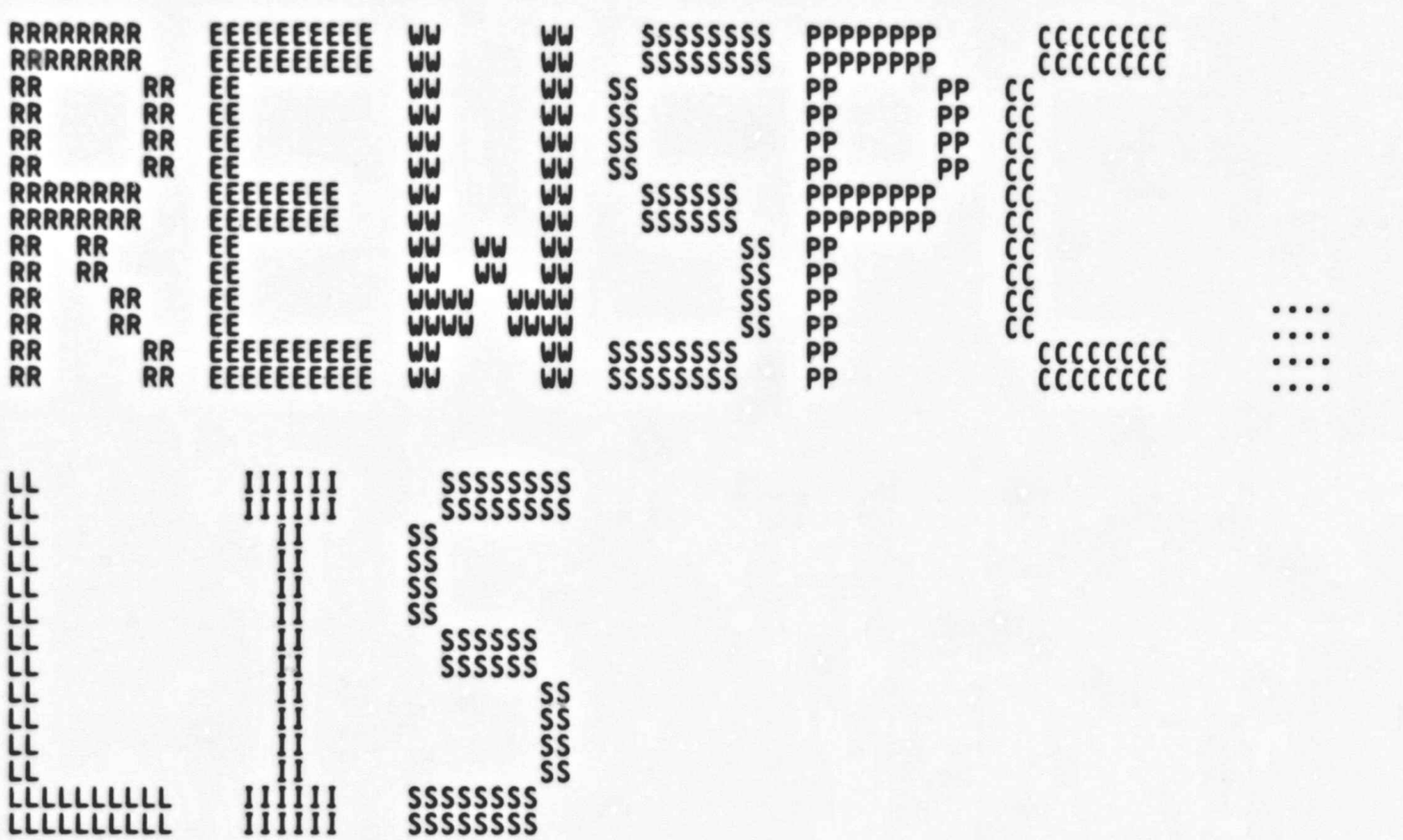

MMM MMM TTTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP

FILEID**REWSPC



```
;     1          0001  0
;     2          0002  0 MODULE REWSPC (LANGUAGE (BLISS32) ,
;     3          0003  0                IDENT = 'V04-000'
;     4          0004  0                ) =
;     5          0005  1 BEGIN
;     6          0006  1 !
;     7          0007  1 !**************************************************************************
;     8          0008  1 !*                                                                        *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                  *
;    12          0012  1 !*                                                                        *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    18          0018  1 !*  TRANSFERRED.                                                           *
;    19          0019  1 !*                                                                        *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;    22          0022  1 !*  CORPORATION.                                                           *
;    23          0023  1 !*                                                                        *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    26          0026  1 !*                                                                        *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !**************************************************************************
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 !
;    32          0032  1 ! FACILITY:  MTAACP
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       This module rewinds a file and spaces within a file
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:
;    39          0039  1 !
;    40          0040  1 !       STARLET operating system, including privileged system services
;    41          0041  1 !       and internal exec routines.
;    42          0042  1 !
;    43          0043  1 !--
;    44          0044  1
;    45          0045  1
;    46          0046  1 ! AUTHOR:  D. H. GILLESPIE,      CREATION DATE:  6-AUG-1977
;    47          0047  1 !
;    48          0048  1 ! MODIFIED BY:
;    49          0049  1 !
;    50          0050  1 !       V03-003 ROW0258         Ralph O. Weber          21-NOV-1983
;    51          0051  1 !                       The Paul Painter Memorial Enhancement
;    52          0052  1 !               Named for one of the unfortunate customers who suffered much
;    53          0053  1 !               to determine the great UCB$L_MT_RECORD secret while trying to
;    54          0054  1 !               create a user-written magtape driver, this change eliminates
;    55          0055  1 !               use of the device dependent field, UCB$L_MT_RECORD in favor of
;    56          0056  1 !               the device independent field, UCB$L_RECORD.
;    57          0057  1 !
```

```
;    58          0058  1 !       V03-002 MMD0147         Meg Dumont,     26-Apr-1983  8:48
;    59          0059  1 !               Change references to 80 to the symbol ANSI_LBLSZ
;    60          0060  1 !
;    61          0061  1 !       V03-001 MMD0001         Meg Dumont,     3-Jan-1983  16:13
;    62          0062  1 !               Add modifier IO$M_CLRSEREXCP to all QIO's issued by the MTAACP,
;    63          0063  1 !               necessary for the MSCP tape drives.
;    64          0064  1 !
;    65          0065  1 !       V02-004 REFORMAT        Maria del C. Nasr       30-Jun-1980
;    66          0066  1 !
