
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

```
FFFFFFFFFF  RRRRRRRR    MM      MM    000000    DDDDDDDD       11
FFFFFFFFFF  RRRRRRRR    MM      MM    000000    DDDDDDDD       11
FF          RR      RR  MMMM  MMMM  00      00  DD      DD   1111
FF          RR      RR  MMMM  MMMM  00      00  DD      DD   1111
FF          RR      RR  MM  MM  MM  00      00  DD      DD     11
FF          RR      RR  MM  MM  MM  00      00  DD      DD     11
FFFFFFFF    RRRRRRRR    MM      MM  00      00  DD      DD     11
FFFFFFFF    RRRRRRRR    MM      MM  00      00  DD      DD     11
FF          RR  RR      MM      MM  00      00  DD      DD     11
FF          RR  RR      MM      MM  00      00  DD      DD     11
FF          RR    RR    MM      MM  00      00  DD      DD     11          ....
FF          RR    RR    MM      MM  00      00  DD      DD     11          ....
FF          RR      RR  MM      MM    000000    DDDDDDDD     111111        ....
FF          RR      RR  MM      MM    000000    DDDDDDDD     111111        ....

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 FRMOD1 (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 formats a files-11 structure level 1
;    37          0037  1 !       file header
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT:
;    40          0040  1 !
;    41          0041  1 !       Starlet operating system, including privileged system services
;    42          0042  1 !       and internal exec routines.
;    43          0043  1 !
;    44          0044  1 !--
;    45          0045  1
;    46          0046  1 !
;    47          0047  1 !
;    48          0048  1 ! AUTHOR:  D. H. GILLESPIE,      CREATION DATE:  19-MAY-1977  17:06
;    49          0049  1 !
;    50          0050  1 ! MODIFIED BY:
;    51          0051  1 !
;    52          0052  1 !       V03-002 LMP0221         L. Mark Pilant,         28-Mar-1984  13:25
;    53          0053  1 !               Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
;    54          0054  1 !               ORB$W_PROT.
;    55          0055  1 !
;    56          0056  1 !       V03-001 MMD0155         Meg Dumont,     26-Apr-1983  9:15
;    57          0057  1 !               Cleanup of comments
```

```
;    58          0058  1 !
;    59          0059  1 !       V02-010 DMW00057        David Michael Walp       7-Dec-1981
;    60          0060  1 !               Changed CALC_VERSION to non-COMMON_CALL
;    61          0061  1 !
;    62          0062  1 !       V02-009 DMW00056        David Michael Walp      30-Nov-1981
;    63          0063  1 !               Added file name parsing routines.  Part of ANSI "a"
;    64          0064  1 !               file name support
;    65          0065  1 !
;    66          0066  1 !       V02-008 DMW00037        David Michael Walp      29-Sep-1981
;    67          0067  1 !               Format the ODS1 header correctly
;    68          0068  1 !
;    69          0069  1 !       V02-007 REFORMAT        Maria del C. Nasr       30-Jun-1980
;    70          0070  1 !
;    71          0071  1 !       A0006   MCN0003         Maria del C. Nasr       16-Oct-1979   14:06
;    72          0072  1 !               Add HDR3 processing
;    73          0073  1 !
;    74          0074  1 !       A0005   MCN0004         Maria del C. Nasr       15-Oct-1979   15:25
;    75          0075  1 !               Changed to use new file header structure name
;    76          0076  1 !
;    77          0077  1 !**
;    78          0078  1
;    79          0079  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    80          0080  1
;    81          0081  1 REQUIRE 'SRC$:MTADEF.B32';
;    82          0465  1
;    83          0466  1 LINKAGE
;    84          0467  1     JSB_CHAR    = JSB () : GLOBAL ( CHAR_COUNT = 1, STRING_PTR = 2);
;    85          0468  1
;    86          0469  1 EXTERNAL ROUTINE
;    87          0470  1     LIB$CVT_DTB : ADDRESSING_MODE (ABSOLUTE);   ! decimal to binary
;    88          0471  1
;    89          0472  1 FORWARD ROUTINE
;    90          0473  1     FORMAT_DATEOD1      : NOVALUE,              ! formats date for ODS1
;    91          0474  1     FORMAT_F11HOD1      : COMMON_CALL NOVALUE,  ! format Files-11 ODS1 Header
;    92          0475  1     FORMAT_FID          : NOVALUE,              ! format File Id
;    93          0476  1     GETCHAR             : JSB_CHAR;             ! get char from input string
;    94          0477  1
;    95          0478  1 EXTERNAL
;    96          0479  1     HDR1                : REF BBLOCK,           ! address of HDR1(EOF1) label
;    97          0480  1     HDR4                : REF BBLOCK;           ! address of HDR4(EOF4) label
```

```
;    99        M 0481  1 MACRO PARSE_TO_RAD50_FILE_ID =
;   100        M 0482  1
;   101        M 0483  1 !++
;   102        M 0484  1 !
;   103        M 0485  1 ! FUNCTIONAL DESCRIPTION:
;   104        M 0486  1 !       This routine parses the file identifier field in HDR1, trailing blanks
;   105        M 0487  1 !       are stripped from type and name fields, translate to RAD50 and
;   106        M 0488  1 !       calculate version number. If translating to RAD50 no need to interpret
;   107        M 0489  1 !       HDR4 information, beacuse the long file names are not supported in
;   108        M 0490  1 !       this mode.
;   109        M 0491  1 !
;   110        M 0492  1 ! CALLING SEQUENCE:
;   111        M 0493  1 !       PARSE_TO_RAD50_FILE_ID
;   112        M 0494  1 !
;   113        M 0495  1 ! INPUT PARAMETERS:
;   114        M 0496  1 !       none
;   115        M 0497  1 !
;   116        M 0498  1 ! IMPLICIT INPUTS:
;   117        M 0499  1 !       none
;   118        M 0500  1 !
;   119        M 0501  1 ! OUTPUT PARAMETERS:
;   120        M 0502  1 !       none
;   121        M 0503  1 !
;   122        M 0504  1 ! IMPLICIT OUTPUTS:
;   123        M 0505  1 !       none
;   124        M 0506  1 !
;   125        M 0507  1 ! ROUTINE VALUE:
;   126        M 0508  1 !       none
;   127        M 0509  1 !
;   128        M 0510  1 !--
;   129        M 0511  1
;   130        M 0512  1     BEGIN
;   131        M 0513  1
;   132        M 0514  1     ! address of the name block in the FILE header
;   133        M 0515  1     !
