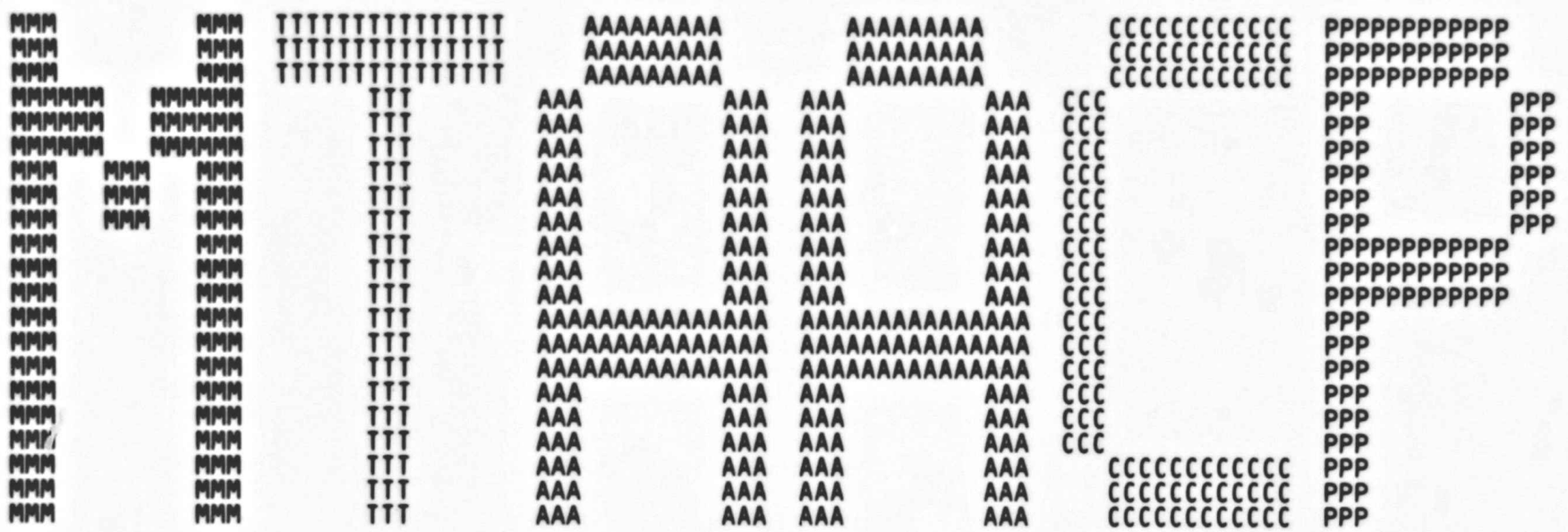
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
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 AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA 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**HEADER

```
HH      HH  EEEEEEEEEE    AAAAAA    DDDDDDDD    EEEEEEEEEE  RRRRRRRR
HH      HH  EEEEEEEEEE    AAAAAA    DDDDDDDD    EEEEEEEEEE  RRRRRRRR
HH      HH  EE           AA    AA   DD      DD  EE          RR      RR
HH      HH  EE           AA    AA   DD      DD  EE          RR      RR
HH      HH  EE           AA    AA   DD      DD  EE          RR      RR
HH      HH  EE           AA    AA   DD      DD  EE          RR      RR
HHHHHHHHHH  EEEEEEEE     AA    AA   DD      DD  EEEEEEEE    RRRRRRRR
HHHHHHHHHH  EEEEEEEE     AA    AA   DD      DD  EEEEEEEE    RRRRRRRR
HH      HH  EE           AAAAAAAAAA   DD      DD  EE          RR  RR
HH      HH  EE           AAAAAAAAAA   DD      DD  EE          RR  RR
HH      HH  EE           AA    AA   DD      DD  EE          RR    RR        ....
HH      HH  EE           AA    AA   DD      DD  EE          RR    RR        ....
HH      HH  EEEEEEEEEE   AA    AA   DDDDDDDD    EEEEEEEEEE  RR      RR      ....
HH      HH  EEEEEEEEEE   AA    AA   DDDDDDDD    EEEEEEEEEE  RR      RR      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0
;     2          0002  0 MODULE HEADER (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 !       This module contains routines which position to headers or trailers
;    36          0036  1 !       and read them.
;    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 !
;    47          0047  1 ! AUTHOR:  D. N. GILLESPIE,      CREATION DATE:  25-MAY-77 15:00
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;    51          0051  1 !       V03-006 MMD0323         Meg Dumont,     13-Aug-1984  15:17
;    52          0052  1 !               Fix to fix MMD0285, the way it was implemented the call
;    53          0053  1 !               wasn't getting made.
;    54          0054  1 !
;    55          0055  1 !       V03-005 MMD0300         Meg Dumont,     20-Jun-1984  11:23
;    56          0056  1 !               Fix to default Buffer offset length to zeros, when no HDR2
;    57          0057  1 !               is present for the file.
```

```
;    58          0058  1 !
;    59          0059  1 !       V03-004 MMD0285         Meg Dumont,     6-Apr-1984  17:18
;    60          0060  1 !               Fix to READ_HDR to incllude calling the clear
;    61          0061  1 !               seriuos exception routine after the headers are
;    62          0062  1 !               read. This is so that we do not leave the
;    63          0063  1 !               TMSCP drives left in serious exception state
;    64          0064  1 !               if we read into the TM while reading the headers.
;    65          0065  1 !
;    66          0066  1 !       V03-003 MMD0280         Meg Dumont,     23-Mar-1984  10:27
;    67          0067  1 !               Fix long file name support such that for ANSI version
;    68          0068  1 !               3 volumes it converts the exentsion length to
;    69          0069  1 !               ASCII characters before writing it to the label.
;    70          0070  1 !
;    71          0071  1 !       V03-002 ROW0258         Ralph O. Weber          21-NOV-1983
;    72          0072  1 !                       The Paul Painter Memorial Enhancement
;    73          0073  1 !               Named for one of the unfortunate customers who suffered much
;    74          0074  1 !               to determine the great UCB$L_MT_RECORD secret while trying to
;    75          0075  1 !               create a user-written magtape driver, this change eliminates
;    76          0076  1 !               use of the device dependent field, UCB$L_MT_RECORD in favor of
;    77          0077  1 !               the device independent field, UCB$L_RECORD.
;    78          0078  1 !
;    79          0079  1 !       V03-001 MMD0162         Meg Dumont,     26-Apr-1983  9:36
;    80          0080  1 !               Change reference to 80 to the symbol ANSI_LBLSZ. Change READ_HDR
;    81          0081  1 !               to read in the HDR4 label or if not found to default the values.
;    82          0082  1 !
;    83          0083  1 !       V02-010 REFORMAT        Maria del C. Nasr       30-Jun-1980
;    84          0084  1 !
;    85          0085  1 !       V02-009 MCN0016         Maria del C. Nasr,      18-Jun-1980  11:55
;    86          0086  1 !               Initialize default HDR2 with blanks, instead of zeroes, to
;    87          0087  1 !               avoid setting the old RMS attributes field.
;    88          0088  1 !
;    89          0089  1 !       A0008   MCN0013         Maria del C. Nasr       11-Mar-1980  11:25
;    90          0090  1 !               Check for HDR3 instead of HDR2 to determine if current file
;    91          0091  1 !               should be included in search or not.
;    92          0092  1 !
;    93          0093  1 !       A0007   MCN0011         Maria del C. Nasr       04-Feb-1980  9:05
;    94          0094  1 !               Add input parameter to UPDVCB_LEOV routine to either clear
;    95          0095  1 !               or set flag, and make routine global.
;    96          0096  1 !
;    97          0097  1 !       A0006   MCN0003         Maria del C. Nasr       28-Sep-79   10:39
;    98          0098  1 !               Add HDR3 processing
;    99          0099  1 !
;   100          0100  1 !       A0005   SPR24948        Maria del C. Nasr       11-Sep-79       17:30
;   101          0101  1 !               Forced spacing to eof when current position bit set to
;   102          0102  1 !               fix bug.
;   103          0103  1 !
;   104          0104  1 !
;   105          0105  1 !**
;   106          0106  1
;   107          0107  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;   108          0108  1
;   109          0109  1 REQUIRE 'SRC$:MTADEF.B32';
;   110          0493  1
;   111          0494  1 FORWARD ROUTINE
;   112          0495  1     READ_HDR            : COMMON_CALL,          ! read HDR1, HDR2, and HDR3 and HDR4 if exist
;   113          0496  1     SPACE_EOF           : COMMON_CALL NOVALUE,  ! space to end of file
;   114          0497  1     SET_NUMBER_OF_LABELS : COMMON_CALL NOVALUE, ! set the number of labels read
```

```
;  115     0498  1      UPDVCB_LEOV          : COMMON_CALL NOVALUE,  ! update VCB logical end of file
;  116     0499  1      MAKE_COR_FILE        : COMMON_CALL NOVALUE,  ! update VCB
;  117     0500  1      WRAP_AROUND          : L$WRAP_AROUND;        ! continue search at beginning of volume set  !
;  118     0501  1
;  119     0502  1  EXTERNAL
;  120     0503  1      CURRENT_UCB          : REF BBLOCK,
;  121     0504  1      IO_PACKET            : REF BBLOCK,           ! address of IO request packet
;  122     0505  1      HDR1                 : REF BBLOCK,           ! address HDR1 label
;  123     0506  1      HDR2                 : REF BBLOCK,           ! address of HDR2 label
;  124     0507  1      HDR3                 : REF BBLOCK,           ! address of HDR3 label
;  125     0508  1      HDR4                 : REF BBLOCK;           ! address of HDR4 label
;  126     0509  1
```

```
;  128     0510  1 GLOBAL ROUTINE GET_START_HDR : L$GET_START_HDR =
;  129     0511  1
;  130     0512  1 !++
;  131     0513  1 !
;  132     0514  1 ! FUNCTIONAL DESCRIPTION:
;  133     0515  1 !       This routine positions to the header label set of the start file
;  134     0516  1 !       in current search and reads HDR1, HDR2, HDR3 and HDR4 labels unless
;  135     0517  1 !       they have already been read.
;  136     0518  1 !
;  137     0519  1 ! CALLING SEQUENCE:
;  138     0520  1 !       GET_START_HDR()
;  139     0521  1 !
;  140     0522  1 ! INPUT PARAMETERS:
;  141     0523  1 !       none
;  142     0524  1 !
;  143     0525  1 ! IMPLICIT INPUTS:
;  144     0526  1 !       CURRENT_VCB, CURRENT_UCB
;  145     0527  1 !
;  146     0528  1 ! OUTPUT PARAMETERS:
;  147     0529  1 !       none
;  148     0530  1 !
;  149     0531  1 ! IMPLICIT OUTPUTS:
;  150     0532  1 !       HDR1 read in, HDR2 read in or defaulted, HDR3 read in or defaulted
;  151     0533  1 !       HDR4 read in or defaulted
;  152     0534  1 !
;  153     0535  1 ! ROUTINE VALUE:
;  154     0536  1 !       0 unsuccessful, logical end of volume set
;  155     0537  1 !       1 successful
;  156     0538  1 !
;  157     0539  1 ! SIDE EFFECTS:
;  158     0540  1 !       none
;  159     0541  1 !
;  160     0542  1 !--
;  161     0543  1
;  162     0544  2     BEGIN
;  163     0545  2
;  164     0546  2     EXTERNAL REGISTER
;  165     0547  2         COMMON_REG;
;  166     0548  2
;  167     0549  2     EXTERNAL ROUTINE
;  168     0550  2         MOUNT_VOL       : COMMON_CALL;          ! mount volume
;  169     0551  2
;  170     0552  2     EXTERNAL
;  171     0553  2         CURRENT_UCB     : REF BBLOCK,           ! address of current ucb
;  172     0554  2         LOCAL_FIB       : BBLOCK;                       ! copy of user's fib
;  173     0555  2
;  174     0556  2     LOCAL
;  175     0557  2         RELATIVE_BLOCK,                 ! relative block number to last tm
;  176     0558  2         TM;                             ! number of tm's
;  177     0559  2
;  178     0560  2     ! mount volume if the current relative volume number is zero
;  179     0561  2     !
;  180     0562  2
;  181     0563  2     IF .CURRENT_VCB[VCB$B_CUR_RVN] EQL 0
;  182     0564  2     THEN
;  183     0565  2         MOUNT_VOL(1, $FIELDMASK(MOU$V_REWIND) + $FIELDMASK(MOU$V_LBLCHECK));
;  184     0566  2
```

```
;  185     0567  2     ! if at logical end of volume set, return immediately
;  186     0568  2     !
;  187     0569  2
;  188     0570  2     IF .CURRENT_VCB[VCB$V_LOGICEOVS]
;  189     0571  2     THEN
;  190     0572  2         RETURN 0;
;  191     0573  2
;  192     0574  2     ! If the number of tape marks into the file is not 0, then the previous file
;  193     0575  2     ! was closed prematurely and should not be included in search except in the
;  194     0576  2     ! case where there is no HDR3 and the tape is left positioned beyond the
;  195     0577  2     ! tm. If the section is not the first, then space to next file
;  196     0578  2     !
;  197     0579  2
;  198     0580  3     IF (.CURRENT_VCB[VCB$B_TM] NEQU 0
;  199     0581  3         AND
;  200     0582  4         NOT (.CURRENT_VCB[VCB$B_TM] EQLU 1 AND .HDR3[HD3$L_HD3LID] NEQU 'HDR3'
;  201     0583  4         AND
;  202     0584  3         (.CURRENT_UCB[UCB$L_RECORD] - .CURRENT_VCB[VCB$L_ST_RECORD]) EQLU 0))
;  203     0585  2         OR
;  204     0586  2         .CURRENT_VCB[VCB$W_CUR_SEQ] GTR 1
;  205     0587  2     THEN
;  206     0588  2         SPACE_EOF()                             ! position to beginning of next file
;  207     0589  2     ELSE
;  208     0590  2
;  209     0591  2         ! If function is create, and current position bit is set, then force
;  210     0592  2         ! spacing to end of file, unless positioned in dummy file header set...
;  211     0593  2         !
;  212     0594  2
;  213     0595  3         IF ((.IO_PACKET[IRP$V_FCODE] EQL IO$_CREATE) AND .LOCAL_FIB[FIB$V_CURPOS])
;  214     0596  2             AND
;  215     0597  3             (.CURRENT_VCB[VCB$B_TM] NEQU 0) AND (.CURRENT_VCB[VCB$W_CUR_NUM] NEQU 0)
;  216     0598  2         THEN
;  217     0599  2             SPACE_EOF();
;  218     0600  2
;  219     0601  2     ! When new volume is mounted, VOL1 has been read but not the header labels.
;  220     0602  2     ! Therefore, the actual block count equals 1.  If relative block count = 0,
;  221     0603  2     ! then the headers have not been read for this file.
;  222     0604  2     !
;  223     0605  2     RELATIVE_BLOCK = .CURRENT_UCB[UCB$L_RECORD] - .CURRENT_VCB[VCB$L_ST_RECORD];
;  224     0606  2
;  225     0607  3     IF (.RELATIVE_BLOCK EQL 0 OR .CURRENT_UCB[UCB$L_RECORD] EQLU 1)
;  226     0608  2         AND
;  227     0609  2         .CURRENT_VCB[VCB$B_TM] EQLU 0
;  228     0610  2     THEN
;  229     0611  2         RETURN READ_HDR();
;  230     0612  2
;  231     0613  2     RETURN 1;
;  232     0614  2
;  233     0615  1     END;                                        ! end of routine


                        .TITLE  HEADER
                        .IDENT  \V04-000\

                        .EXTRN  CURRENT_UCB, IO_PACKET
                        .EXTRN  HDR1, HDR2, HDR3
                        .EXTRN  HDR4, MOUNT_VOL
```

```
                                                        .EXTRN  LOCAL_FIB

                                                        .PSECT  $CODE$,NOWRT,2

                                                5A  DD 00000 GET_START_HDR::
                                                            PUSHL   R10                                                 ; 0510
                                             2F     AB  95 00002            TSTB    47(CURRENT_VCB)                     ; 0563
                                                    09  12 00005            BNEQ    1$                                  ;
                                                    03  DD 00007            PUSHL   #3                                  ; 0565
                                                    01  DD 00009            PUSHL   #1                                  ;
                                          0000G     CF  02 FB 0000B            CALLS   #2, MOUNT_VOL                   ;
                                      73    0B      AB  01 E0 00010 1$:        BBS     #1, 11(CURRENT_VCB), 7$         ; 0570
                                                    5A  D4 00015               CLRL    R10                             ; 0580
                                             2E     AB  95 00017               TSTB    46(CURRENT_VCB)                 ;
                                                    20  13 0001A               BEQL    2$                              ;
                                                    5A  D6 0001C               INCL    R10                             ;
                                             01     2E  AB  91 0001E           CMPB    46(CURRENT_VCB), #1             ; 0582
                                                    39  12 00022               BNEQ    3$                              ;
                                      33524448      8F      0000G  DF  D1 00024   CMPL    @HDR3, #861029448               ;
                                                    2E  13 0002D               BEQL    3$                              ;
                                             50     0000G   CF  D0 0002F       MOVL    CURRENT_UCB, R0                 ; 0584
                                          30    AB      00B0    C0  D1 00034   CMPL    176(R0), 48(CURRENT_VCB)        ;
                                                    21  12 0003A               BNEQ    3$                              ;
                                             01     26  AB  B1 0003C 2$:       CMPW    38(CURRENT_VCB), #1             ; 0586
                                                    1B  1A 00040               BGTRU   3$                              ;
                                             50     0000G   CF  D0 00042       MOVL    IO_PACKET, R0                   ; 0595
                                33    20  A0              06  00  ED 00047   CMPZV   #0, #6, 32(R0), #51              ;