;    67          0067  1 !**
;    68          0068  1
;    69          0069  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    70          0070  1
;    71          0071  1 REQUIRE 'SRC$:MTADEF.B32';
;    72          0455  1
;    73          0456  1 FORWARD ROUTINE
;    74          0457  1     REWIND_FILE         : COMMON_CALL NOVALUE,  ! main control for rewind file
;    75          0458  1     SPACE_IN_FILE       : COMMON_CALL NOVALUE,  ! space within a file
;    76          0459  1     SETUP_AT_END        : COMMON_CALL NOVALUE,  ! setup at end of file after checking position
;    77          0460  1     SETUP_END           : COMMON_CALL NOVALUE,  ! setup at end
;    78          0461  1     UPD_ST_RECORD       : COMMON_CALL NOVALUE;  ! update start record in current file section
;    79          0462  1
;    80          0463  1 EXTERNAL ROUTINE
;    81          0464  1     CLOSE_FILE          : L$CLOSE_FILE,         ! close file
;    82          0465  1     FORMAT_FID          : COMMON_CALL,          ! format file ID in current VCB
;    83          0466  1     GTNEXT_VOL_READ     : NOVALUE JSB,          ! get next volume on read
;    84          0467  1     MOUNT_VOL,                                  ! mount relative volume
;    85          0468  1     POSITION_BY_FID     : COMMON_CALL,          ! position by FID
;    86          0469  1     READ_BLOCK          : COMMON_CALL,          ! read data block
;    87          0470  1     RESTORE_ACCESS      : COMMON_CALL,          ! restore original access to file
;    88          0471  1     SPACE               : COMMON_CALL,          ! space blocks
;    89          0472  1     SPACE_TM            : COMMON_CALL,          ! space tape marks
;    90          0473  1     SYS$QIOW            : ADDRESSING_MODE (ABSOLUTE);
;    91          0474  1
;    92          0475  1 EXTERNAL
;    93          0476  1     CURRENT_UCB : REF BBLOCK,                   ! address of current UCB
;    94          0477  1     CURRENT_WCB : REF BBLOCK,                   ! address of current window control block
;    95          0478  1     HDR1        : REF BBLOCK,                   ! address of HDR1 (EOF1) label
;    96          0479  1     HDR2        : REF BBLOCK,                   ! address of HDR2(EOF2) label
;    97          0480  1     LOCAL_FIB   : BBLOCK,                       ! copy of user's FIB
;    98          0481  1     IO_CHANNEL,
;    99          0482  1     IO_STATUS   : VECTOR [2],                   ! IO status
;   100          0483  1     USER_STATUS : VECTOR [2];                   ! status returned to user
;   101          0484  1
```

```
;  103     0485  1 GLOBAL ROUTINE REWIND_FILE : COMMON_CALL NOVALUE =
;  104     0486  1
;  105     0487  1 !++
;  106     0488  1 !
;  107     0489  1 ! FUNCTIONAL DESCRIPTION:
;  108     0490  1 !       This routine rewinds to the beginning of the current file. if
;  109     0491  1 !       the beginning is on another volume, that volume is mounted and
;  110     0492  1 !       positioned to the beginning of the files data area.  Once at the
;  111     0493  1 !       beginning the access to the file is reset to the original
;  112     0494  1 !       requested access.
;  113     0495  1 !
;  114     0496  1 ! CALLING SEQUENCE:
;  115     0497  1 !       REWIND_FILE()
;  116     0498  1 !
;  117     0499  1 ! INPUT PARAMETERS:
;  118     0500  1 !       none
;  119     0501  1 !
;  120     0502  1 ! IMPLICIT INPUTS:
;  121     0503  1 !       CURRENT_VCB - address of current volume control block
;  122     0504  1 !       CURRENT_WCB - address of current window control block
;  123     0505  1 !
;  124     0506  1 ! OUTPUT PARAMETERS:
;  125     0507  1 !       none
;  126     0508  1 !
;  127     0509  1 ! IMPLICIT OUTPUTS:
;  128     0510  1 !       none
;  129     0511  1 !
;  130     0512  1 ! ROUTINE VALUE:
;  131     0513  1 !       none
;  132     0514  1 !
;  133     0515  1 ! SIDE EFFECTS:
;  134     0516  1 !       file positioned to beginning
;  135     0517  1 !       original access restore
;  136     0518  1 !
;  137     0519  1 ! USER ERRORS:
;  138     0520  1 !       SS$_FILNOTACC - file not accessed
;  139     0521  1 !--
;  140     0522  1
;  141     0523  2     BEGIN
;  142     0524  2
;  143     0525  2     EXTERNAL REGISTER
;  144     0526  2         COMMON_REG;
;  145     0527  2
;  146     0528  2     LOCAL
;  147     0529  2         FID,                                    ! file identification
;  148     0530  2         SEQ,                                    ! file section sequence number
;  149     0531  2         TM,                                     ! tape marks
;  150     0532  2         VOL;                                    ! relative volume number
;  151     0533  2
;  152     0534  2     ! if file is not accessed give error
;  153     0535  2     !
;  154     0536  2
;  155     0537  2     IF .CURRENT_WCB EQL 0
;  156     0538  2     THEN
;  157     0539  2         ERR_EXIT(SS$_FILNOTACC);
;  158     0540  2
;  159     0541  2     ! if writing, then close out file
```

```
;  160     0542  2      !
;  161     0543  2
;  162     0544  2      IF NOT .CURRENT_WCB[WCB$V_READ]
;  163     0545  2      THEN
;  164     0546  2          CLOSE_FILE();
;  165     0547  2
;  166     0548  2      ! calculate which relative volume the beginning is on
;  167     0549  2      !
;  168     0550  2      SEQ = .CURRENT_VCB[VCB$W_CUR_SEQ];          ! file section number
;  169     0551  2
;  170     0552  2      IF .SEQ EQL 1
;  171     0553  2      THEN
;  172     0554  3          BEGIN                                   ! currently in first file section
;  173     0555  3
;  174     0556  3          IF .CURRENT_VCB[VCB$V_LOGICEOVS]
;  175     0557  3          THEN
;  176     0558  3              SPACE_TM(-4)                        ! write case
;  177     0559  3          ELSE
;  178     0560  4              BEGIN                               ! read case
;  179     0561  4
;  180     0562  4              ! number of tape marks into current file section
;  181     0563  4              !
;  182     0564  4              TM = .CURRENT_VCB[VCB$B_TM];
;  183     0565  4
;  184     0566  4              IF .TM EQL 0  AND  .HDR1[HD1$L_HD1LID] NEQ 'HDR1'
;  185     0567  4              THEN
;  186     0568  4                  TM = 3;
;  187     0569  4
;  188     0570  4              IF .TM GEQ 1
;  189     0571  4              THEN
;  190     0572  4
;  191     0573  4                  ! backspace to tape mark preceding start of data
;  192     0574  4                  !
;  193     0575  4                  SPACE_TM(-.TM);
;  194     0576  4
;  195     0577  3              END;
;  196     0578  3
;  197     0579  3          SPACE_TM(1);                            ! pass over TM
;  198     0580  3          HDR1[HD1$L_HD1LID] = 'HDR1';
;  199     0581  3
;  200     0582  3          IF HDR2[HD2$L_HD2LID] NEQ 0
;  201     0583  3          THEN
;  202     0584  3              HDR2[HD2$L_HD2LID] = 'HDR2';
;  203     0585  3
;  204     0586  3          END
;  205     0587  2      ELSE
;  206     0588  3          BEGIN
;  207     0589  3
;  208     0590  3          ! current file number and section
;  209     0591  3          !