;   134        M 0516  1     BIND
;   135        M 0517  1         NAM_BLK = HEADER[ FH1$C_LENGTH + FI1$W_FILENAME ] : VECTOR [ , WORD ];
;   136        M 0518  1
;   137        M 0519  1     EXTERNAL ROUTINE
;   138        M 0520  1         CALC_VERSION;                               ! get the version number
;   139        M 0521  1
;   140        M 0522  1     LOCAL
;   141        M 0523  1         TYPE_PTR: REF VECTOR [ , BYTE ];            ! addr of file type field
;   142        M 0524  1
;   143        M 0525  1     GLOBAL REGISTER
;   144        M 0526  1         CHAR_COUNT = 1  : LONG,                     ! number of characters left
;   145        M 0527  1         STRING_PTR = 2  : REF VECTOR [ , BYTE ];! string pointer
;   146        M 0528  1
;   147        M 0529  1     LITERAL
;   148        M 0530  1         MAX_NAME_LEN   = 9,                         ! maximum file name length
;   149        M 0531  1         MAX_TYPE_LEN   = 3,                         ! maximum file type length
;   150        M 0532  1         NAM_TYPE_FIELD = 3,                         ! file type offset in name BLK
;   151        M 0533  1         NAM_VER_FIELD  = 4;                         ! file ver offset in name BLK
;   152        M 0534  1
;   153        M 0535  1
;   154        M 0536  1     !++
;   155        M 0537  1     !
```

```
;  156          M 0538  1      !  PARSE and TRANSLATE to RAD50 THE FILE NAME FILE FIELD
;  157          M 0539  1      !
;  158          M 0540  1      !--
;  159          M 0541  1
;  160          M 0542  1      ! setup address of the file id in HDR1
;  161          M 0543  1      !
;  162          M 0544  1      STRING_PTR  = HDR1 [ HD1$T_FILEID ];
;  163          M 0545  1
;  164          M 0546  1      ! scan up to 9 characters or until a period is found
;  165          M 0547  1      !
;  166          M 0548  1      CHAR_COUNT = CH$FIND_CH ( MAX_NAME_LEN, .STRING_PTR, '.' );
;  167          M 0549  1      CHAR_COUNT = ( IF CH$FAIL ( .CHAR_COUNT )
;  168          M 0550  1                     THEN MAX_NAME_LEN
;  169          M 0551  1                     ELSE .CHAR_COUNT - .STRING_PTR );
;  170          M 0552  1
;  171          M 0553  1      ! save the pointer to the start of the file type field
;  172          M 0554  1      !
;  173          M 0555  1      TYPE_PTR = STRING_PTR [ .CHAR_COUNT ];
;  174          M 0556  1
;  175          M 0557  1      ! strip off trailing blanks
;  176          M 0558  1      !
;  177          M 0559  1      DECR I FROM (.CHAR_COUNT - 1) TO 0 DO
;  178          M 0560  1          IF .STRING_PTR[.I] NEQ ' '
;  179          M 0561  1          THEN EXITLOOP
;  180          M 0562  1          ELSE CHAR_COUNT = .CHAR_COUNT - 1;
;  181          M 0563  1
;  182          M 0564  1      ! convert file name to rad50
;  183          M 0565  1      !
;  184          M 0566  1      INCRU I FROM 0 TO 2 DO
;  185          M 0567  1          DECRU J FROM 3 TO 1 DO NAM_BLK[.I] = .NAM_BLK[.I]*40 + GETCHAR();
;  186          M 0568  1
;  187          M 0569  1
;  188          M 0570  1      !++
;  189          M 0571  1      !
;  190          M 0572  1      !  PARSE and TRANSLATE to RAD50 THE FILE NAME TYPE FIELD
;  191          M 0573  1      !
;  192          M 0574  1      !--
;  193          M 0575  1
;  194          M 0576  1      ! pickup the saved address of file type string
;  195          M 0577  1      !
;  196          M 0578  1      STRING_PTR = .TYPE_PTR;
;  197          M 0579  1
;  198          M 0580  1      ! if we are at a "." use it as a delimitor
;  199          M 0581  1      !
;  200          M 0582  1      IF .STRING_PTR[0] EQLU '.' THEN STRING_PTR = .STRING_PTR + 1;
;  201          M 0583  1
;  202          M 0584  1      ! strip trailing spaces
;  203          M 0585  1      !
;  204          M 0586  1      CHAR_COUNT = MAX_TYPE_LEN;
;  205          M 0587  1      DECR I FROM 2 TO 0 DO
;  206          M 0588  1          IF .STRING_PTR [.I] NEQ ' '
;  207          M 0589  1          THEN EXITLOOP
;  208          M 0590  1          ELSE CHAR_COUNT = .CHAR_COUNT - 1;
;  209          M 0591  1
;  210          M 0592  1      ! convert the file type to RAD50
;  211          M 0593  1      !
;  212          M 0594  1      DECRU I FROM 3 TO 1 DO
```

```
;  213          M 0595  1          NAM_BLK[NAM_TYPE_FIELD] = .NAM_BLK[NAM_TYPE_FIELD]*40 + GETCHAR();
;  214          M 0596  1
;  215          M 0597  1
;  216          M 0598  1      !++
;  217          M 0599  1      !
;  218          M 0600  1      ! convert generation number and generation version number to binary
;  219          M 0601  1      ! file version number
;  220          M 0602  1      !
;  221          M 0603  1      !--
;  222          M 0604  1
;  223          M 0605  1      CALC_VERSION ( NAM_BLK [ NAM_VER_FIELD ] )
;  224          M 0606  1
;  225          M 0607  1      END;
;  226            0608  1      %;
```

```
;  228     0609  1 ROUTINE GETCHAR : JSB_CHAR =
;  229     0610  1
;  230     0611  1 !++
;  231     0612  1 !
;  232     0613  1 ! FUNCTIONAL DESCRIPTION:
;  233     0614  1 !
;  234     0615  1 !       This routine returns the rad-50 code of the next character in the
;  235     0616  1 !       input string if it is in the rad-50 set. If end of string has been
;  236     0617  1 !       reached, it returns zero.  If it is a non-rad50 char, then it is
;  237     0618  1 !       mapped to z.  It uppercase letters.
;  238     0619  1 !
;  239     0620  1 ! CALLING SEQUENCE:
;  240     0621  1 !       GETCHAR ()
;  241     0622  1 !
;  242     0623  1 ! INPUT PARAMETERS:
;  243     0624  1 !       none
;  244     0625  1 !
;  245     0626  1 ! IMPLICIT INPUTS:
;  246     0627  1 !       STRING_PTR - string pointer
;  247     0628  1 !       CHAR_COUNT - count of the number of characters
;  248     0629  1 !
;  249     0630  1 ! OUTPUT PARAMETERS:
;  250     0631  1 !       none
;  251     0632  1 !
;  252     0633  1 ! IMPLICIT OUTPUTS:
;  253     0634  1 !       STRING_PTR - string pointer
;  254     0635  1 !       CHAR_COUNT - count of the number of characters
;  255     0636  1 !
;  256     0637  1 ! ROUTINE VALUE:
;  257     0638  1 !       character code
;  258     0639  1 !
;  259     0640  1 ! SIDE EFFECTS:
;  260     0641  1 !       Count decremented and stringp advanced.
;  261     0642  1 !
