```

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE REWRITE_RR

```
OFFSET (HEX)  0000 0038 003E 005E 006A 0078 0088 008E
STATEMENT NO   152  153  154  155  156  157  158  159
```

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE WRITE_RR

```
OFFSET (HEX)  0000 0038 003E 005E 006A 0076 0084 0088 0098 009E
STATEMENT NO   160  161  162  163  164  165  166  167  168  169
```

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   170  171  172  173  174  175  176  177  178  179  180  181  182  183
```

TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE RRDSRND

```
OFFSET (HEX)  0000 0134 0142 014C 015A 0166 017E 018A 01A2 01AE 01C6 01CC 01D2 01D8 01DE 01E4 01EA 01F0 01F6 01FC 0218
STATEMENT NO     1    2    9   10   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27   28

OFFSET (HEX)  0224 0224 022E 0234 0234 023C 0254 025C 0266 026A 0274 027A 0296 02A2 02AC 02C6 02E2 02FA 0314 0330 0348
STATEMENT NO    29   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48   49

OFFSET (HEX)  0362 037E 0396 039C
STATEMENT NO    50   51   52  193
```

COMPILER DIAGNOSTICS.

WARNINGS.

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

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

IEM1790I DATA CONVERSIONS WILL BE DONE BY SUBROUTINE CALL IN THE FOLLOWING STATEMENTS 62, 70, 71, 89, 90, 114, 115, 175, 177, 179.

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.
```

```
RRDSRAND: READ/UPDATE RRDS DIRECT
---------------------------------

RRN:    501  DATA: 9532957501 JOHN J GLASSMAN      7663 SEASIDE AVENUE        NEWPORT BEACH   CA     ADDING
                                                                                                     ADDED OK
RRN:    502  DATA: 1964475502 CAITLIN V BROCKTON   9540 PARKER COURT          KANSAS CITY     MO     ADDING
                                                                                                     ADDED OK
RRN:    503  DATA: 9441505503 SUE P MOORE          640 JACKSON STREET         LOS ANGELES     CA     ADDING
                                                                                                     ADDED OK
RRN:      1  DATA: 0045557001 LARRY E BENSON       4778 DESERT STREET         LA HABRA        CA     ADDING
                                                                                                     *DUPLICATE RECORD*
READING FOR UPDATE:    505  *** NOT FOUND ***
RRN:     28  DATA: 9966129028 JOYCE R NORTON       907 BRANDYWINE CIRCLE      SAN ANTONIO     TX     BEFORE
                   9966129028 ZELDA H LOCKETT      314 ROSA LINDA COURT       MONTGOMERY      AL     AFTER
RRN:    503  DATA: 9441505503 SUE P MOORE          640 JACKSON STREET         LOS ANGELES     CA     BEFORE
                   9441505503 CLARA B ALEXANDER    8427 PECAN VALLEY STREET   BOISE           ID     AFTER
READING FOR DELETE:    508  *** NOT FOUND ***
RRN:     66  DATA: 3129003066 BILL W BECK          1798 SEABREEZE AVENUE      BOSTON          MA     BEFORE
                                                                                                     DELETED
         3 RECORDS ADDED
         2 RECORDS CHANGED
         1 RECORDS DELETED
```