                                                    13  12 0004D               BNEQ    4$                              ;
                                          0D    0000G   CF      04  E1 0004F   BBC     #4, LOCAL_FIB, 4$               ;
                                                            0A  5A  E9 00055   BLBC    R10, 4$                         ; 0597
                                                    24  AB  B5 00058           TSTW    36(CURRENT_VCB)                 ;
                                                    05  13 0005B               BEQL    4$                              ;
                                          0000V     CF      00  FB 0005D 3$:   CALLS   #0, SPACE_EOF                   ; 0599
                                             50     0000G   CF  D0 00062 4$:   MOVL    CURRENT_UCB, R0                 ; 0605
                                      51    00B0    C0      30  AB  C3 00067   SUBL3   48(CURRENT_VCB), 176(R0), RELATIVE_BLOCK ;
                                                    07  13 0006E               BEQL    5$                              ; 0607
                                             01     00B0    C0  D1 00070       CMPL    176(R0), #1                     ;
                                                    0C  12 00075               BNEQ    6$                              ;
                                                    2E  AB  95 00077 5$:       TSTB    46(CURRENT_VCB)                 ; 0609
                                                    07  12 0007A               BNEQ    6$                              ;
                                          0000V     CF      00  FB 0007C       CALLS   #0, READ_HDR                    ; 0611
                                                    07  11 00081               BRB     8$                              ;
                                             50             01  D0 00083 6$:   MOVL    #1, R0                          ; 0613
                                                    02  11 00086               BRB     8$                              ;
                                                    50  D4 00088 7$:           CLRL    R0                              ; 0615
                                             5A             8E  D0 0008A 8$:   MOVL    (SP)+, R10                      ;
                                                            05  0008D          RSB                                     ;

; Routine Size:  142 bytes,    Routine Base:  $CODE$ + 0000

;  234          0616  1
```

```
;  236     0617  1 GLOBAL ROUTINE READ_HDR : COMMON_CALL =
;  237     0618  1
;  238     0619  1 !++
;  239     0620  1 !
;  240     0621  1 ! FUNCTIONAL DESCRIPTION:
;  241     0622  1 !       Read HDR1, and HDR2 if it exists - otherwise, it is defaulted.
;  242     0623  1 !       HDR3 is read only if HDR2 is found, and if starlet file. HDR4
;  243     0624  1 !       is read if the HDR3 is read.
;  244     0625  1 !
;  245     0626  1 ! CALLING SEQUENCE:
;  246     0627  1 !       READ_HDR()
;  247     0628  1 !
;  248     0629  1 ! INPUT PARAMETERS:
;  249     0630  1 !       none
;  250     0631  1 !
;  251     0632  1 ! IMPLICIT INPUTS:
;  252     0633  1 !       CURRENT_VCB - address of VCB
;  253     0634  1 !
;  254     0635  1 ! OUTPUT PARAMETERS:
;  255     0636  1 !       none
;  256     0637  1 !
;  257     0638  1 ! IMPLICIT OUTPUTS:
;  258     0639  1 !       HDR1, HDR2, HDR3 , and HDR4 read in
;  259     0640  1 !       If starlet file, VCB notes this fact
;  260     0641  1 !       Also the number of labels that the mtaacp found is set in the VCB
;  261     0642  1 !       If logical end of tape (ie:  tm encountered on read of HDR1) then this fact is noted in VCB
;  262     0643  1 !
;  263     0644  1 ! ROUTINE VALUE:
;  264     0645  1 !       0 - tm encountered when reading HDR1, logical end of volume set
;  265     0646  1 !       1 - successful
;  266     0647  1 !
;  267     0648  1 ! SIDE EFFECTS:
;  268     0649  1 !       first user label may be located in scratch label area
;  269     0650  1 !
;  270     0651  1 ! USER ERRORS:
;  271     0652  1 !       SS$_TAPEPOSLOST - HDR1 not encountered on read
;  272     0653  1 !
;  273     0654  1 !--
;  274     0655  1
;  275     0656  2     BEGIN
;  276     0657  2
;  277     0658  2     LOCAL
;  278     0659  2         MVL      : REF BBLOCK,
;  279     0660  2         NUMBER_OF_LABELS,
;  280     0661  2         SCRATCH  : REF BBLOCK,
;  281     0662  2         DESCR    : VECTOR [2,LONG];
;  282     0663  2
;  283     0664  2     EXTERNAL REGISTER
;  284     0665  2         COMMON_REG;
;  285     0666  2
;  286     0667  2     EXTERNAL ROUTINE
;  287     0668  2         CHCK_IO_CLR_EXCP : COMMON_CALL NOVALUE,
;  288     0669  2         ISSUE_IO         : L$ISSUE_IO,          ! Issue an IO to tape drive
;  289     0670  2         READ_BLOCK       : COMMON_CALL;         ! read one magtape block
;  290     0671  2
;  291     0672  2     BIND
;  292     0673  2         CVT5 = DESCRIPTOR('!5ZW'),
```

```
;  293     0674  2          DEFAULT = UPLIT ('00512');
;  294     0675  2
;  295     0676  2      ! Initialize the number of labels read. This number will eventually
;  296     0677  2      ! be stored in the VCB and will be used on volume switch and file close
;  297     0678  2      ! to determine the number of labels to write to the tape
;  298     0679  2
;  299     0680  2      NUMBER_OF_LABELS = 0;
;  300     0681  2      IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
;  301     0682  2      THEN
;  302     0683  3          BEGIN
;  303     0684  3          KERNEL_CALL(UPDVCB_LEOV, 1);
;  304     0685  3          RETURN 0;
;  305     0686  3
;  306     0687  2          END;
;  307     0688  2
;  308     0689  2      WHILE 1
;  309     0690  2      DO
;  310     0691  3          BEGIN
;  311     0692  3
;  312     0693  3          IF .HDR1[HD1$L_HD1LID] EQLU 'HDR1'
;  313     0694  3          THEN
;  314     0695  3              EXITLOOP;
;  315     0696  3
;  316     0697  3          IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
;  317     0698  3          THEN
;  318     0699  3              ERR_EXIT(SS$_TAPEPOSLOST);
;  319     0700  3
;  320     0701  2          END;
;  321     0702  2
;  322     0703  2      NUMBER_OF_LABELS = 1;
;  323     0704  2      KERNEL_CALL(MAKE_CUR_FILE, .HDR1);
;  324     0705  2
;  325     0706  2      ! Default HDR2, HDR3, and HDR4 values
;  326     0707  2
;  327     0708  2      CH$FILL(' ', ANSI_LBLSZ, .HDR2);
;  328     0709  2      CH$FILL(0, ANSI_LBLSZ, .HDR3);                  ! clear HDR3 area
;  329     0710  2      CH$FILL(' ',ANSI_LBLSZ, .HDR4);                 ! clear HDR4 area
;  330     0711  2
;  331     0712  2      ! Default the HDR4 fields according to the version type.
;  332     0713  2
;  333     0714  2      MVL = .CURRENT_VCB[VCB$L_MVL];
;  334     0715  2      IF .MVL[MVL$B_STDVER] GTR 3
;  335     0716  2          THEN
;  336     0717  2              HDR4[HD4$B_FILEID_EXT_SIZE] = 0             ! Default size to 0
;  337     0718  2          ELSE
;  338     0719  2              CH$FILL('0', HD4$S_FILEID_EXT_V3, HDR4[HD4$T_FILEID_EXT_V3]);
;  339     0720  2
;  340     0721  2      HDR2[HD2$B_RECFORMAT] = 'F';
;  341     0722  2      DESCR[0] = HD2$S_BLOCKLEN;
;  342     0723  2      DESCR[1] = HDR2[HD2$T_BLOCKLEN];
;  343     0724  2
;  344     0725  3      IF NOT $FAO(CVT5, 0, DESCR, .CURRENT_UCB[UCB$W_DEVBUFSIZ])
;  345     0726  2      THEN
;  346     0727  2          CH$MOVE(HD2$S_BLOCKLEN, DEFAULT, HDR2[HD2$T_BLOCKLEN]);
;  347     0728  2
;  348     0729  2      CH$MOVE(HD2$S_RECLEN, HDR2[HD2$T_BLOCKLEN], HDR2[HD2$T_RECLEN]);
;  349     0730  2
```

```
;  350          0731  2      IF .CURRENT_VCB[VCB$W_RECORDSZ] NEQ 0
;  351          0732  2      THEN