;  262     0643  1 !--
;  263     0644  1
;  264     0645  2     BEGIN
;  265     0646  2
;  266     0647  2     EXTERNAL REGISTER
;  267     0648  2         CHAR_COUNT = 1  : LONG,                     ! number of characters left
;  268     0649  2         STRING_PTR = 2  : REF VECTOR [, BYTE];      ! pointer into the string
;  269     0650  2
;  270     0651  2     LITERAL
;  271     0652  2         RAD50_Z = ('Z' - 'A') + 1;
;  272     0653  2
;  273     0654  2     LOCAL
;  274     0655  2         CHAR;                                       ! character in process
;  275     0656  2
;  276     0657  2     ! If the string is empty return 0 as the character
;  277     0658  2     !
;  278     0659  2     IF .CHAR_COUNT LEQ 0 THEN RETURN 0;
;  279     0660  2
;  280     0661  2     ! Get the next character from the string
;  281     0662  2     !
;  282     0663  2     CHAR = .STRING_PTR[0];
;  283     0664  2
;  284     0665  2     ! advance to next character
```

```
;  285          0666  2     !
;  286          0667  2     CHAR_COUNT = .CHAR_COUNT - 1;
;  287          0668  2     STRING_PTR = .STRING_PTR + 1;
;  288          0669  2
;  289          0670  2     RETURN SELECTONE .CHAR OF
;  290          0671  2             SET
;  291          0672  2                 ['A' TO 'Z'] : (.CHAR - 'A') + 1;
;  292          0673  2                 ['a' TO 'z'] : (.CHAR - 'a') + 1;
;  293          0674  2                 ['0' TO '9'] : (.CHAR - '0') + 30;
;  294          0675  2                 [OTHERWISE]  : RAD50_Z;
;  295          0676  2             TES;
;  296          0677  1     END;                                    ! end of routine getchar


                                                .TITLE  FRMOD1
                                                .IDENT  \V04-000\

                                                .EXTRN  LIB$CVT_DTB, HDR1
                                                .EXTRN  HDR4

                                                .PSECT  $CODE$,NOWRT,2

                                    53  DD 00000 GETCHAR:PUSHL   R3                              ; 0609
                                    51  D5 00002         TSTL    CHAR_COUNT                      ; 0659
                                    4C  15 00004         BLEQ    5$                              ;
                          50        82  9A 00006         MOVZBL  (STRING_PTR)+, CHAR             ; 0663
                                    51  D7 00009         DECL    CHAR_COUNT                      ; 0667
                00000041  8F        50  D1 0000B         CMPL    CHAR, #65                       ; 0672
                                    0F  19 00012         BLSS    1$                              ;
                0000005A  8F        50  D1 00014         CMPL    CHAR, #90                       ;
                                    06  14 0001B         BGTR    1$                              ;
                          53    C0  A0  9E 0001D         MOVAB   -64(R0), R3                     ;
                                    16  11 00021         BRB     2$                              ;
                00000061  8F        50  D1 00023 1$:     CMPL    CHAR, #97                       ; 0673
                                    12  19 0002A         BLSS    3$                              ;
                0000007A  8F        50  D1 0002C         CMPL    CHAR, #122                      ;
                                    09  14 00033         BGTR    3$                              ;
                          53    A0  A0  9E 00035         MOVAB   -96(R0), R3                     ;
                          50        53  D0 00039 2$:     MOVL    R3, R0                          ;
                                    16  11 0003C         BRB     6$                              ;
                          30        50  D1 0003E 3$:     CMPL    CHAR, #48                       ; 0674
                                    0A  19 00041         BLSS    4$                              ;
                          39        50  D1 00043         CMPL    CHAR, #57                       ;
                                    05  14 00046         BGTR    4$                              ;
                          50        12  C2 00048         SUBL2   #18, R0                         ;
                                    07  11 0004B         BRB     6$                              ;
                          50        1A  D0 0004D 4$:     MOVL    #26, R0                         ; 0675
                                    02  11 00050         BRB     6$                              ; 0670
                                    50  D4 00052 5$:     CLRL    R0                              ; 0677
                                    08  BA 00054 6$:     POPR    #^M<R3>                         ;
                                        05 00056         RSB                                     ;

; Routine Size:  87 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  298          0678  1 GLOBAL ROUTINE FORMAT_F11HOD1 (HEADER) : COMMON_CALL NOVALUE =
;  299          0679  1
;  300          0680  1 !++
;  301          0681  1 !
;  302          0682  1 ! FUNCTIONAL DESCRIPTION:
;  303          0683  1 !       This routine formats an Ansi labeled magnetic tape file header
;  304          0684  1 !       for files-11 on-disk structure 1
;  305          0685  1 !
;  306          0686  1 ! CALLING SEQUENCE:
;  307          0687  1 !       FORMAT_F11HOD1(ARG1)
;  308          0688  1 !
;  309          0689  1 ! INPUT PARAMETERS:
;  310          0690  1 !       ARG1 - address of buffer
;  311          0691  1 !
;  312          0692  1 ! IMPLICIT INPUTS:
;  313          0693  1 !       CURRENT_VCB - address of current vcb
;  314          0694  1 !       tape header labels read in or defaulted
;  315          0695  1 !
;  316          0696  1 ! OUTPUT PARAMETERS:
;  317          0697  1 !       ARG1 - address of buffer
;  318          0698  1 !
;  319          0699  1 ! IMPLICIT OUTPUTS:
;  320          0700  1 !       none
;  321          0701  1 !
;  322          0702  1 ! ROUTINE VALUE:
;  323          0703  1 !       none
;  324          0704  1 !
;  325          0705  1 ! SIDE EFFECTS:
;  326          0706  1 !       header receives 512 characters of files-11 ODS-1 file header
;  327          0707  1 !
;  328          0708  1 !--
;  329          0709  1
;  330          0710  2     BEGIN
;  331          0711  2
;  332          0712  2     EXTERNAL REGISTER
;  333          0713  2         COMMON_REG;
;  334          0714  2
;  335          0715  2     EXTERNAL ROUTINE
;  336          0716  2         LIB$CVT_HTB     : ADDRESSING_MODE (ABSOLUTE);   ! hexadecimal to binary
;  337          0717  2
;  338          0718  2     MAP HEADER          : REF BBLOCK;           ! addr of output buffer
;  339          0719  2
;  340          0720  2     EXTERNAL
;  341          0721  2         LOCAL_FIB       : BBLOCK,
;  342          0722  2         CURRENT_UCB     : REF BBLOCK,           ! addr of current UCB
;  343          0723  2         HDR2            : REF BBLOCK,           ! addr of HDR2(EOF2) label
;  344          0724  2         HDR3            : REF BBLOCK;           ! addr of HDR3(EOF3) label
;  345          0725  2
;  346          0726  2     LOCAL
;  347          0727  2         ORB             : REF BBLOCK;           ! Address of ORB
;  348          0728  2
;  349          0729  2     BIND
;  350          0730  2         RECTYPE_ANSI    = UPLIT BYTE('UFDVS')           : VECTOR [, BYTE],
;  351          0731  2         RECTYPE_FILE11  = UPLIT BYTE(0, 1, 2, 3, 0)     : VECTOR [, BYTE],
;  352          0732  2         RECATTR_ASCI    = UPLIT BYTE('MA ')             : VECTOR [, BYTE],
;  353          0733  2         RECATTR_FILE11  = UPLIT BYTE(0, 1, 2)           : VECTOR [, BYTE],
;  354          0734  2
```

```
;  355     0735  2          ! pointer into the record attributes in the header part
;  356     0736  2          !