;  210     0592  3          FID = .CURRENT_VCB[VCB$L_CUR_FID];
;  211     0593  3          FID<16, 16> = 1;                        ! want section one
;  212     0594  3          VOL = .CURRENT_VCB[VCB$B_CUR_RVN];      ! current volume
;  213     0595  3          VOL = .VOL - .SEQ + 1;                  ! calculate volume wanted
;  214     0596  3          POSITION_BY_FID(.FID, .VOL);            ! position to file section
;  215     0597  3
;  216     0598  3          IF .CURRENT_VCB[VCB$B_TM] EQL 0
```

```
;  217     0599  3         THEN
;  218     0600  3             SPACE_TM(1);
;  219     0601  3
;  220     0602  2         END;
;  221     0603  2
;  222     0604  2     KERNEL_CALL(RESTORE_ACCESS);          ! restore original access to the file
;  223     0605  1     END;


                                                        .TITLE  REWSPC
                                                        .IDENT  \V04-000\

                                                        .EXTRN  CLOSE_FILE, FORMAT_FID
                                                        .EXTRN  GTNEXT_VOL_READ
                                                        .EXTRN  MOUNT_VOL, POSITION_BY_FID
                                                        .EXTRN  READ_BLOCK, RESTORE_ACCESS
                                                        .EXTRN  SPACE, SPACE_TM
                                                        .EXTRN  SYS$QIOW, CURRENT_UCB
                                                        .EXTRN  CURRENT_WCB, HDR1
                                                        .EXTRN  HDR2, LOCAL_FIB
                                                        .EXTRN  IO_CHANNEL, IO_STATUS
                                                        .EXTRN  USER_STATUS, SYS$CMKRNL

                                                        .PSECT  $CODE$,NOWRT,2

                                     040C 00000         .ENTRY  REWIND_FILE, Save R2,R3,R10                 ; 0485
                        53     0000G  CF  9E 00002      MOVAB   SPACE_TM, R3                                ;
                               0000G  CF  D5 00007      TSTL    CURRENT_WCB                                 ; 0537
                                      04  12 0000B      BNEQ    1$                                          ;
                               00AC   8F  BF 0000D      CHMU    #172                                        ; 0539
                        50     0000G  CF  D0 00011 1$:  MOVL    CURRENT_WCB, R0                             ; 0544
                        03        0B  A0  E8 00016      BLBS    11(R0), 2$                                  ;
                                   0000G  30 0001A      BSBW    CLOSE_FILE                                  ; 0546
                        52        26  AB  3C 0001D 2$:  MOVZWL  38(CURRENT_VCB), SEQ                        ; 0550
                        01            52  D1 00021      CMPL    SEQ, #1                                     ; 0552
                                      46  12 00024      BNEQ    7$                                          ;
               05       0B            AB  01  E1 00026  BBC     #1, 11(CURRENT_VCB), 3$                     ; 0556
                                      7E  04  CE 0002B  MNEGL   #4, -(SP)                                   ; 0558
                                      1B  11 0002E      BRB     5$                                          ;
                        50        2E  AB  9A 00030 3$:  MOVZBL  46(CURRENT_VCB), TM                         ; 0564
                                      0E  12 00034      BNEQ    4$                                          ; 0566
           31524448     8F     0000G  DF  D1 00036      CMPL    @HDR1, #827475016                           ;
                                      03  13 0003F      BEQL    4$                                          ;
                        50            03  D0 00041      MOVL    #3, TM                                      ; 0568
                                      50  D5 00044 4$:  TSTL    TM                                          ; 0570
                                      06  15 00046      BLEQ    6$                                          ;
                        7E            50  CE 00048      MNEGL   TM, -(SP)                                   ; 0575
                        63            01  FB 0004B 5$:  CALLS   #1, SPACE_TM                                ;
                                      01  DD 0004E 6$:  PUSHL   #1                                          ; 0579
                        63            01  FB 00050      CALLS   #1, SPACE_TM                                ;
              0000G  DF  31524448     8F  D0 00053      MOVL    #827475016, @HDR1                           ; 0580
                        50     0000G  CF  D0 0005C      MOVL    HDR2, R0                                    ; 0582
                                      31  13 00061      BEQL    8$                                          ;
                     60  32524448     8F  D0 00063      MOVL    #844252232, (R0)                            ; 0584
                                      28  11 0006A      BRB     8$                                          ; 0552
                        51        24  AB  D0 0006C 7$:  MOVL    36(CURRENT_VCB), FID                        ; 0592
   51          10       10            01  F0 00070      INSV    #1, #16, #16, FID                           ; 0593
```

```
                                50      2F    AB  9A 00075          MOVZBL  47(CURRENT_VCB), VOL                    ; 0594
                          52    50            52  C3 00079          SUBL3   SEQ, VOL, R2                            ; 0595
                                50      01    A2  9E 0007D          MOVAB   1(R2), VOL                              ;
                                                50  DD 00081          PUSHL   VOL                                     ; 0596
                                                51  DD 00083          PUSHL   FID                                     ;
                             0000G  CF          02  FB 00085          CALLS   #2, POSITION_BY_FID                     ;
                                            2E  AB  95 0008A          TSTB    46(CURRENT_VCB)                         ; 0598
                                                05  12 0008D          BNEQ    8$                                      ;
                                                01  DD 0008F          PUSHL   #1                                      ; 0600
                                63              01  FB 00091          CALLS   #1, SPACE_TM                            ;
                                                7E  D4 00094 8$:       CLRL    -(SP)                                   ; 0604
                                                5E  DD 00096          PUSHL   SP                                      ;
                                        0000G   CF  9F 00098          PUSHAB  RESTORE_ACCESS                          ;
                          00000000G     9F          03  FB 0009C          CALLS   #3, @#SYS$CMKRNL                        ;
                                                    04  000A3          RET                                             ; 0605

; Routine Size:  164 bytes,    Routine Base:  $CODE$ + 0000

;  224          0606  1
```

```
;  226          0607  1 ROUTINE SETUP_END (TM) : COMMON_CALL NOVALUE =
;  227          0608  1
;  228          0609  1 !++
;  229          0610  1 !
;  230          0611  1 ! FUNCTIONAL DESCRIPTION:
;  231          0612  1 !       Setup at end of file
;  232          0613  1 !
;  233          0614  1 ! CALLING SEQUENCE:
;  234          0615  1 !       SETUP_END(ARG1)
;  235          0616  1 !
;  236          0617  1 ! INPUT PARAMETERS:
;  237          0618  1 !       ARG1 - number of tape marks to be spaced and direction
;  238          0619  1 !
;  239          0620  1 ! IMPLICIT INPUTS:
;  240          0621  1 !       CURRENT_UCB - address of current unit control block
;  241          0622  1 !       HDR1 - address of 'HDR1' and 'EOF1' label
;  242          0623  1 !
;  243          0624  1 ! OUTPUT PARAMETERS:
;  244          0625  1 !       none
;  245          0626  1 !
;  246          0627  1 ! IMPLICIT OUTPUTS:
;  247          0628  1 !       CURRENT_VCB[VCB$L_ST_RECORD]
;  248          0629  1 !