;  352          0733  3          BEGIN
;  353          0734  3          DESCR[0] = HD2$S_RECLEN;
;  354          0735  3          DESCR[1] = HDR2[HD2$T_RECLEN];
;  355          0736  3
;  356          0737  4          IF NOT $FAO(CVT5, 0, DESCR, .CURRENT_VCB[VCB$W_RECORDSZ])
;  357          0738  3          THEN
;  358          0739  3              CH$MOVE(HD2$S_RECLEN, HDR2[HD2$T_BLOCKLEN], HDR2[HD2$T_RECLEN]);
;  359          0740  3
;  360          0741  2          END;
;  361          0742  2
;  362          0743  2      ! Set up the default buffer offset length field. In case there
;  363          0744  2      ! is no HDR2 label
;  364          0745  2
;  365          0746  2          HDR2[HD2$T_BUFOFF] = '00';
;  366          0747  2
;  367          0748  2      ! Set up the Scratch area to read the labels into to determine if
;  368          0749  2      ! this is a good label, before copying it into the real label field.
;  369          0750  2
;  370          0751  2      SCRATCH = .HDR1 + SCRATCH_OFFSET;
;  371          0752  2
;  372          0753  2      ! Now try to read HDR2
;  373          0754  2      !
;  374          0755  2
;  375          0756  2      IF READ_BLOCK(.SCRATCH, ANSI_LBLSZ)          ! read into scratch area
;  376          0757  2      THEN
;  377          0758  2
;  378          0759  2          IF .(.SCRATCH) EQLU 'HDR2'
;  379          0760  2          THEN
;  380          0761  3              BEGIN
;  381          0762  3              CH$MOVE(ANSI_LBLSZ, .SCRATCH, .HDR2);        ! HDR2 found
;  382          0763  3              NUMBER_OF_LABELS = 2;
;  383          0764  3
;  384          0765  3              IF .CURRENT_VCB[VCB$V_STARFILE]      ! if starlet file
;  385          0766  3              THEN
;  386          0767  4              BEGIN
;  387          0768  4                  IF READ_BLOCK(.SCRATCH , ANSI_LBLSZ)     ! try to read HDR3
;  388          0769  4                  THEN
;  389          0770  5                  BEGIN
;  390          0771  5                      IF .(.SCRATCH) EQLU 'HDR3'
;  391          0772  5                      THEN
;  392          0773  6                      BEGIN
;  393          0774  6                          CH$MOVE(ANSI_LBLSZ, .SCRATCH, .HDR3);    ! HDR3 found
;  394          0775  6                          NUMBER_OF_LABELS = 3;
;  395          0776  5                      END;
;  396          0777  5                      IF READ_BLOCK(.SCRATCH , ANSI_LBLSZ)     ! try to read HDR4
;  397          0778  5                          THEN
;  398          0779  5                              IF .(.SCRATCH) EQLU 'HDR4'
;  399          0780  5                              THEN
;  400          0781  6                              BEGIN
;  401          0782  6                                  CH$MOVE(ANSI_LBLSZ, .SCRATCH, .HDR4);    ! HDR4 found
;  402          0783  6                                  NUMBER_OF_LABELS = 4;
;  403          0784  5                              END;
;  404          0785  4                  END;
;  405          0786  3              END;
;  406          0787  2              END;
```

```
;  407          0788  2
;  408          0789  2      ! Call to clear TMSCP drives of the serious exception (reading the tape
;  409          0790  2      ! mark) before returning to the user
;  410          0791  2
;  411          0792  2      CHCK_IO_CLR_EXCP();
;  412          0793  2      KERNEL_CALL(SET_NUMBER_OF_LABELS, NUMBER_OF_LABELS);
;  413          0794  2      RETURN 1;                                       ! return success
;  414          0795  2
;  415          0796  1      END;                                            ! end of routine


                                                57  5A  35  21  0008E P.AAB:  .ASCII  \!5ZW\                                ;
                                                                00092        .BLKB   2
                                                  00000004      00094 P.AAA:  .LONG   4                                       ;
                                                  00000000'     00098        .ADDRESS P.AAB                                  ;
                                            00  00  00  32  31  35  30  30  0009C P.AAC:  .ASCII  \00512\<0><0><0>           ;

                                                                            CVT5=           P.AAA
                                                                            DEFAULT=        P.AAC
                                                                                .EXTRN  CHCK_IO_CLR_EXCP
                                                                                .EXTRN  ISSUE_IO, READ_BLOCK
                                                                                .EXTRN  SYS$CMKRNL, SYS$FAO

                                                        07FC 00000              .ENTRY  READ_HDR, Save R2,R3,R4,R5,R6,R7,R8,R9,R10  ; 0617
                                              5A      0000G  CF   9E 00002      MOVAB   HDR2, R10                                   ;
                                              59 00000000G   9F   9E 00007      MOVAB   @#SYS$CMKRNL, R9                            ;
                                              5E             08   C2 0000E      SUBL2   #8, SP                                      ;
                                                             58   D4 00011      CLRL    NUMBER_OF_LABELS                            ; 0680
                                                             7E       50  8F  9A 00013  MOVZBL  #80, -(SP)                        ; 0681
                                                                0000G  CF   DD 00017    PUSHL   HDR1                            ;
                                                      0000G  CF             02  FB 0001B  CALLS   #2, READ_BLOCK                  ;
                                                             10             50  E8 00020  BLBS    R0, 1$                          ;
                                                                            01  DD 00023  PUSHL   #1                              ; 0684
                                                                            01  DD 00025  PUSHL   #1                              ;
                                                                            5E  DD 00027  PUSHL   SP                            ;
                                                                0000V  CF   9F 00029    PUSHAB  UPDVCB_LEOV                     ;
                                                                69             04  FB 0002D  CALLS   #4, SYS$CMKRNL               ;
                                                                          0165  31 00030  BRW     9$                            ; 0685
                                                   31524448     8F       0000G  DF  D1 00033 1$:  CMPL    @HDR1, #827475016     ; 0693
                                                                            16  13 0003C  BEQL    2$                            ;
                                                                7E       50  8F  9A 0003E  MOVZBL  #80, -(SP)                  ; 0697
                                                                0000G  CF   DD 00042    PUSHL   HDR1                            ;
                                                      0000G  CF             02  FB 00046  CALLS   #2, READ_BLOCK                  ;
                                                             E5             50  E8 0004B  BLBS    R0, 1$                          ;
                                                                0224   8F   BF 0004E    CHMU    #548                            ; 0699
                                                                            DF  11 00052  BRB     1$                            ; 0689