;  357     0737  2          RECATTR         = HEADER[FH1$W_RECATTR]      : BBLOCK;
;  358     0738  2
;  359     0739  2      LITERAL
;  360     0740  2          RECTYPE_LEN = 5,
;  361     0741  2          RECATTR_LEN = 3;
;  362     0742  2
;  363     0743  2
;  364     0744  2      ! zero in the header
;  365     0745  2      !
;  366     0746  2      CH$FILL(0, 512, .HEADER);
;  367     0747  2
;  368     0748  2      ! fill in offset to ID and MAP areas
;  369     0749  2      !
;  370     0750  2      HEADER[FH1$B_IDOFFSET] = FH1$C_LENGTH/2;
;  371     0751  2      HEADER[FH1$B_MPOFFSET] = (FH1$C_LENGTH + FI1$C_LENGTH +
;  372     0752  2                                FI1$S_MTHDR1 + FI1$S_MTHDR2 + FI1$S_MTHDR3)/2;
;  373     0753  2
;  374     0754  2
;  375     0755  2      !++
;  376     0756  2      !
;  377     0757  2      ! FILL IN THE FILE HEADER AREA OF THE HEADER
;  378     0758  2      !
;  379     0759  2      !--
;  380     0760  2
;  381     0761  2      ! structure level  ( known constant )
;  382     0762  2      !
;  383     0763  2      HEADER[FH1$W_STRUCLEV] = FH1$C_LEVEL1;
;  384     0764  2
;  385     0765  2      ! fill in the File ID (FID)
;  386     0766  2      !
;  387     0767  2      FORMAT_FID(HEADER[FH1$W_FID_NUM]);
;  388     0768  2
;  389     0769  2      ! Get the address of the Object's Rights Block (ORB).
;  390     0770  2
;  391     0771  2      ORB = .CURRENT_UCB[UCB$L_ORB];
;  392     0772  2
;  393     0773  2      ! format fileowner from UCB, if the a long UIC, fold into [377,377]
;  394     0774  2      !
;  395     0775  2      IF .(ORB[ORB$W_UICGROUP])<8,8> NEQ 0
;  396     0776  2      OR .(ORB[ORB$W_UICMEMBER])<8,8> NEQ 0
;  397     0777  2      THEN HEADER[FH1$W_FILEOWNER] = -1
;  398     0778  2      ELSE
;  399     0779  3          BEGIN
;  400     0780  3          HEADER[FH1$B_UICMEMBER] = .ORB[ORB$W_UICMEMBER];
;  401     0781  3          HEADER[FH1$B_UICGROUP] = .ORB[ORB$W_UICGROUP];
;  402     0782  2          END;
;  403     0783  2
;  404     0784  2      ! protection from UCB
;  405     0785  2      !
;  406     0786  2      IF .ORB[ORB$V_PROT_16]
;  407     0787  2      THEN  HEADER[FH1$W_FILEPROT] = .ORB[ORB$W_PROT]
;  408     0788  2      ELSE
;  409     0789  3          BEGIN
;  410     0790  3          (HEADER[FH1$W_FILEPROT])<0,4> = .(ORB[ORB$L_SYS_PROT])<0,4>;
;  411     0791  3          (HEADER[FH1$W_FILEPROT])<4,4> = .(ORB[ORB$L_OWN_PROT])<0,4>;
```

```
;  412          0792  3          (HEADER[FH1$W_FILEPROT])<8,4> = .(ORB[ORB$L_GRP_PROT])<0,4>;
;  413          0793  3          (HEADER[FH1$W_FILEPROT])<12,4> = .(ORB[ORB$L_WOR_PROT])<0,4>;
;  414          0794  2          END;
;  415          0795  2
;  416          0796  2      !++
;  417          0797  2      !
;  418          0798  2      !  Fill in the Record Attributes in the Header Id Area
;  419          0799  2      !
;  420          0800  2      !  1st  Fill in the stuff from the ANSI field
;  421          0801  2      !  2nd  Get the RMS stuff in HDR2 or HDR3
;  422          0802  2      !
;  423          0803  2      !--
;  424          0804  2
;  425          0805  2      ! fill in the type of record ( variable, fixed, undefined, spaned )
;  426          0806  2      ! assume undefined
;  427          0807  2      !
;  428          0808  2      RECATTR[FAT$B_RTYPE] = 0;
;  429          0809  2      DECR I FROM RECTYPE_LEN - 1 TO 0 DO
;  430          0810  2          IF .RECTYPE_ANSI[.I] EQL .HDR2[HD2$B_RECFORMAT]
;  431          0811  2          THEN
;  432          0812  3              BEGIN
;  433          0813  3              RECATTR[FAT$B_RTYPE] = .RECTYPE_FILE11[.I];
;  434          0814  3              EXITLOOP;
;  435          0815  2              END;
;  436          0816  2
;  437          0817  2      ! fill in form control ( Fortran, Form control in record, implied <CRLF> )
;  438          0818  2      ! assume implied <CRLF>
;  439          0819  2      !
;  440          0820  2      RECATTR[FAT$B_RATTRIB] = 2;
;  441          0821  2      DECR I FROM RECATTR_LEN - 1 TO 0 DO
;  442          0822  2          IF .RECATTR_ASCI[.I] EQL .HDR2[HD2$B_FORMCNTRL]
;  443          0823  2          THEN
;  444          0824  3              BEGIN
;  445          0825  3              RECATTR[FAT$B_RATTRIB] = .RECATTR_FILE11[.I];
;  446          0826  3              EXITLOOP;
;  447          0827  2              END;
;  448          0828  2
;  449          0829  2      ! fill in the record size and max size
;  450          0830  2      !
;  451          0831  3      BEGIN
;  452          0832  3      LOCAL VALUE;
;  453          0833  3
;  454          0834  3      LIB$CVT_DTB(HD2$S_RECLEN, HDR2[HD2$T_RECLEN], VALUE);
;  455          0835  3
;  456          0836  3      ! if variable length records subtract overhead
;  457          0837  3      !
;  458          0838  3      IF .HDR2[HD2$B_RECFORMAT] EQL 'D' THEN VALUE = .VALUE - 4;
;  459          0839  3
;  460          0840  3      RECATTR[FAT$W_RSIZE]  = .VALUE<0, 16>;
;  461          0841  3      RECATTR[FAT$W_MAXREC] = .VALUE<0, 16>
;  462          0842  2      END;
;  463          0843  2
;  464          0844  2      ! if the file was created on a VMS system fill in RMS attributes
;  465          0845  2      !