;  249          0630  1 ! ROUTINE VALUE:
;  250          0631  1 !       none
;  251          0632  1 !
;  252          0633  1 ! SIDE EFFECTS:
;  253          0634  1 !       none
;  254          0635  1 !
;  255          0636  1 !--
;  256          0637  1
;  257          0638  2     BEGIN
;  258          0639  2
;  259          0640  2     EXTERNAL REGISTER
;  260          0641  2         COMMON_REG;
;  261          0642  2
;  262          0643  2     EXTERNAL ROUTINE
;  263          0644  2         LIB$CVT_DTB     : ADDRESSING_MODE (ABSOLUTE);
;  264          0645  2
;  265          0646  2     LOCAL
;  266          0647  2         BLOCK;
;  267          0648  2
;  268          0649  2     SPACE_TM(.TM);      ! space to end of file, right before end date TM
;  269          0650  2
;  270          0651  2     ! setup as if trailers had not been read
;  271          0652  2     !
;  272          0653  2     HDR1[HD1$L_HD1LID] = 'HDR1';
;  273          0654  2
;  274          0655  2     IF HDR2[HD2$L_HD2LID] NEQ 0
;  275          0656  2     THEN
;  276          0657  2         HDR2[HD2$L_HD2LID] = 'HDR2';
;  277          0658  2
;  278          0659  2     IF NOT LIB$CVT_DTB(EO1$S_BLOCKCNT, HDR1[EO1$T_BLOCKCNT], BLOCK)
;  279          0660  2     THEN
;  280          0661  2         ERR_EXIT(SS$_BLOCKCNTERR);
;  281          0662  2
;  282          0663  2     BLOCK = .CURRENT_UCB[UCB$L_RECORD] - .BLOCK;
```

```
;  283          0664  2     KERNEL_CALL(UPD_ST_RECORD, .BLOCK);
;  284          0665  1     END;


                                                        .EXTRN  LIB$CVT_DTB

                                                0000 00000 SETUP_END:
                                                        .WORD   Save nothing                    ; 0607
                            5E                  04  C2 00002         SUBL2   #4, SP
                                        04      AC  DD 00005         PUSHL   TM                 ; 0649
                        0000G   CF              01  FB 00008         CALLS   #1, SPACE_TM
                        0000G   DF 31524448     8F  D0 0000D         MOVL    #827475016, @HDR1  ; 0653
                                50      0000G   CF  D0 00016         MOVL    HDR2, R0           ; 0655
                                                07  13 0001B         BEQL    1$
                                60 32524448     8F  D0 0001D         MOVL    #844252232, (R0)   ; 0657
                                                5E  DD 00024 1$:     PUSHL   SP                 ; 0659
                    7E      0000G   CF              36  C1 00026     ADDL3   #54, HDR1, -(SP)
                                                06  DD 0002C         PUSHL   #6
                00000000G   9F                  03  FB 0002E         CALLS   #3, @#LIB$CVT_DTB
                            04                  50  E8 00035         BLBS    R0, 2$
                                        0940    8F  BF 00038         CHMU    #2368              ; 0661
                            50          0000G   CF  D0 0003C 2$:     MOVL    CURRENT_UCB, R0    ; 0663
                    6E      00B0        C0          6E  C3 00041     SUBL3   BLOCK, T76(R0), BLOCK
                                                6E  DD 00047         PUSHL   BLOCK              ; 0664
                                                01  DD 00049         PUSHL   #1
                                                5E  DD 0004B         PUSHL   SP
                                        0000V   CF  9F 0004D         PUSHAB  UPD_ST_RECORD
                00000000G   9F                  04  FB 00051         CALLS   #4, @#SYS$CMKRNL
                                                    04 00058         RET                        ; 0665

; Routine Size:  89 bytes,    Routine Base:  $CODE$ + 00A4
```

```
;  286        0666  1 ROUTINE UPD_ST_RECORD (BLOCK) : COMMON_CALL NOVALUE =
;  287        0667  1
;  288        0668  1 !++
;  289        0669  1 !
;  290        0670  1 ! FUNCTIONAL DESCRIPTION:
;  291        0671  1 !       This routine updates the start record count in the volume control block
;  292        0672  1 !       and sets the TM count to 1 because now positioned before end data TM
;  293        0673  1 !
;  294        0674  1 ! CALLING SEQUENCE:
;  295        0675  1 !       UPD_ST_RECORD(ARG1)
;  296        0676  1 !       called in kernel mode
;  297        0677  1 !
;  298        0678  1 ! INPUT PARAMETERS:
;  299        0679  1 !       ARG1 - new value of start record count
;  300        0680  1 !
;  301        0681  1 ! IMPLICIT INPUTS:
;  302        0682  1 !       CURRENT_VCB
;  303        0683  1 !
;  304        0684  1 ! OUTPUT PARAMETERS:
;  305        0685  1 !       none
;  306        0686  1 !
;  307        0687  1 ! IMPLICIT OUTPUTS:
;  308        0688  1 !       CURRENT_VCB[VCB$L_ST_RECORD] = BLOCK
;  309        0689  1 !
;  310        0690  1 ! ROUTINE VALUE:
;  311        0691  1 !       none
;  312        0692  1 !
;  313        0693  1 ! SIDE EFFECTS:
;  314        0694  1 !       none
;  315        0695  1 !
;  316        0696  1 !--
;  317        0697  1
;  318        0698  2     BEGIN
;  319        0699  2
;  320        0700  2     EXTERNAL REGISTER
;  321        0701  2         COMMON_REG;
;  322        0702  2
;  323        0703  2     CURRENT_VCB[VCB$B_TM] = 1;
;  324        0704  2     CURRENT_VCB[VCB$L_ST_RECORD] = .BLOCK;
;  325        0705  1     END;


                                      0000 00000 UPD_ST_RECORD:
                                                       .WORD   Save nothing                            ; 0666
                  2E   AB          01  90 00002        MOVB    #1, 46(CURRENT_VCB)                      ; 0703
                  30   AB      04  AC  D0 00006        MOVL    BLOCK, 48(CURRENT_VCB)                   ; 0704
                                       04 0000B        RET                                              ; 0705

; Routine Size:  12 bytes,    Routine Base:  $CODE$ + 00FD


;  326        0706  1
```

```
;  328     0707  1 GLOBAL ROUTINE SPACE_IN_FILE : COMMON_CALL NOVALUE =
;  329     0708  1
;  330     0709  1 !++
;  331     0710  1 !
;  332     0711  1 ! FUNCTIONAL DESCRIPTION:
;  333     0712  1 !       This routine spaces forwards and backwards within a file
;  334     0713  1 !
;  335     0714  1 ! CALLING SEQUENCE:
;  336     0715  1 !       SPACE_IN_FILE()
;  337     0716  1 !
;  338     0717  1 ! INPUT PARAMETERS:
;  339     0718  1 !       none
;  340     0719  1 !
;  341     0720  1 ! IMPLICIT INPUTS:
;  342     0721  1 !       CURRENT_VCB - address of current volume control block
;  343     0722  1 !       LOCAL_FIB - copy of user's file information block
;  344     0723  1 !