                                                                58          01  D0 00054 2$:  MOVL    #1, NUMBER_OF_LABELS    ; 0703
                                                                0000G  CF   DD 00057    PUSHL   HDR1                            ; 0704
                                                                            01  DD 0005B  PUSHL   #1                              ;
                                                                            5E  DD 0005D  PUSHL   SP                            ;
                                                                0000V  CF   9F 0005F    PUSHAB  MAKE_CUR_FILE                   ;
                                                                69             04  FB 00063  CALLS   #4, SYS$CMKRNL               ;
                                                                57             6A  D0 00066  MOVL    HDR2, R7                  ; 0708
                0050  8F              20                        6E       00  2C 00069  MOVC5   #0, (SP), #32, #80, (R7)       ;
                                                                                67  00070
                0050  8F              00                        6E       00  2C 00071  MOVC5   #0, (SP), #0, #80, @HDR3     ; 0709
```

```
                                0000G  DF     00078
                        56      0000G  CF  D0 0007B        MOVL    HDR4, R6                              ; 0710
 0050   8F        20    6E             00  2C 00080        MOVC5   #0, (SP), #32, #80, (R6)
                                       66     00087
                        50         34  AB  D0 00088        MOVL    52(CURRENT_VCB), MVL                  ; 0714
                        03         22  A0  91 0008C        CMPB    34(MVL), #3                           ; 0715
                                       05  1B 00090        BLEQU   3$
                                   04  A6  94 00092        CLRB    4(R6)                                 ; 0717
                                       06  11 00095        BRB     4$
                  43    A6       3030  8F  B0 00097 3$:    MOVW    #12336, 67(R6)                        ; 0719
                  04    A7         46  8F  90 0009D 4$:    MOVB    #70, 4(R7)                            ; 0721
                        6E             05  D0 000A2        MOVL    #5, DESCR                             ; 0722
                  04    AE         05  A7  9E 000A5        MOVAB   5(R7), DESCR+4                        ; 0723
                        50      0000G  CF  D0 000AA        MOVL    CURRENT_UCB, R0                       ; 0725
                        7E         42  A0  3C 000AF        MOVZWL  66(R0), -(SP)
                                   04  AE  9F 000B3        PUSHAB  DESCR
                                       7E  D4 000B6        CLRL    -(SP)
                                 FF34  CF  9F 000B8        PUSHAB  CVT5
            00000000G   00             04  FB 000BC        CALLS   #4, SYS$FAO
                        0A             50  E8 000C3        BLBS    R0, 5$
                        50             6A  D0 000C6        MOVL    HDR2, R0                              ; 0727
        05    A0  FF2A  CF             05  28 000C9        MOVC3   #5, DEFAULT, 5(R0)
                        57             6A  D0 000D0 5$:    MOVL    HDR2, R7                              ; 0729
        0A    A7    05  A7             05  28 000D3        MOVC3   #5, 5(R7), 10(R7)
                                   50  AB  B5 000D9        TSTW    80(CURRENT_VCB)                       ; 0731
                                       28  13 000DC        BEQL    6$
                        6E             05  D0 000DE        MOVL    #5, DESCR                             ; 0734
                  04    AE         0A  A7  9E 000E1        MOVAB   10(R7), DESCR+4                       ; 0735
                        7E         50  AB  3C 000E6        MOVZWL  80(CURRENT_VCB), -(SP)                ; 0737
                                   04  AE  9F 000EA        PUSHAB  DESCR
                                       7E  D4 000ED        CLRL    -(SP)
                                 FEFD  CF  9F 000EF        PUSHAB  CVT5
            00000000G   00             04  FB 000F3        CALLS   #4, SYS$FAO
                        09             50  E8 000FA        BLBS    R0, 6$
                        50             6A  D0 000FD        MOVL    HDR2, R0                              ; 0739
        0A    A0    05  A0             05  28 00100        MOVC3   #5, 5(R0), 10(R0)
                        50             6A  D0 00106 6$:    MOVL    HDR2, R0                              ; 0746
                  32    A0       3030  8F  B0 00109        MOVW    #12336, 50(R0)
              57     0000G  CF 00000140  8F  C1 0010F      ADDL3   #320, HDR1, SCRATCH                   ; 0751
                        7E         50  8F  9A 00119        MOVZBL  #80, -(SP)                            ; 0756
                                       57  DD 0011D        PUSHL   SCRATCH
                0000G   CF             02  FB 0011F        CALLS   #2, READ_BLOCK
                        5B             50  E9 00124        BLBC    R0, 8$
             32524448   8F             67  D1 00127        CMPL    (SCRATCH), #844252232                 ; 0759
                                       52  12 0012E        BNEQ    8$
        00    BA        67       0050  8F  28 00130        MOVC3   #80, (SCRATCH), @HDR2                 ; 0762
                        58             02  D0 00137        MOVL    #2, NUMBER_OF_LABELS                  ; 0763
                        44         2D  AB  E9 0013A        BLBC    45(CURRENT_VCB), 8$                   ; 0765
                        7E         50  8F  9A 0013E        MOVZBL  #80, -(SP)                            ; 0768
                                       57  DD 00142        PUSHL   SCRATCH
                0000G   CF             02  FB 00144        CALLS   #2, READ_BLOCK
                        36             50  E9 00149        BLBC    R0, 8$
             33524448   8F             67  D1 0014C        CMPL    (SCRATCH), #861029448                 ; 0771
                                       0B  12 00153        BNEQ    7$
     0000G    DF        67       0050  8F  28 00155        MOVC3   #80, (SCRATCH), @HDR3                 ; 0774
                        58             03  D0 0015D        MOVL    #3, NUMBER_OF_LABELS                  ; 0775
                        7E         50  8F  9A 00160 7$:    MOVZBL  #80, -(SP)                            ; 0777
```

```
                                                57  DD 00164          PUSHL   SCRATCH                                          ;
                                         0000G  CF                    02  FB 00166          CALLS   #2, READ_BLOCK                                   ;
                                                14                    50  E9 0016B          BLBC    R0, 8$                                           ;
                                    34524448    8F                    67  D1 0016E          CMPL    (SCRATCH), #877806664                            ; 0779
                                                                      0B  12 00175          BNEQ    8$                                               ;
                                0000G   DF              67      0050  8F  28 00177          MOVC3   #80, (SCRATCH), @HDR4                            ; 0782
                                                        58            04  D0 0017F          MOVL    #4, NUMBER_OF_LABELS                             ; 0783
                                         0000G  CF                    00  FB 00182  8$:     CALLS   #0, CHCK_IO_CLR_EXCP                             ; 0792