;  466          0846  2      IF .CURRENT_VCB[VCB$V_STARFILE]
;  467          0847  2      THEN
;  468          0848  2          IF .(HDR2[HD2$T_RECATR1])<0,8> NEQ ' '
```

```
;  469     0849  2
;  470     0850  2          ! old tape, attributes are in HDR2
;  471     0851  2          !
;  472     0852  2          THEN
;  473     0853  3              BEGIN
;  474     0854  3              CH$MOVE(HD2$S_RECATR1, HDR2[HD2$T_RECATR1], RECATTR);
;  475     0855  3              CH$MOVE(HD2$S_RECATR2, HDR2[HD2$T_RECATR2], RECATTR+HD2$S_RECATR1);
;  476     0856  3              END
;  477     0857  3
;  478     0858  3          ! get attributes from HDR3, converting to binary
;  479     0859  3          !
;  480     0860  2          ELSE
;  481     0861  2              IF .HDR3[HD3$L_HD3LID] EQL 'HDR3'
;  482     0862  2                  THEN INCR I FROM 0 TO 28 BY 4 DO
;  483     0863  2                      LIB$CVT_HTB(8, HDR3[HD3$T_RECATR] + (.I*2), RECATTR + .I);
;  484     0864  2
;  485     0865  2
;  486     0866  2      ! records do not span block boundaries if the user requests the original
;  487     0867  2      ! attributes of the file, don't set span
;  488     0868  2      !
;  489     0869  2      IF NOT .LOCAL_FIB[FIB$V_PRSRV_ATR]
;  490     0870  2      THEN RECATTR[FAT$B_RATTRIB] = .RECATTR[FAT$B_RATTRIB] OR FAT$M_NOSPAN;
;  491     0871  2
;  492     0872  2
;  493     0873  2      !++
;  494     0874  2      !
;  495     0875  2      ! FILL IN THE FILE ID AREA OF THE HEADER
;  496     0876  2      !
;  497     0877  2      !--
;  498     0878  2
;  499     0879  2      ! fill in the file name, type and version
;  500     0880  2      !
;  501     0881  2      PARSE_TO_RAD50_FILE_ID;
;  502     0882  2
;  503     0883  2      ! fill the creation and expiration dates
;  504     0884  2      !
;  505     0885  2      FORMAT_DATEOD1(HDR1[HD1$T_CREATEDT],HEADER[FH1$C_LENGTH+FI1$T_CREDATE],1);
;  506     0886  2      FORMAT_DATEOD1(HDR1[HD1$T_EXPIREDT],HEADER[FH1$C_LENGTH+FI1$T_EXPDATE],0);
;  507     0887  2
;  508     0888  2      ! move the tape file header labels into the ident area of the ODS1 header
;  509     0889  2      !
;  510     0890  2      CH$MOVE( FI1$S_MTHDR1 + FI1$S_MTHDR2 + FI1$S_MTHDR3,
;  511     0891  2               .HDR1, HEADER[FH1$C_LENGTH + FI1$T_MTHDR1]);
;  512     0892  2
;  513     0893  2
;  514     0894  2      !++
;  515     0895  2      !
;  516     0896  2      ! CREATE A DUMMY MAP AREA FOR THE ODS1 HEADER
;  517     0897  2      !
;  518     0898  2      !--
;  519     0899  3      BEGIN
;  520     0900  3
;  521     0901  3      LOCAL MAP_AREA : REF BBLOCK;
;  522     0902  3
;  523     0903  3      MAP_AREA = .HEADER + (.HEADER [ FH1$B_MPOFFSET ] * 2 );
;  524     0904  3      MAP_AREA[FM1$B_COUNTSIZE] = 1;
;  525     0905  3      MAP_AREA[FM1$B_LBNSIZE]   = 3;
```

```
;  526          0906  3     MAP_AREA[FM1$B_AVAIL]      =
;  527          0907  3         (HEADER[FHT$W_CHECKSUM] - .MAP_AREA - FM1$C_LENGTH ) / 2;
;  528          0908  3
;  529          0909  2     END;
;  530          0910  2
;  531          0911  2
;  532          0912  2     !++
;  533          0913  2     !
;  534          0914  2     ! CALCULATE THE CHECKSUM
;  535          0915  2     !
;  536          0916  2     !--
;  537          0917  3     BEGIN
;  538          0918  3     BIND
;  539          0919  3         ! reference the dummy ODS1 header as word
;  540          0920  3         !
;  541          0921  3         VHEADER = HEADER : REF VECTOR [, WORD];
;  542          0922  3
;  543          0923  3     ! now calculate checksum
;  544          0924  3     !
;  545          0925  3     DECR I FROM 254 TO 0 DO
;  546          0926  3         HEADER[FH1$W_CHECKSUM] = .HEADER[FH1$W_CHECKSUM] + .VHEADER[.I]
;  547          0927  2     END;
;  548          0928  2
;  549          0929  1     END;                                    ! end of routine


                                                  53  56  44  46  55  00057 P.AAA:  .ASCII  \UFDVS\                          ;
                                                  00  03  02  01  00  0005C P.AAB:  .BYTE   0, 1, 2, 3, 0                     ;
                                                          20  41  41  00061 P.AAC:  .ASCII  \MA \                             ;
                                                          02  01  00  00064 P.AAD:  .BYTE   0, 1, 2                           ;

                                                                              RECTYPE_ANSI=       P.AAA
                                                                              RECTYPE_FILE11=     P.AAB
                                                                              RECATTR_ASCI=       P.AAC
                                                                              RECATTR_FILE11=     P.AAD
                                                                                    .EXTRN  LIB$CVT_HTB, LOCAL_FIB
                                                                                    .EXTRN  CURRENT_UCB, HDR2
                                                                                    .EXTRN  HDR3, CALC_VERSION

                                                                  07FC 00000        .ENTRY  FORMAT_F11HOD1, Save R2,R3,R4,R5,R6,R7,R8,- ; 0678
                                                                                            R9,R10
                                                      5A          94   AF  9E 00002        MOVAB   GETCHAR, R10                        ;
                                                      5E               04  C2 00006        SUBL2   #4, SP                              ;
                                                      58          04   AC  D0 00009        MOVL    HEADER, R8                          ; 0737
                                                      56          0E   A8  9E 0000D        MOVAB   14(R8), R6                          ;
0200  8F                    00                        6E               00  2C 00011        MOVC5   #0, (SP), #0, #512, (R8)            ; 0746
                                                                         68     00018
                                                      68        A617   8F  B0 00019        MOVW    #42519, (R8)                        ; 0750
                                                06    A8        0101   8F  B0 0001E        MOVW    #257, 6(R8)                         ; 0763
                                                            02          A8  9F 00024        PUSHAB  2(R8)                               ; 0767
                                              0000V CF                  01  FB 00027        CALLS   #1, FORMAT_FID                      ;
                                                      50  0000G   CF   D0 0002C        MOVL    CURRENT_UCB, R0                     ; 0771
                                                      50    1C    A0   D0 00031        MOVL    28(R0), ORB                         ;
                                                            03    A0   95 00035        TSTB    3(ORB)                              ; 0775
                                                                  05   12 00038        BNEQ    1$                                  ;
                                                            01    A0   95 0003A        TSTB    1(ORB)                              ; 0776
```

```
                                          06   13 0003D        BEQL    2$                              ;
                    08   A8               01   AE 0003F  1$:   MNEGW   #1, 8(R8)                       ; 0777
                                          09   11 00043        BRB     3$                              ;