;  345     0724  1 ! OUTPUT PARAMETERS:
;  346     0725  1 !       none
;  347     0726  1 !
;  348     0727  1 ! IMPLICIT OUTPUTS:
;  349     0728  1 !       none
;  350     0729  1 !
;  351     0730  1 ! ROUTINE VALUE:
;  352     0731  1 !       none
;  353     0732  1 !
;  354     0733  1 ! SIDE EFFECTS:
;  355     0734  1 !       none
;  356     0735  1 !
;  357     0736  1 ! USER ERRORS:
;  358     0737  1 !       SS$_BEGOFFILE - beginning of file
;  359     0738  1 !       SS$_ENDOFFILE - end of file
;  360     0739  1 !       SS$_FILNOTACC - file not accessed
;  361     0740  1 !       SS$_BADPARAM - can not space forward if writing
;  362     0741  1 !       SS$_TAPEPOSLOST - tape position lost
;  363     0742  1 !--
;  364     0743  1
;  365     0744  2     BEGIN
;  366     0745  2
;  367     0746  2     EXTERNAL REGISTER
;  368     0747  2         COMMON_REG;
;  369     0748  2
;  370     0749  2     STACKLOCAL
;  371     0750  2         BLOCKS;                     ! number of blocks to space
;  372     0751  2
;  373     0752  2     LOCAL
;  374     0753  2         TM;                         ! number of tape marks into file section
;  375     0754  2
;  376     0755  2     ! file must be accessed
;  377     0756  2     !
;  378     0757  2
;  379     0758  2     IF .CURRENT_WCB EQL 0
;  380     0759  2     THEN
;  381     0760  2         ERR_EXIT(SS$_FILNOTACC);
;  382     0761  2
;  383     0762  2     BLOCKS = .LOCAL_FIB[FIB$L_CNTRLVAL];
;  384     0763  2
```

```
;  385     0764  2     IF .BLOCKS GTR 0
;  386     0765  2     THEN                                    ! sign determines direction to space
;  387     0766  2         BEGIN                               ! beginning of forward space
;  388     0767  2
;  389     0768  2         IF .BLOCKS<16, 16> NEQ 0
;  390     0769  2         THEN
;  391     0770  2             ERR_EXIT(SS$_BADPARAM);
;  392     0771  2
;  393     0772  2         ! can not space forward if writing
;  394     0773  2         !
;  395     0774  2
;  396     0775  2         IF NOT .CURRENT_WCB[WCB$V_READ]
;  397     0776  2         THEN
;  398     0777  2             ERR_EXIT(SS$_BADPARAM);
;  399     0778  2
;  400     0779  2         ! position to data in current file section
;  401     0780  2         !
;  402     0781  2
;  403     0782  2         IF  .CURRENT_VCB[VCB$B_TM] EQL 0
;  404     0783  2             AND
;  405     0784  2             .HDR1[HD1$L_HD1LID] EQL 'HDR1'
;  406     0785  2         THEN
;  407     0786  2             SPACE_TM(1);
;  408     0787  2
;  409     0788  2         IF .CURRENT_VCB[VCB$B_TM] NEQ 1
;  410     0789  2         THEN
;  411     0790  3             BEGIN
;  412     0791  3
;  413     0792  3             IF .CURRENT_VCB[VCB$B_TM] EQL 2
;  414     0793  3             THEN
;  415     0794  3                 TM = -1
;  416     0795  3             ELSE
;  417     0796  4                 BEGIN
;  418     0797  4
;  419     0798  4                 IF .CURRENT_VCB[VCB$V_LOGICEOVS]
;  420     0799  4                 THEN
;  421     0800  4                     TM = -3
;  422     0801  4                 ELSE
;  423     0802  4                     TM = -2;
;  424     0803  3                 END;
;  425     0804  3
;  426     0805  3             SPACE_TM(.TM);
;  427     0806  2             END;
;  428     0807  2
;  429     0808  2         WHILE 1
;  430     0809  2         DO
;  431     0810  3             BEGIN                                   ! forward space loop
;  432     0811  3
;  433     0812  3             IF SPACE(.BLOCKS)
;  434     0813  3             THEN
;  435     0814  3                 EXITLOOP;
;  436     0815  3
;  437     0816  3             USER_STATUS<16, 16> = .USER_STATUS<16, 16> + .IO_STATUS<16, 16> - 1;
;  438     0817  3             BLOCKS = .BLOCKS - .IO_STATUS<16, 16> + 1;  ! TM counts
;  439     0818  3
;  440     0819  3             IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
;  441     0820  3             THEN
```

```
;  442     0821  4                ERR_EXIT(SS$_TAPEPOSLOST);
;  443     0822  4
;  444     0823  4            IF .HDR1[HD1$L_HD1LID] EQL 'EOF1'
;  445     0824  4            THEN
;  446     0825  5                BEGIN
;  447     0826  5                SETUP_END(-1);
;  448     0827  5                KERNEL_CALL(RESTORE_ACCESS);
;  449     0828  5                ERR_EXIT(SS$_ENDOFFILE);
;  450     0829  4                END;
;  451     0830  4
;  452     0831  4            IF .HDR1[HD1$L_HD1LID] NEQ 'EOV1'
;  453     0832  4            THEN
;  454     0833  4                ERR_EXIT(SS$_TAPEPOSLOST);
;  455     0834  4
;  456     0835  4            GTNEXT_VOL_READ();          ! get next volume in volume set
;  457     0836  4
;  458     0837  4            IF .CURRENT_VCB[VCB$B_TM] EQL 0
;  459     0838  4            THEN
;  460     0839  4                SPACE_TM(1);
;  461     0840  4
;  462     0841  3            END;
;  463     0842  3
;  464     0843  3        END                                     ! end of forward space loop
;  465     0844  2    ELSE
;  466     0845  3        BEGIN                                   ! begin of backspace
;  467     0846  3
;  468     0847  3        IF .BLOCKS NEQ 0
;  469     0848  3        THEN
;  470     0849  3            BLOCKS = -(.BLOCKS);
;  471     0850  3
;  472     0851  3        IF .BLOCKS<15, 17> NEQ 0
;  473     0852  3        THEN
;  474     0853  3            ERR_EXIT(SS$_BADPARAM);
;  475     0854  3
;  476     0855  3        ! position to data if not there
;  477     0856  3        !