                                                                      58  DD 00187          PUSHL   NUMBER_OF_LABELS                                 ; 0793
                                                                      01  DD 00189          PUSHL   #1                                               ;
                                                                      5E  DD 0018B          PUSHL   SP                                               ;
                                                        0000V         CF  9F 0018D          PUSHAB  SET_NUMBER_OF_LABELS                             ;
                                                        69            04  FB 00191          CALLS   #4, SYS$CMKRNL                                   ;
                                                        50            01  D0 00194          MOVL    #1, R0                                           ; 0794
                                                                          04 00197          RET                                                      ;
                                                                      50  D4 00198  9$:     CLRL    R0                                               ; 0796
                                                                          04 0019A          RET                                                      ;

; Routine Size:  411 bytes,    Routine Base:  $CODE$ + 00A4

;  416          0797  1
```

```
;  418     0798  1 GLOBAL ROUTINE WRAP_AROUND : L$WRAP_AROUND =
;  419     0799  1
;  420     0800  1 !++
;  421     0801  1 !
;  422     0802  1 ! FUNCTIONAL DESCRIPTION:
;  423     0803  1 !       If this is not the first time through and the search started
;  424     0804  1 !       at the beginning of the volume set then return error else rewind volume set
;  425     0805  1 !
;  426     0806  1 ! CALLING SEQUENCE:
;  427     0807  1 !       WRAP_AROUND()
;  428     0808  1 !
;  429     0809  1 ! INPUT PARAMETERS:
;  430     0810  1 !       none
;  431     0811  1 !
;  432     0812  1 ! IMPLICIT INPUTS:
;  433     0813  1 !       LOCAL_FIB - copy of user's fib
;  434     0814  1 !       CURRENT_VCB - address of current volume VCB
;  435     0815  1 !
;  436     0816  1 ! OUTPUT PARAMETERS:
;  437     0817  1 !       none
;  438     0818  1 !
;  439     0819  1 ! IMPLICIT OUTPUTS:
;  440     0820  1 !       none
;  441     0821  1 !
;  442     0822  1 ! ROUTINE VALUE:
;  443     0823  1 !       0 back to beginning of search
;  444     0824  1 !       1 at beginning of volume set
;  445     0825  1 !
;  446     0826  1 ! SIDE EFFECTS:
;  447     0827  1 !       none
;  448     0828  1 !
;  449     0829  1 !--
;  450     0830  1
;  451     0831  2     BEGIN
;  452     0832  2
;  453     0833  2     EXTERNAL REGISTER
;  454     0834  2         COMMON_REG;
;  455     0835  2
;  456     0836  2     EXTERNAL ROUTINE
;  457     0837  2         MOUNT_VOL       : COMMON_CALL,          ! mount volume
;  458     0838  2         REWIND_VOL_SET  : COMMON_CALL;          ! rewind volume set
;  459     0839  2
;  460     0840  2     EXTERNAL
;  461     0841  2         LOCAL_FIB       : BBLOCK;               ! copy of user's fib
;  462     0842  2
;  463     0843  2     IF .CURRENT_VCB[VCB$L_START_FID] EQL %X'00010001'
;  464     0844  2     THEN
;  465     0845  2         RETURN 0
;  466     0846  2     ELSE
;  467     0847  3         BEGIN
;  468     0848  3         REWIND_VOL_SET();
;  469     0849  3
;  470     0850  3         ! get first volume mounted
;  471     0851  3         !
;  472     0852  3         MOUNT_VOL(1, $FIELDMASK(MOU$V_REWIND) + $FIELDMASK(MOU$V_LBLCHECK));
;  473     0853  3
;  474     0854  3         IF NOT READ_HDR()
```

```
;  475          0855  3          THEN
;  476          0856  3              ERR_EXIT(SS$_TAPEPOSLOST);
;  477          0857  3
;  478          0858  2          END;
;  479          0859  2
;  480          0860  2      RETURN 1;
;  481          0861  2
;  482          0862  1      END;                                    ! end of routine


                                                                     .EXTRN  REWIND_VOL_SET

                         00010001   8F       28   AB  D1 00000 WRAP_AROUND::
                                                                     CMPL    40(CURRENT_VCB), #65537          ; 0843
                                                    1E  13 00008     BEQL    2$                               ;
                             0000G  CF            00  FB 0000A     CALLS   #0, REWIND_VOL_SET               ; 0848
                                                    03  DD 0000F     PUSHL   #3                               ; 0852
                                                    01  DD 00011     PUSHL   #1                               ;
                             0000G  CF            02  FB 00013     CALLS   #2, MOUNT_VOL                    ;
                             FE48   CF            00  FB 00018     CALLS   #0, READ_HDR                     ; 0854
                                    04            50  E8 0001D     BLBS    R0, 1$                           ;
                                           0224   8F  BF 00020     CHMU    #548                             ; 0856
                                    50            01  D0 00024 1$:  MOVL    #1, R0                           ; 0860
                                                    05  00027     RSB                                      ;
                                                    50  D4 00028 2$:  CLRL    R0                               ; 0862
                                                    05  0002A     RSB                                      ;

; Routine Size:  43 bytes,    Routine Base:  $CODE$ + 023F

;  483          0863  1
```

```
;  485     0864  1 GLOBAL ROUTINE SPACE_EOF : COMMON_CALL NOVALUE =
;  486     0865  1
;  487     0866  1 !++
;  488     0867  1 !
;  489     0868  1 ! FUNCTIONAL DESCRIPTION:
;  490     0869  1 !       This routine spaces to the end of the current file, right
;  491     0870  1 !       before the next file.
;  492     0871  1 !
;  493     0872  1 ! CALLING SEQUENCE:
;  494     0873  1 !       SPACE_EOF()
;  495     0874  1 !
;  496     0875  1 ! INPUT PARAMETERS:
;  497     0876  1 !       none
;  498     0877  1 !
;  499     0878  1 ! IMPLICIT INPUTS:
;  500     0879  1 !       CURRENT_VCB _ address of current VCB
;  501     0880  1 !
;  502     0881  1 ! OUTPUT PARAMETERS:
;  503     0882  1 !       none
;  504     0883  1 !
;  505     0884  1 ! IMPLICIT OUTPUTS:
;  506     0885  1 !       none
;  507     0886  1 !
;  508     0887  1 ! ROUTINE VALUE:
;  509     0888  1 !       none
;  510     0889  1 !
;  511     0890  1 ! SIDE EFFECTS:
;  512     0891  1 !       The tape is left positioned in front of HDR1 of the next file
;  513     0892  1 !
;  514     0893  1 !--
;  515     0894  1
;  516     0895  2     BEGIN
;  517     0896  2
;  518     0897  2     SWITCHES NOOPTIMIZE;
;  519     0898  2
;  520     0899  2     EXTERNAL REGISTER
;  521     0900  2         COMMON_REG;
;  522     0901  2
;  523     0902  2     EXTERNAL ROUTINE
;  524     0903  2         GTNEXT_VOL_READ : JSB,                  ! get next volume on read
;  525     0904  2         READ_BLOCK      : COMMON_CALL,          ! read mag tape block
;  526     0905  2         SPACE_TM        : COMMON_CALL;          ! space tm's
;  527     0906  2
;  528     0907  2     EXTERNAL
;  529     0908  2         CURRENT_UCB     : REF BBLOCK;           ! address of current ucb
;  530     0909  2
;  531     0910  2     LOCAL
;  532     0911  2         TM;
;  533     0912  2
;  534     0913  2     ! If tape is positioned in header set, space 2 tape marks
;  535     0914  2     !
;  536     0915  2