                    08   A8               60   90 00045  2$:   MOVB    (ORB), 8(R8)                    ; 0780
                    09   A8        02     A0   90 00049        MOVB    2(ORB), 9(R8)                   ; 0781
                         51        0A     A8   9E 0004E  3$:   MOVAB   10(R8), R1                      ; 0787
                         06        0B     A0   E9 00052        BLBC    11(ORB), 4$                     ; 0786
                         61        18     A0   B0 00056        MOVW    24(ORB), (R1)                   ; 0787
                                          19   11 0005A        BRB     5$                              ;
     61        04        00        18     A0   F0 0005C  4$:   INSV    24(ORB), #0, #4, (R1)           ; 0790
     61        04        04        1C     A0   F0 00062        INSV    28(ORB), #4, #4, (R1)           ; 0791
 01  A1        04        00        20     A0   F0 00068        INSV    32(ORB), #0, #4, 1(R1)          ; 0792
     61        04        0C        24     A0   F0 0006F        INSV    36(ORB), #12, #4, (R1)          ; 0793
                                          66   94 00075  5$:   CLRB    (R6)                            ; 0808
                         51     0000G     CF   D0 00077        MOVL    HDR2, R1                        ; 0810
                         50               04   D0 0007C        MOVL    #4, I                           ;
                    04   A1     FF6C    CF40   91 0007F  6$:   CMPB    RECTYPE_ANSI[I], 4(R1)          ;
                                          08   12 00086        BNEQ    7$                              ;
                         66     FF68    CF40   90 00088        MOVB    RECTYPE_FILE11[I], (R6)         ; 0813
                                          03   11 0008E        BRB     8$                              ; 0812
                         EC               50   F4 00090  7$:   SOBGEQ  I, 6$                           ; 0810
                    01   A6               02   90 00093  8$:   MOVB    #2, 1(R6)                       ; 0820
                         50               02   D0 00097        MOVL    #2, I                           ; 0822
                    24   A1     FF5B    CF40   91 0009A  9$:   CMPB    RECATTR_ASCI[I], 36(R1)         ;
                                          09   12 000A1        BNEQ    10$                             ;
                    01   A6     FF55    CF40   90 000A3        MOVB    RECATTR_FILE11[I], 1(R6)        ; 0825
                                          03   11 000AA        BRB     11$                             ; 0824
                         EB               50   F4 000AC  10$:  SOBGEQ  I, 9$                           ; 0822
                                          5E   DD 000AF  11$:  PUSHL   SP                              ; 0834
                                   0A     A1   9F 000B1        PUSHAB  10(R1)                          ;
                                          05   DD 000B4        PUSHL   #5                              ;
               00000000G  9F              03   FB 000B6        CALLS   #3, @#LIB$CVT_DTB               ;
                         57     0000G     CF   D0 000BD        MOVL    HDR2, R7                        ; 0838
                    44   8F        04     A7   91 000C2        CMPB    4(R7), #68                      ;
                                          03   12 000C7        BNEQ    12$                             ;
                         6E               04   C2 000C9        SUBL2   #4, VALUE                       ;
                    02   A6               6E   B0 000CC  12$:  MOVW    VALUE, 2(R6)                    ; 0840
                    10   A6               6E   B0 000D0        MOVW    VALUE, 16(R6)                   ; 0841
                         3A        2D     AB   E9 000D4        BLBC    45(CURRENT_VCB), 15$            ; 0846
                         20        0F     A7   91 000D8        CMPB    15(R7), #32                     ; 0848
                                          0D   13 000DC        BEQL    13$                             ;
               66        0F        A7     14   28 000DE        MOVC3   #20, 15(R7), (R6)               ; 0854
          14   A6        25        A7     0C   28 000E3        MOVC3   #12, 37(R7), 20(R6)             ; 0855
                                          27   11 000E9        BRB     15$                             ; 0848
               33524448   8F    0000G     DF   D1 000EB  13$:  CMPL    @HDR3, #861029448               ; 0861
                                          1C   12 000F4        BNEQ    15$                             ;
                                          52   D4 000F6        CLRL    I                               ; 0862
                                        6246   9F 000F8  14$:  PUSHAB  (I)[R6]                         ; 0863
                                0000GDF42      3F 000FB        PUSHAW  @HDR3[I]                        ;
                         6E               04   C0 00100        ADDL2   #4, (SP)                        ;
                                          08   DD 00103        PUSHL   #8                              ;
               00000000G  9F              03   FB 00105        CALLS   #3, @#LIB$CVT_HTB               ;
   FFE6        52        04               1C   F1 0010C        ACBL    #28, #4, I, 14$                 ;
               04     0000G      CF       01   E0 00112  15$:  BBS     #1, LOCAL_FIB+2, 16$            ; 0869
                         01      A6       08   88 00118        BISB2   #8, 1(R6)                       ; 0870
                         54        2E     A8   9E 0011C  16$:  MOVAB   46(R8), R4                      ;
               52     0000G      CF       04   C1 00120        ADDL3   #4, HDR1, STRING_PTR            ;
```

```
                62      09      2E  3A 00126          LOCC    #46, #9, (STRING_PTR)                   ;
                                02  12 0012A          BNEQ    17$                                     ;
                                51  D4 0012C          CLRL    R1                                      ;
                                51  D5 0012E 17$:     TSTL    CHAR_COUNT                              ;
                                05  12 00130          BNEQ    18$                                     ;
                            51      09  D0 00132      MOVL    #9, CHAR_COUNT                          ;
                                    03  11 00135      BRB     19$                                     ;
                            51      52  C2 00137 18$:  SUBL2   STRING_PTR, CHAR_COUNT                 ;
                59          52      51  C1 0013A 19$:  ADDL3   CHAR_COUNT, STRING_PTR, TYPE_PTR       ;
                50          51  D0 0013E              MOVL    CHAR_COUNT, I                           ;
                                08  11 00141          BRB     21$                                     ;
                20          6042    91  00143 20$:    CMPB    (I)[STRING_PTR], #32                    ;
                                05  12 00147          BNEQ    22$                                     ;
                                51  D7 00149          DECL    CHAR_COUNT                              ;
                F5              50  F4 0014B 21$:     SOBGEQ  I, 20$                                  ;
                                53  D4 0014E 22$:     CLRL    I                                       ;
                55              53  01  78 00150 23$:  ASHL    #1, I, R5                               ;
                                56  03  D0 00154      MOVL    #3, J                                   ;