;  478     0857  3
;  479     0858  3        IF NOT .CURRENT_WCB[WCB$V_READ]
;  480     0859  3        THEN
;  481     0860  3            CLOSE_FILE();
;  482     0861  3
;  483     0862  3        IF .CURRENT_VCB[VCB$V_LOGICEOVS]
;  484     0863  3        THEN
;  485     0864  3            SETUP_END(-3)
;  486     0865  3        ELSE
;  487     0866  4            BEGIN                               ! read case
;  488     0867  4            TM = .CURRENT_VCB[VCB$B_TM];
;  489     0868  4
;  490     0869  4            IF .TM EQL 0  AND  .HDR1[HD1$L_HD1LID] EQL 'HDR1'
;  491     0870  4            THEN
;  492     0871  5                BEGIN
;  493     0872  5                SPACE_TM(1);
;  494     0873  5                KERNEL_CALL(RESTORE_ACCESS);
;  495     0874  5                ERR_EXIT(SS$_BEGOFFILE);
;  496     0875  4                END;
;  497     0876  4
;  498     0877  4            IF .TM EQL 0
```

```
;  499     0878  4              THEN
;  500     0879  4                  TM = 3;
;  501     0880  4
;  502     0881  4              IF .TM GEQ 2
;  503     0882  4              THEN
;  504     0883  4                  SETUP_END(-(.TM - 1));
;  505     0884  4
;  506     0885  3              END;
;  507     0886  3
;  508     0887  3          WHILE 1
;  509     0888  3          DO
;  510     0889  4              BEGIN                                   ! beginning of backspace loop
;  511     0890  4
;  512     0891  4              LOCAL
;  513     0892  4                  FID,
;  514     0893  4                  SEQ,
;  515     0894  4                  VOL;
;  516     0895  4
;  517     0896  4              IF SPACE(-(.BLOCKS))
;  518     0897  4              THEN
;  519     0898  4                  EXITLOOP;
;  520     0899  4
;  521     0900  4              USER_STATUS<16, 16> = .USER_STATUS<16, 16> + .IO_STATUS<16, 16> - 1;
;  522     0901  4
;  523     0902  4              ! calc number remaining to space
;  524     0903  4              !
;  525     0904  4              BLOCKS = .BLOCKS - .IO_STATUS<16, 16> + 1;
;  526     0905  4              FID = .CURRENT_VCB[VCB$L_CUR_FID];
;  527     0906  4              SEQ = .CURRENT_VCB[VCB$W_CUR_SEQ];
;  528     0907  4
;  529     0908  4              IF .SEQ EQL 1
;  530     0909  4              THEN                    ! is tape positioned at beginning of file?
;  531     0910  4
;  532     0911  5                  BEGIN
;  533     0912  5
;  534     0913  5                  ! space to beginning of data to avoid blocking virtual IO
;  535     0914  5                  !
;  536     0915  5                  SPACE_TM(1);
;  537     0916  5                  KERNEL_CALL(RESTORE_ACCESS);
;  538     0917  5                  ERR_EXIT(SS$_BEGOFFILE);
;  539     0918  4                  END;
;  540     0919  4
;  541     0920  4              ! previous volume number
;  542     0921  4              !
;  543     0922  4              VOL = .CURRENT_VCB[VCB$B_CUR_RVN] - 1;
;  544     0923  4              MOUNT_VOL(.VOL, $FIELDMASK(MOU$V_LBLCHECK));
;  545     0924  5              BEGIN
;  546     0925  5
;  547     0926  5              LOCAL
;  548     0927  5                  STATUS;
;  549     0928  5
;  550     0929  5              STATUS = SYS$QIOW(EFN, .IO_CHANNEL, IO$_SENSECHAR OR IO$M_CLSEREXCP,
;  551     0930  5                  IO_STATUS, 0,0,0,0,0,0,0,0);
;  552     0931  5
;  553     0932  5              IF .STATUS  AND  .IO_STATUS EQL SS$_ENDOFTAPE
;  554     0933  5              THEN
;  555     0934  6                  BEGIN
```

```
                    SPACE_TM(-3);
                    SPACE_TM(1);
                    SETUP_AT_END();
                    KERNEL_CALL(FORMAT_FID, CURRENT_VCB[VCB$L_CUR_FID]);
                    END
               ELSE
                    BEGIN
                    FID<16, 16> = .SEQ - 1;
                    POSITION_BY_FID(.FID, .VOL);
                    TM = 2 - .CURRENT_VCB[VCB$B_TM];
                    SPACE_TM(.TM);
                    SETUP_AT_END();
                    END;

               END;
               END;                                  ! end of while loop

          END;                                       ! end of forward and backward space

     USER_STATUS<16, 16> = .USER_STATUS<16, 16> + .IO_STATUS<16, 16>;
     KERNEL_CALL(RESTORE_ACCESS);
     END;
```

```
                                     07FC 00000        .ENTRY  SPACE_IN_FILE, Save R2,R3,R4,R5,R6,R7,R8,-   ; 0707
                                                               R9,R10                                       ;
              5A      0000G   CF  9E 00002             MOVAB   RESTORE_ACCESS, R10                          ;
              59      0000G   CF  9E 00007             MOVAB   HDR1, R9                                     ;
              58      0000G   CF  9E 0000C             MOVAB   IO_STATUS, R8                                ;
              57      0000G   CF  9E 00011             MOVAB   SPACE_TM, R7                                 ;
              56  00000000G   9F  9E 00016             MOVAB   @#SYS$CMKRNL, R6                             ;
              5E              04  C2 0001D             SUBL2   #4, SP                                       ;
                      0000G   CF  D5 00020             TSTL    CURRENT_WCB                                  ; 0758
                              04  12 00024             BNEQ    1$                                           ;
                      00AC    8F  BF 00026             CHMU    #172                                         ; 0760
              6E      0000G   CF  D0 0002A  1$:        MOVL    LOCAL_FIB+24, BLOCKS                         ; 0762
              50              6E  D0 0002F             MOVL    BLOCKS, R0                                   ; 0764
                              03  14 00032             BGTR    2$                                           ;
                            00BF  31 00034             BRW     15$                                          ;
                         02   AE  B5 00037  2$:        TSTW    BLOCKS+2                                     ; 0768
                              02  13 0003A             BEQL    3$                                           ;
                              14  BF 0003C             CHMU    #20                                          ; 0770
              50      0000G   CF  D0 0003E  3$:        MOVL    CURRENT_WCB, R0                              ; 0775
              02         0B   A0  E8 00043             BLBS    11(R0), 4$                                   ;
                              14  BF 00047             CHMU    #20                                          ; 0777
                         2E   AB  95 00049  4$:        TSTB    46(CURRENT_VCB)                              ; 0782
                              0F  12 0004C             BNEQ    5$                                           ;
  31524448    8F         00   B9  D1 0004E             CMPL    @HDR1, #827475016                            ; 0784
                              05  12 00056             BNEQ    5$                                           ;
                              01  DD 00058             PUSHL   #1                                           ; 0786
              67              01  FB 0005A             CALLS   #1, SPACE_TM                                 ;
              01         2E   AB  91 0005D  5$:        CMPB    46(CURRENT_VCB), #1                          ; 0788
                              1D  13 00061             BEQL    10$                                          ;
              02         2E   AB  91 00063             CMPB    46(CURRENT_VCB), #2                          ; 0792
```

```
                                    05  12 00067         BNEQ    6$                                  ;
                          52        01  CE 00069         MNEGL   #1, TM                              ; 0794
                                    0D  11 0006C         BRB     8$                                  ;
         05        0B     AB        01  E1 0006E 6$:     BBC     #1, 11(CURRENT_VCB), 7$             ; 0798