;  537     0916  2     IF .CURRENT_VCB[VCB$B_TM] EQL 0 AND .HDR1[HD1$L_HD1LID] EQL 'HDR1'
;  538     0917  2     THEN
;  539     0918  2         SPACE_TM(2);
;  540     0919  2
;  541     0920  2     ! if in data area, space 1 tape mark
```

```
;  542          0921  2     !
;  543          0922  2
;  544          0923  2     IF .CURRENT_VCB[VCB$B_TM] EQLU 1
;  545          0924  2     THEN
;  546          0925  2         SPACE_TM(1);
;  547          0926  2
;  548          0927  2     ! Now if trailer label has not been read, read it
;  549          0928  2     !
;  550          0929  2
;  551          0930  2     IF .CURRENT_VCB[VCB$B_TM] EQLU 2
;  552          0931  2         AND
;  553          0932  2         (.CURRENT_UCB[UCB$L_RECORD] - .CURRENT_VCB[VCB$L_ST_RECORD]) EQL 0
;  554          0933  2     THEN
;  555          0934  2
;  556          0935  2         IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
;  557          0936  2         THEN
;  558          0937  2             ERR_EXIT(SS$_TAPEPOSLOST);
;  559          0938  2
;  560          0939  2     WHILE 1
;  561          0940  2     DO
;  562          0941  3         BEGIN
;  563          0942  3
;  564          0943  3         IF .HDR1[HD1$L_HD1LID] EQL 'EOF1'
;  565          0944  3         THEN
;  566          0945  3             EXITLOOP;
;  567          0946  3
;  568          0947  3         IF .HDR1[HD1$L_HD1LID] NEQ 'EOV1'
;  569          0948  3         THEN
;  570          0949  3             ERR_EXIT(SS$_TAPEPOSLOST);
;  571          0950  3
;  572          0951  3         GTNEXT_VOL_READ();
;  573          0952  3
;  574          0953  3         IF .CURRENT_VCB[VCB$B_TM] EQLU 0
;  575          0954  3         THEN
;  576          0955  3             SPACE_TM(2)
;  577          0956  3         ELSE
;  578          0957  3             SPACE_TM(1);
;  579          0958  3
;  580          0959  3         IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
;  581          0960  3         THEN
;  582          0961  3             ERR_EXIT(SS$_TAPEPOSLOST);
;  583          0962  3
;  584          0963  2         END;
;  585          0964  2
;  586          0965  2     IF .CURRENT_VCB[VCB$B_TM] EQLU 2
;  587          0966  2     THEN
;  588          0967  2         SPACE_TM(1);
;  589          0968  2
;  590          0969  1     END;                                    ! end of routine


                                                                    .EXTRN  GTNEXT_VOL_READ
                                                                    .EXTRN  SPACE_TM

                                                07FC 00000          .ENTRY  SPACE_EOF, Save R2,R3,R4,R5,R6,R7,R8,R9,R10 ; 0864
                                    52      0000G  CF   9E 00002    MOVAB   SPACE_TM, R2                                 ;
```

```
                    2E   AB  95 00007          TSTB    46(CURRENT_VCB)                 ; 0916
                    10   12  00009          BNEQ    1$
 31524448     8F        0000G DF  D1 0000C          CMPL    @HDR1, #827475016
                    05   12  00015          BNEQ    1$
                    02   DD  00017          PUSHL   #2                              ; 0918
                62  00       01  FB 00019          CALLS   #1, SPACE_TM
                01       2E  AB  91 0001C 1$:     CMPB    46(CURRENT_VCB), #1     ; 0923
                    05   12  00020          BNEQ    2$
                    01   DD  00022          PUSHL   #1                              ; 0925
                62  00       01  FB 00024          CALLS   #1, SPACE_TM
                02       2E  AB  91 00027 2$:     CMPB    46(CURRENT_VCB), #2     ; 0930
                    21   12  0002B          BNEQ    4$
                50  00       0000G CF  D0 0002D          MOVL    CURRENT_UCB, R0         ; 0932
              30    AB       00B0  C0  D1 00032          CMPL    176(R0), 48(CURRENT_VCB)
                    14   12  00038          BNEQ    4$
                7E  00       50  8F  9A 0003A 3$:     MOVZBL  #80, -(SP)              ; 0935
                00       0000G CF  DD 0003E          PUSHL   HDR1
            0000G   CF           02  FB 00042          CALLS   #2, READ_BLOCK
                04   50  E8  00047          BLBS    R0, 4$
                0224     8F  BF  0004A          CHMU    #548                    ; 0937
 31464F45     8F        0000G DF  D1 0004E 4$:     CMPL    @HDR1, #826691397       ; 0943
                    22   13  00057          BEQL    8$
 31564F45     8F        0000G DF  D1 00059          CMPL    @HDR1, #827739973       ; 0947
                    04   13  00062          BEQL    5$
                0224     8F  BF  00064          CHMU    #548                    ; 0949
                    0000G 30 00068 5$:     BSBW    GTNEXT_VOL_READ         ; 0951
                    2E   AB  95 0006B          TSTB    46(CURRENT_VCB)         ; 0953
                    04   12  0006E          BNEQ    6$
                    02   DD  00070          PUSHL   #2                              ; 0955
                    02   11  00072          BRB     7$
                    01   DD  00074 6$:     PUSHL   #1                              ; 0957
                62  00       01  FB 00076 7$:     CALLS   #1, SPACE_TM
                    BF   11  00079          BRB     3$                      ; 0959
                02  00   2E  AB  91 0007B 8$:     CMPB    46(CURRENT_VCB), #2     ; 0965
                    05   12  0007F          BNEQ    9$
                    01   DD  00081          PUSHL   #1                              ; 0967
                62  00       01  FB 00083          CALLS   #1, SPACE_TM
                    04   00086 9$:     RET                             ; 0969

; Routine Size:  135 bytes,    Routine Base:  $CODE$ + 026A

;  591          0970  1
```

```
;  593          0971  1 ROUTINE MAKE_CUR_FILE (LABELS) : COMMON_CALL NOVALUE =
;  594          0972  1
;  595          0973  1 !++
;  596          0974  1 !
;  597          0975  1 ! FUNCTIONAL DESCRIPTION:
;  598          0976  1 !       This routine updates the current file number and the Starlet
;  599          0977  1 !       file indicator.
;  600          0978  1 !
;  601          0979  1 ! CALLING SEQUENCE:
;  602          0980  1 !       MAKE_CUR_FILE(ARG1), call in kernel mode
;  603          0981  1 !
;  604          0982  1 ! INPUT PARAMETERS:
;  605          0983  1 !       ARG1 - address of labels
;  606          0984  1 !
;  607          0985  1 ! IMPLICIT INPUTS:
;  608          0986  1 !       none
;  609          0987  1 !
;  610          0988  1 ! OUTPUT PARAMETERS:
;  611          0989  1 !       none
;  612          0990  1 !
;  613          0991  1 ! IMPLICIT OUTPUTS:
;  614          0992  1 !       If file is Starlet file, then VCB$V_STARFILE = 1
;  615          0993  1 !       CUR_NUM is updated
;  616          0994  1 !
;  617          0995  1 ! ROUTINE VALUE:
;  618          0996  1 !       none
;  619          0997  1 !
;  620          0998  1 ! SIDE EFFECTS:
;  621          0999  1 !       none
;  622          1000  1 !
;  623          1001  1 !--
;  624          1002  1
;  625          1003  2     BEGIN
;  626          1004  2
;  627          1005  2     EXTERNAL REGISTER
;  628          1006  2         COMMON_REG;
;  629          1007  2
;  630          1008  2     MAP
;  631          1009  2         LABELS  : REF BBLOCK;                   ! HDR1, HDR2, and HDR3 address
;  632          1010  2
;  633          1011  2     BIND
;  634          1012  2
;  635          1013  2         ! Any file with 11 code will be supported, instead of only 11A
;  636          1014  2         !
;  637          1015  2         STARID = UPLIT ('DECFILE11');
;  638          1016  2
;  639          1017  2     EXTERNAL ROUTINE
;  640          1018  2         FORMAT_FID      : COMMON_CALL;          ! format file id