                                6544    9F  00157 24$:  PUSHAB  (R5)[R4]                            ;
                                57  9E  3C 0015A      MOVZWL  @(SP)+, R7                              ;
                                57  28  C4 0015D      MULL2   #40, R7                                 ;
                                    6A  16 00160      JSB     GETCHAR                                 ;
                                6544    9F  00162      PUSHAB  (R5)[R4]                            ;
                9E              57  50  A1 00165      ADDW3   R0, R7, @(SP)+                          ;
                                    56  D7 00169      DECL    J                                       ;
                                    EA  12 0016B      BNEQ    24$                                     ;
                                    53  D6 0016D      INCL    I                                       ;
                            02      53  D1 0016F      CMPL    I, #2                                   ;
                                    DC  1B 00172      BLEQU   23$                                     ;
                            52      59  D0 00174      MOVL    TYPE_PTR, STRING_PTR                    ;
                            2E      62  91 00177      CMPB    (STRING_PTR), #46                       ;
                                    02  12 0017A      BNEQ    25$                                     ;
                                    52  D6 0017C      INCL    STRING_PTR                              ;
                            51      03  D0 0017E 25$:  MOVL    #3, CHAR_COUNT                          ;
                            50      02  D0 00181      MOVL    #2, I                                   ;
                            20          6042    91  00184 26$:  CMPB    (I)[STRING_PTR], #32            ;
                                        05  12 00188      BNEQ    27$                                 ;
                                        51  D7 0018A      DECL    CHAR_COUNT                          ;
                            F5          50  F4 0018C      SOBGEQ  I, 26$                              ;
                            53          03  D0 0018F 27$:  MOVL    #3, I                               ;
                            55      06  A4  3C 00192 28$:  MOVZWL  6(R4), R5                          ;
                            55          28  C4 00196      MULL2   #40, R5                             ;
                                        6A  16 00199      JSB     GETCHAR                             ;
        06      A4          55          50  A1 0019B      ADDW3   R0, R5, 6(R4)                       ;
                                        53  D7 001A0      DECL    I                                   ;
                                        EE  12 001A2      BNEQ    28$                                 ;
                                    08  A4  9F  001A4      PUSHAB  8(R4)                               ;
                    0000G   CF              01  FB 001A7      CALLS   #1, CALC_VERSION                ;
                                            01  DD 001AC      PUSHL   #1                              ; 0885
                                    47  A8  9F  001AE      PUSHAB  71(R8)                              ;
                7E      0000G   CF          29  C1 001B1      ADDL3   #41, HDR1, -(SP)                ;
                        0000V   CF          03  FB 001B7      CALLS   #3, FORMAT_DATEOD1              ;
                                            7E  D4 001BC      CLRL    -(SP)                           ; 0886
                                    54  A8  9F  001BE      PUSHAB  84(R8)                              ;
                7E      0000G   CF          2F  C1 001C1      ADDL3   #47, HDR1, -(SP)                ;
                        0000V   CF          03  FB 001C7      CALLS   #3, FORMAT_DATEOD1              ;
        5C      A8      0000G   DF  00F0    8F  28 001CC      MOVC3   #240, @HDRT, 92(R8)             ; 0891
```

```
                          50     01   A8  9A 001D5       MOVZBL  1(R8), R0                           ; 0903
                          50   6840       3E 001D9       MOVAW   (R8)[R0], MAP_AREA                  ;
                  06      A0   0301   8F  B0 001DD       MOVW    #769, 6(MAP_AREA)                   ; 0904
          52              58          50  C3 001E3       SUBL3   MAP_AREA, R8, R2                    ; 0907
                          52   01F4   C2  9E 001E7       MOVAB   500(R2), R2                         ;
          51              52          02  C7 001EC       DIVL3   #2, R2, R1                          ;
                  09      A0          51  90 001F0       MOVB    R1, 9(MAP_AREA)                     ;
                          51     FE   8F  9A 001F4       MOVZBL  #254, I                             ; 0925
                          50          58  D0 001F8 29$:  MOVL    R8, R0                              ; 0926
                          52          58  D0 001FB       MOVL    R8, R2                              ;
   01FE   C0      01FE    C2        6841  A1 001FE       ADDW3   (R8)[I], 510(R2), 510(R0)           ;
                          EE          51  F4 00207       SOBGEQ  I, 29$                              ;
                                          04 0020A       RET                                         ; 0929
```

; Routine Size: 523 bytes, Routine Base: $CODE$ + 0067

```
;  551     0930  1 ROUTINE FORMAT_DATEOD1 (INPDT, OUTPTR, TIME_REQUESTED) : NOVALUE =
;  552     0931  1
;  553     0932  1 !++
;  554     0933  1 !
;  555     0934  1 ! FUNCTIONAL DESCRIPTION:
;  556     0935  1 !       This routine formats an on-disk structure one date from the
;  557     0936  1 !       julian date on an Ansi labeled magnetic tape.
;  558     0937  1 !
;  559     0938  1 ! CALLING SEQUENCE:
;  560     0939  1 !       FORMAT_DATEOD1(ARG1,ARG2,ARG3)
;  561     0940  1 !
;  562     0941  1 ! INPUT PARAMETERS:
;  563     0942  1 !       ARG1 - address of input julian date
;  564     0943  1 !       ARG2 - address of output buffer
;  565     0944  1 !       ARG3 - 0 if time is not requested, 1 if time is requested
;  566     0945  1 !
;  567     0946  1 ! IMPLICIT INPUTS:
;  568     0947  1 !       none
;  569     0948  1 !
;  570     0949  1 ! OUTPUT PARAMETERS:
;  571     0950  1 !       ARG2 - address of output buffer
;  572     0951  1 !
;  573     0952  1 ! IMPLICIT OUTPUTS:
;  574     0953  1 !       none
;  575     0954  1 !
;  576     0955  1 ! ROUTINE VALUE:
;  577     0956  1 !       none
;  578     0957  1 !
;  579     0958  1 ! SIDE EFFECTS:
;  580     0959  1 !       date attribute written to output buffer (ddmmmyy)
;  581     0960  1 !       if time wanted, followed by (000000)
;  582     0961  1 !
;  583     0962  1 !--
;  584     0963  1
;  585     0964  2     BEGIN
;  586     0965  2
;  587     0966  2     LOCAL
;  588     0967  2         DATA    : BLOCK [ BYTE, 12 ],   ! work area in which date is formatted
;  589     0968  2         PTR,                            ! pointer to output buffer
;  590     0969  2         INPTR,                          ! pointer to input buffer
;  591     0970  2         CHAR;                           ! char from input string
;  592     0971  2
;  593     0972  2     EXTERNAL ROUTINE
;  594     0973  2         CONVDATE_J2R;                   ! convert ANSI Julian date to VMS date
;  595     0974  2
;  596     0975  2     ! convert julian date in hdr1 to ddmmmyy and if time is requested, return
;  597     0976  2     !  zeros
;  598     0977  2     ! assume output buffer initialized to zero
;  599     0978  2     !