                          52        03  CE 00073         MNEGL   #3, TM                              ; 0800
                                    03  11 00076         BRB     8$                                  ;
                          52        02  CE 00078 7$:     MNEGL   #2, TM                              ; 0802
                                    52  DD 0007B 8$:     PUSHL   TM                                  ; 0805
                          67        01  FB 0007D 9$:     CALLS   #1, SPACE_TM                        ;
                                    6E  DD 00080 10$:    PUSHL   BLOCKS                              ; 0812
                 0000G    CF        01  FB 00082         CALLS   #1, SPACE                           ;
                          03        50  E9 00087         BLBC    R0, 11$                             ;
                                  018E  31 0008A         BRW     28$                                 ;
                          50 0000G  CF  3C 0008D 11$:    MOVZWL  USER_STATUS+2, R0                   ; 0816
                          51    02  A8  3C 00092         MOVZWL  IO_STATUS+2, R1                     ;
                          50        51  C0 00096         ADDL2   R1, R0                              ;
        0000G    CF       50        01  A3 00099         SUBW3   #1, R0, USER_STATUS+2               ;
                          50    02  A8  3C 0009F         MOVZWL  IO_STATUS+2, R0                     ; 0817
         50               6E        50  C3 000A3         SUBL3   R0, BLOCKS, R0                      ;
                          6E    01  A0  9E 000A7         MOVAB   1(R0), BLOCKS                       ;
                          7E    50  8F  9A 000AB         MOVZBL  #80, -(SP)                          ; 0819
                                    69  DD 000AF         PUSHL   HDR1                                ;
                 0000G    CF        02  FB 000B1         CALLS   #2, READ_BLOCK                      ;
                          04        50  E8 000B6         BLBS    R0, 12$                             ;
                              0224  8F  BF 000B9         CHMU    #548                                ; 0821
              31464F45    8F    00  B9  D1 000BD 12$:    CMPL    @HDR1, #826691397                   ; 0823
                                    15  12 000C5         BNEQ    13$                                 ;
                          7E        01  CE 000C7         MNEGL   #1, -(SP)                           ; 0826
                  FECC    CF        01  FB 000CA         CALLS   #1, SETUP_END                       ;
                                    7E  D4 000CF         CLRL    -(SP)                               ; 0827
                              4400  8F  BB 000D1         PUSHR   #^M<R10,SP>                         ;
                          66        03  FB 000D5         CALLS   #3, SYS$CMKRNL                      ;
                              0870  8F  BF 000D8         CHMU    #2160                               ; 0828
              31564F45    8F    00  B9  D1 000DC 13$:    CMPL    @HDR1, #827739973                   ; 0831
                                    04  13 000E4         BEQL    14$                                 ;
                              0224  8F  BF 000E6         CHMU    #548                                ; 0833
                                 0000G  30 000EA 14$:    BSBW    GTNEXT_VOL_READ                     ; 0835
                                2E  AB  95 000ED         TSTB    46(CURRENT_VCB)                     ; 0837
                                    8E  12 000F0         BNEQ    10$                                 ;
                                    01  DD 000F2         PUSHL   #1                                  ; 0839
                                    87  11 000F4         BRB     9$                                  ;
                                    03  13 000F6 15$:    BEQL    16$                                 ; 0847
                          6E        50  CE 000F8         MNEGL   R0, BLOCKS                          ; 0849
   00           01        AE  11        07  ED 000FB 16$:    CMPZV   #7, #17, BLOCKS+1, #0           ; 0851
                                    02  13 00101         BEQL    17$                                 ;
                                    14  BF 00103         CHMU    #20                                 ; 0853
                          50 0000G  CF  D0 00105 17$:    MOVL    CURRENT_WCB, R0                     ; 0858
                          03    0B  A0  E8 0010A         BLBS    11(R0), 18$                         ;
                                 0000G  30 0010E         BSBW    CLOSE_FILE                          ; 0860
         05        0B     AB        01  E1 00111 18$:    BBC     #1, 11(CURRENT_VCB), 19$            ; 0862
                          7E        03  CE 00116         MNEGL   #3, -(SP)                           ; 0864
                                    34  11 00119         BRB     22$                                 ;
                          52    2E  AB  9A 0011B 19$:    MOVZBL  46(CURRENT_VCB), TM                 ; 0867
                                    1C  12 0011F         BNEQ    20$                                 ; 0869
              31524448    8F    00  B9  D1 00121         CMPL    @HDR1, #827475016                   ;
                                    12  12 00129         BNEQ    20$                                 ;
                                    01  DD 0012B         PUSHL   #1                                  ; 0872
```

```
                67          01  FB 0012D        CALLS   #1, SPACE_TM                  ;
                            7E  D4 00130        CLRL    -(SP)                         ; 0873
                      4400  8F  BB 00132        PUSHR   #^M<R10,SP>                   ;
                66          03  FB 00136        CALLS   #3, SYS$CMKRNL                ;
                      0938  8F  BF 00139        CHMU    #2360                         ; 0874
                            52  D5 0013D  20$:  TSTL    TM                            ; 0877
                            03  12 0013F        BNEQ    21$                           ;
                52          03  D0 00141        MOVL    #3, TM                        ; 0879
                02          52  D1 00144  21$:  CMPL    TM, #2                        ; 0881
                            0B  19 00147        BLSS    23$                           ;
                        FF  A2  9F 00149        PUSHAB  -1(TM)                        ; 0883
                6E          6E  CE 0014C        MNEGL   (SP), (SP)                    ;
           FE47 CF          01  FB 0014F  22$:  CALLS   #1, SETUP_END                 ;
                7E          6E  CE 00154  23$:  MNEGL   BLOCKS, -(SP)                 ; 0896
          0000G CF          01  FB 00157        CALLS   #1, SPACE                     ;
                03          50  E9 0015C        BLBC    R0, 24$                       ;
                        00B9    31 0015F        BRW     28$                           ;
                50    0000G CF  3C 00162  24$:  MOVZWL  USER_STATUS+2, R0             ; 0900
                51       02 A8  3C 00167        MOVZWL  IO_STATUS+2, R1               ;
                50          51  C0 0016B        ADDL2   R1, R0                        ;
    0000G CF    50          01  A3 0016E        SUBW3   #1, R0, USER_STATUS+2         ;
                50       02 A8  3C 00174        MOVZWL  IO_STATUS+2, R0               ; 0904
          50    6E          50  C3 00178        SUBL3   R0, BLOCKS, R0                ;
                6E       01 A0  9E 0017C        MOVAB   1(R0), BLOCKS                 ;
                55       24 AB  D0 00180        MOVL    36(CURRENT_VCB), FID          ; 0905
                54       26 AB  3C 00184        MOVZWL  38(CURRENT_VCB), SEQ          ; 0906
                01          54  D1 00188        CMPL    SEQ, #1                       ; 0908