;  641          1019  2
;  642          1020  2     CURRENT_VCB[VCB$V_STARFILE] = CH$EQL(9, STARID, 9, LABELS[HD1$T_SYSCODE],0);
;  643          1021  2     FORMAT_FID(CURRENT_VCB[VCB$W_CUR_NUM]);
;  644          1022  1     END;                                        ! end of routine


                                                002F1                   .BLKB   3
                00  00  00  31  31  45  4C  49  46  43  45  44  002F4 P.AAD:  .ASCII  \DECFILE11\<0><0><0>                ;
```

```
                                                        STARID=          P.AAD
                                                                .EXTRN   FORMAT_FID

                                                001C 00000 MAKE_CUR_FILE:
                                                                .WORD    Save R2,R3,R4                          ; 0971
                                        50      04   AC  D0 00002       MOVL    LABELS, R0                     ; 1020
                                                     54  D4 00006       CLRL    R4                             ;
                        3C   A0         E8      AF       09 29 00008   CMPC3   #9, STARID, 60(R0)             ;
                                                          02 12 0000E   BNEQ    1$                             ;
                                                          54 D6 00010   INCL    R4                             ;
        2D   AB              01                 00       54 F0 00012 1$: INSV    R4, #0, #1, 45(CURRENT_VCB)    ;
                                                     24   AB 9F 00018   PUSHAB  36(CURRENT_VCB)                ; 1021
                                      0000G     CF       01 FB 0001B   CALLS   #1, FORMAT_FID                  ;
                                                          04 00020       RET                                    ; 1022

; Routine Size:  33 bytes,    Routine Base:  $CODE$ + 0300

;  645          1023  1
```

```
;  647          1024  1 GLOBAL ROUTINE UPDVCB_LEOV (BIT_VALUE) : COMMON_CALL NOVALUE =
;  648          1025  1
;  649          1026  1 !++
;  650          1027  1 !
;  651          1028  1 ! FUNCTIONAL DESCRIPTION:
;  652          1029  1 !       This routine sets or clears the logical end of file bit in the VCB
;  653          1030  1 !
;  654          1031  1 ! CALLING SEQUENCE:
;  655          1032  1 !       UPDVCB_LEOV(ARG1), called in kernel mode
;  656          1033  1 !
;  657          1034  1 ! INPUT PARAMETERS:
;  658          1035  1 !       value to set logical end of volume to:
;  659          1036  1 !               0 - clear bit
;  660          1037  1 !               1 - set bit
;  661          1038  1 !
;  662          1039  1 ! IMPLICIT INPUTS:
;  663          1040  1 !       CURRENT_VCB - address of volume control block
;  664          1041  1 !
;  665          1042  1 ! OUTPUT PARAMETERS:
;  666          1043  1 !       none
;  667          1044  1 !
;  668          1045  1 ! IMPLICIT OUTPUTS:
;  669          1046  1 !       CURRENT_VCB[VCB$V_LOGICEOVS] is set or cleared
;  670          1047  1 !
;  671          1048  1 ! ROUTINE VALUE:
;  672          1049  1 !       none
;  673          1050  1 !
;  674          1051  1 ! SIDE EFFECTS:
;  675          1052  1 !       none
;  676          1053  1 !
;  677          1054  1 !--
;  678          1055  1
;  679          1056  2     BEGIN
;  680          1057  2
;  681          1058  2     EXTERNAL REGISTER
;  682          1059  2         COMMON_REG;
;  683          1060  2
;  684          1061  2     CURRENT_VCB[VCB$V_LOGICEOVS] = .BIT_VALUE;
;  685          1062  1     END;                                        ! end of routine



                                                0000 00000          .ENTRY  UPDVCB_LEOV, Save nothing                   ; 1024
      0B   AB           01            01       04   AC  F0 00002    INSV    BIT_VALUE, #1, #1, 11(CURRENT_VCB)          ; 1061
                                                    04 00009        RET                                                 ; 1062

; Routine Size:  10 bytes,    Routine Base:  $CODE$ + 0321

;  686          1063  1
```

```
;  688          1064  1 ROUTINE SET_NUMBER_OF_LABELS (NUMBER_OF_LABELS) : COMMON_CALL NOVALUE =
;  689          1065  1
;  690          1066  1 !++
;  691          1067  1 !
;  692          1068  1 ! FUNCTIONAL DESCRIPTION:
;  693          1069  1 !       This routine sets then number of labels read by the MTAACP in the VCB.
;  694          1070  1 !       This value will be used to determine how many labels are written out
;  695          1071  1 !       won volume switch or at end of file processing. The reason this is
;  696          1072  1 !       necessary is so that if a file is op with fewer labels then we support
;  697          1073  1 !       we do not write the greater number o HDR labels out to the tape. This
;  698          1074  1 !       would be a noncompliance with the ANSI standard for tape label
;  699          1075  1 !       processing.
;  700          1076  1 !
;  701          1077  1 ! CALLING SEQUENCE:
;  702          1078  1 !       SET_NUMBER_OF_LABELS(ARG1), called in kernel mode
;  703          1079  1 !
;  704          1080  1 ! INPUT PARAMETERS:
;  705          1081  1 !       Number of labels read.
;  706          1082  1 !
;  707          1083  1 ! IMPLICIT INPUTS:
;  708          1084  1 !       CURRENT_VCB - address of volume control block
;  709          1085  1 !
;  710          1086  1 ! OUTPUT PARAMETERS:
;  711          1087  1 !       none
;  712          1088  1 !
;  713          1089  1 ! IMPLICIT OUTPUTS:
;  714          1090  1 !       CURRENT_VCB[VCB$B_LBLCNT] is set
;  715          1091  1 !
;  716          1092  1 ! ROUTINE VALUE:
;  717          1093  1 !       none
;  718          1094  1 !
;  719          1095  1 ! SIDE EFFECTS:
;  720          1096  1 !       none
;  721          1097  1 !
;  722          1098  1 !--
;  723          1099  1
;  724          1100  2     BEGIN
;  725          1101  2
;  726          1102  2     EXTERNAL REGISTER
;  727          1103  2         COMMON_REG;
;  728          1104  2
;  729          1105  2     CURRENT_VCB[VCB$B_LBLCNT] = .NUMBER_OF_LABELS;
;  730          1106  1     END;                                                    ! end of routine


                                        0000 00000 SET_NUMBER_OF_LABELS:
                                                   .WORD   Save nothing                                  ; 1064
                    48   AB       04   AC  90 00002       MOVB    NUMBER_OF_LABELS, 72(CURRENT_VCB)    ; 1105
                                             04 00007       RET                                          ; 1106

; Routine Size:  8 bytes,    Routine Base:  $CODE$ + 032B

;  731          1107  1
```

```
;  732          1108  1 END
;  733          1109  1
;  734          1110  0 ELUDOM



;                                PSECT SUMMARY
;
;      Name                    Bytes                          Attributes
;
;  $CODE$                        819  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        36         0      1000          00:01.8



;                                   COMMAND QUALIFIERS

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

; Size:         782 code + 37 data bytes
; Run Time:         00:17.7
; Elapsed Time:     00:40.6
; Lines/CPU Min:     3771
; Lexemes/CPU-Min: 18091
; Memory Used:  163 pages
; Compilation Complete
```

FREEPG
LIS

DEACCS
LIS

FIND
LIS

DATECN
LIS

FRMOD1
LIS

CREATE
LIS

GETREQ
LIS

EXPIRE
LIS

CNTRL
LIS

FRMHDR
LIS

HEADER
LIS

COMLABPRC
LIS

ENDVOL
LIS

GETFIB
LIS

LOCKDB LIS

LOGIO LIS

MATCHNAME LIS

IODONE LIS

LOCKDN LIS

NXTVOL LIS

MAIL LIS

MOUVOL LIS

OPRCOM LIS

INIMTA LIS

MODIFY LIS

PARSE LIS