;  600     0979  2     IF NOT CONVDATE_J2R(DATA, .INPDT) THEN RETURN;
;  601     0980  2
;  602     0981  2     (.OUTPTR)<0, 16> = .(DATA)<0, 16>;
;  603     0982  2     (.OUTPTR + 2)<0, 24> = .(DATA + 3)<0, 24>;
;  604     0983  2     (.OUTPTR + 5)<0, 16> = .(DATA + 9)<0, 16>;
;  605     0984  2
;  606     0985  2     IF .TIME_REQUESTED THEN CH$FILL('0', 6, (.OUTPTR + 7));
;  607     0986  2
```

```
;  608          0987  1     END;                                    ! end of routine


                                                                    .EXTRN  CONVDATE_J2R

                                                        003C 00000 FORMAT_DATEOD1:
                                                                    .WORD   Save R2,R3,R4,R5                    ; 0930
                                                 5E             0C  C2 00002            SUBL2   #12, SP
                                                          04    AC  DD 00005            PUSHL   INPDT          ; 0979
                                                          04    AE  9F 00008            PUSHAB  DATA
                                              0000G       CF        02  FB 0000B        CALLS   #2, CONVDATE_J2R
                                                          1E        50  E9 00010        BLBC    R0, 1$
                                                          50    08  AC  D0 00013        MOVL    OUTPTR, R0      ; 0981
                                                          60        6E  B0 00017        MOVW    DATA, (R0)
                02    A0                    18          00    03  AE  F0 0001A        INSV    DATA+3, #0, #24, 2(R0)  ; 0982
                                                    05    A0    09  AE  B0 00021        MOVW    DATA+9, 5(R0)   ; 0983
                                                          07    0C  AC  E9 00026        BLBC    TIME_REQUESTED, 1$  ; 0985
                      06                30          6E        00  2C  0002A        MOVC5   #0, (SP), #48, #6, 7(R0)
                                                          07    A0      0002F
                                                                04 00031 1$:    RET                         ; 0987

; Routine Size:  50 bytes,    Routine Base:  $CODE$ + 0272

;  609          0988  1
```

```
;  611     0989  1 GLOBAL ROUTINE FORMAT_FID (BUFFER) : NOVALUE =
;  612     0990  1
;  613     0991  1 !++
;  614     0992  1 !
;  615     0993  1 ! FUNCTIONAL DESCRIPTION:
;  616     0994  1 !       This routine formats the binary file id.
;  617     0995  1 !
;  618     0996  1 ! CALLING SEQUENCE:
;  619     0997  1 !       FORMAT_FID(ARG1)
;  620     0998  1 !
;  621     0999  1 ! INPUT PARAMETERS:
;  622     1000  1 !       ARG1 - address of buffer to receive 4 byte binary file id
;  623     1001  1 !
;  624     1002  1 ! IMPLICIT INPUTS:
;  625     1003  1 !       HDR1 in label area
;  626     1004  1 !
;  627     1005  1 ! OUTPUT PARAMETERS:
;  628     1006  1 !       ARG1 - address of buffer to received binary file id
;  629     1007  1 !
;  630     1008  1 ! IMPLICIT OUTPUTS:
;  631     1009  1 !       none
;  632     1010  1 !
;  633     1011  1 ! ROUTINE VALUE:
;  634     1012  1 !       none
;  635     1013  1 !
;  636     1014  1 ! SIDE EFFECTS:
;  637     1015  1 !       buffer receives binary file id
;  638     1016  1 !       1 word file number (HDR1 section number)
;  639     1017  1 !       1 word file sequence number (HDR1 sequence number)
;  640     1018  1 !
;  641     1019  1 !--
;  642     1020  1
;  643     1021  2     BEGIN
;  644     1022  2
;  645     1023  2     LOCAL
;  646     1024  2         RESULT;                                 ! long word work area for conversions
;  647     1025  2
;  648     1026  2     ! convert file number first
;  649     1027  2     !
;  650     1028  2     IF NOT LIB$CVT_DTB(HD1$S_FILESEQNO, HDR1[HD1$T_FILESEQNO], RESULT)
;  651     1029  2     THEN
;  652     1030  2         RESULT = 0;
;  653     1031  2
;  654     1032  2     ! store file number
;  655     1033  2     !
;  656     1034  2     (.BUFFER)<0, 16> = .RESULT<0, 16>;
;  657     1035  2
;  658     1036  2     ! convert file sequence number
;  659     1037  2     !
;  660     1038  2     IF NOT LIB$CVT_DTB(HD1$S_FILESECNO, HDR1[HD1$T_FILESECNO], RESULT)
;  661     1039  2     THEN
;  662     1040  2         RESULT = 0;
;  663     1041  2
;  664     1042  2     ! store file sequence number
;  665     1043  2     !
;  666     1044  2     (.BUFFER)<16, 16> = .RESULT<0, 16>;
;  667     1045  1     END;                                        ! end of routine
```

```
                                                     0004 00000          .ENTRY  FORMAT_FID, Save R2                             ; 0989
                                          52 00000000G  9F   9E 00002    MOVAB   @#LIB$CVT_DTB, R2                              ;
                                          5E            04   C2 00009    SUBL2   #4, SP                                         ;
                                                        5E   DD 0000C    PUSHL   SP                                             ; 1028
                                7E  0000G CF            1F   C1 0000E    ADDL3   #31, HDR1, -(SP)                               ;
                                                        04   DD 00014    PUSHL   #4                                             ;
                                          62            03   FB 00016    CALLS   #3, LIB$CVT_DTB                                ;
                                          02            50   E8 00019    BLBS    R0, 1$                                         ;
                                                        6E   D4 0001C    CLRL    RESULT                                         ; 1030
                                  04      BC            6E   B0 0001E 1$: MOVW    RESULT, @BUFFER                                ; 1034
                                                        5E   DD 00022    PUSHL   SP                                             ; 1038
                                7E  0000G CF            1B   C1 00024    ADDL3   #27, HDR1, -(SP)                               ;
                                                        04   DD 0002A    PUSHL   #4                                             ;
                                          62            03   FB 0002C    CALLS   #3, LIB$CVT_DTB                                ;
                                          02            50   E8 0002F    BLBS    R0, 2$                                         ;
                                                        6E   D4 00032    CLRL    RESULT                                         ; 1040
    04  BC                10              10            6E   F0 00034 2$: INSV    RESULT, #16, #16, @BUFFER                      ; 1044
                                                        04   00037    RET                                                    ; 1045

; Routine Size:  59 bytes,    Routine Base:  $CODE$ + 02A4

;  668          1046  1
;  669          1047  1 END
;  670          1048  1
;  671          1049  0 ELUDOM


;                                  PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $CODE$                         735  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         61         0        1000        00:01.8
```

```
;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:FRMOD1/OBJ=OBJ$:FRMOD1 MSRC$:FRMOD1/UPDATE=(ENH$:FRMOD1)

; Size:         719 code + 16 data bytes
; Run Time:        00:17.8
; Elapsed Time:    00:35.3
; Lines/CPU Min:    3528
; Lexemes/CPU-Min: 20468
; Memory Used:  223 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