                            12  12 0018B        BNEQ    25$                           ;
                            01  DD 0018D        PUSHL   #1                            ; 0915
                67          01  FB 0018F        CALLS   #1, SPACE_TM                  ;
                            7E  D4 00192        CLRL    -(SP)                         ; 0916
                      4400  8F  BB 00194        PUSHR   #^M<R10,SP>                   ;
                66          03  FB 00198        CALLS   #3, SYS$CMKRNL                ;
                      0938  8F  BF 0019B        CHMU    #2360                         ; 0917
                53       2F AB  9A 0019F  25$:  MOVZBL  47(CURRENT_VCB), VOL          ; 0922
                            02  DD 001A3        PUSHL   #2                            ; 0923
                            73  9F 001A5        PUSHAB  -(VOL)                        ;
          0000G CF          02  FB 001A7        CALLS   #2, MOUNT_VOL                 ;
                            7E  7C 001AC        CLRQ    -(SP)                         ; 0929
                            7E  7C 001AE        CLRQ    -(SP)                         ;
                            7E  7C 001B0        CLRQ    -(SP)                         ;
                            7E  7C 001B2        CLRQ    -(SP)                         ;
                            58  DD 001B4        PUSHL   R8                            ;
                7E    021B  8F  3C 001B6        MOVZWL  #539, -(SP)                   ;
                      0000G CF  DD 001BB        PUSHL   IO_CHANNEL                    ;
                            01  DD 001BF        PUSHL   #1                            ;
      00000000G 9F          0C  FB 001C1        CALLS   #12, @#SYS$QIOW               ;
                29          50  E9 001C8        BLBC    STATUS, 26$                   ; 0932
       00000878 8F          68  D1 001CB        CMPL    IO_STATUS, #2168              ;
                            20  12 001D2        BNEQ    26$                           ;
                7E          03  CE 001D4        MNEGL   #3, -(SP)                     ; 0935
                67          01  FB 001D7        CALLS   #1, SPACE_TM                  ;
                            01  DD 001DA        PUSHL   #1                            ; 0936
                67          01  FB 001DC        CALLS   #1, SPACE_TM                  ;
          0000V CF          00  FB 001DF        CALLS   #0, SETUP_AT_END              ; 0937
                         24 AB  9F 001E4        PUSHAB  36(CURRENT_VCB)               ; 0938
                            01  DD 001E7        PUSHL   #1                            ;
```

```
                                 5E  DD 001E9          PUSHL    SP                                   ;
                      0000G      CF  9F 001EB          PUSHAB   FORMAT_FID                           ;
                      66         04  FB 001EF          CALLS    #4, SYS$CMKRNL                       ;
                                 24  11 001F2          BRB      27$                                  ; 0932
                      50     FF  A4  9E 001F4  26$:    MOVAB    -1(R4), R0                           ; 0942
          55      10  10         50  F0 001F8          INSV     R0, #16, #16, FID                    ;
                                 53  DD 001FD          PUSHL    VOL                                  ; 0943
                                 55  DD 001FF          PUSHL    FID                                  ;
                 0000G   CF      02  FB 00201          CALLS    #2, POSITION_BY_FID                  ;
                         52  2E  AB  9A 00206          MOVZBL   46(CURRENT_VCB), TM                  ; 0944
                      52         02  52  C3 0020A      SUBL3    TM, #2, TM                           ;
                                 52  DD 0020E          PUSHL    TM                                   ; 0945
                      67         01  FB 00210          CALLS    #1, SPACE_TM                         ;
                 0000V   CF      00  FB 00213          CALLS    #0, SETUP_AT_END                     ; 0946
                               FF39  31 00218  27$:    BRW      23$                                  ; 0887
                 0000G   CF  02  A8  A0 0021B  28$:    ADDW2    IO_STATUS+2, USER_STATUS+2           ; 0954
                                 7E  D4 00221          CLRL     -(SP)                                ; 0955
                      4400       8F  BB 00223          PUSHR    #^M<R10,SP>                          ;
                      66         03  FB 00227          CALLS    #3, SYS$CMKRNL                       ;
                                     04 0022A          RET                                           ; 0956

; Routine Size:  555 bytes,    Routine Base:  $CODE$ + 0109

;  578         0957  1
```

```
1 ROUTINE SETUP_AT_END : COMMON_CALL NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine makes the current file section current
1 !       and positions at end of this file section's data
1 !
1 ! CALLING SEQUENCE:
1 !       SETUP_AT_END()
1 !
1 ! INPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT INPUTS:
1 !       none
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       file section made current
1 !       start record of data section calculated
1 !
1 ! ROUTINE VALUE:
1 !       none
1 !
1 ! SIDE EFFECTS:
1 !       none
1 !
1 !--
1
2    BEGIN
2
2    EXTERNAL REGISTER
2        COMMON_REG;
2
2    IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
2    THEN
2        ERR_EXIT(SS$_TAPEPOSLOST);
2
2    IF .HDR1[EO1$L_EO1LID] NEQ 'EOV1'
2    THEN
2        ERR_EXIT(SS$_TAPEPOSLOST);
2
2    SETUP_END(-1);
1    END;
```

```
                         0000 00000 SETUP_AT_END:
                                     .WORD    Save nothing                      ; 0958
        7E      50    8F 9A 00002    MOVZBL   #80, -(SP)                        ; 0995
             0000G    CF DD 00006    PUSHL    HDR1                              ;
  0000G CF            02 FB 0000A    CALLS    #2, READ_BLOCK                    ;
```

```
                04              50 E8 0000F          BLBS    R0, 1$                          ;
                          0224  8F BF 00012          CHMU    #548                            ; 0997
       31564F45  8F       0000G DF D1 00016  1$:     CMPL    @HDR1, #827739973               ; 0999
                                04 13 0001F          BEQL    2$                              ;
                          0224  8F BF 00021          CHMU    #548                            ; 1001
                7E              01 CE 00025  2$:     MNEGL   #1, -(SP)                       ; 1003
           FD43  CF             01 FB 00028          CALLS   #1, SETUP_END                   ;
                                   04 0002D          RET                                     ; 1004

; Routine Size:  46 bytes,    Routine Base:  $CODE$ + 0334


;  627          1005  1 END
;  628          1006  1
;  629          1007  0 ELUDOM




;                                   PSECT SUMMARY
;
;      Name                    Bytes                        Attributes
;
;  $CODE$                           866  NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)




;                           Library Statistics
;
;                                         -------- Symbols --------       Pages       Processing
;      File                               Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619        28         0       1000         00:01.9




;                                   COMMAND QUALIFIERS

;      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:REWSPC/OBJ=OBJ$:REWSPC MSRC$:REWSPC/UPDATE=(ENH$:REWSPC)

; Size:           866 code + 0 data bytes
; Run Time:          00:17.9
; Elapsed Time:      00:37.9
; Lines/CPU Min:      3367
; Lexemes/CPU-Min: 17073
; Memory Used:  197 pages
; Compilation Complete
```

WTATTR
LIS

POSFID
LIS

RESACC
LIS

RWVOL
LIS

MTHRTL

STRVIO
LIS

MTHRTL
MAP

POSBE
LIS

RDATTR
LIS

REWSPC
LIS

UUMTHRTL
MAP