
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**WTATTR

```
WW      WW  TTTTTTTTTT    AAAAAA    TTTTTTTTTT  TTTTTTTTTT  RRRRRRRR
WW      WW  TTTTTTTTTT    AAAAAA    TTTTTTTTTT  TTTTTTTTTT  RRRRRRRR
WW      WW      TT       AA    AA       TT          TT      RR      RR
WW      WW      TT       AA    AA       TT          TT      RR      RR
WW      WW      TT       AA    AA       TT          TT      RR      RR
WW      WW      TT       AA    AA       TT          TT      RR      RR
WW      WW      TT       AA    AA       TT          TT      RRRRRRRR
WW      WW      TT       AA    AA       TT          TT      RRRRRRRR
WW  WW  WW      TT       AAAAAAAAAA     TT          TT      RR  RR
WW  WW  WW      TT       AAAAAAAAAA     TT          TT      RR  RR
WWWW  WWWW      TT       AA    AA       TT          TT      RR    RR        ....
WWWW  WWWW      TT       AA    AA       TT          TT      RR    RR        ....
WW      WW      TT       AA    AA       TT          TT      RR      RR      ....
WW      WW      TT       AA    AA       TT          TT      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 WTATTR ( 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 handles the writing of attributes on a create
;    36          0036  1 !
;    37          0037  1 ! ENVIRONMENT:
;    38          0038  1 !
;    39          0039  1 !       STARLET operating system, including privileged system services
;    40          0040  1 !       and internal exec routines.
;    41          0041  1 !
;    42          0042  1 !--
;    43          0043  1
;    44          0044  1 !
;    45          0045  1 ! AUTHOR:  D. H. GILLESPIE,      CREATION DATE:  2-JUN-77
;    46          0046  1 !
;    47          0047  1 ! MODIFIED BY:
;    48          0048  1 !
;    49          0049  1 !       V03-002 LMP0221         L. Mark Pilant,         12-Apr-1984  12:11
;    50          0050  1 !               Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VFROT to
;    51          0051  1 !               ORB$W_PROT.
;    52          0052  1 !
;    53          0053  1 !       V03-002 MMD0278         Meg Dumont,     23-Mar-1984  10:09
;    54          0054  1 !               Change the processing of the accessibility character fields
;    55          0055  1 !               in the HDR1 label to call the installation
;    56          0056  1 !               specific accessibility routine. The return from this
;    57          0057  1 !               routine determines the users access to the file.
```

```
;    58          0058  1 !               Fix long file name support such that for ANSI version
;    59          0059  1 !               3 volumes it converts the exentsion length to
;    60          0060  1 !               ASCII chars before writing it to the label.
;    61          0061  1 !
;    62          0062  1 !       V03-001 MMD0156         Meg Dumont,     26-Apr-1983  9:16
;    63          0063  1 !               Add support fot interchange qualifier. Add support for long
;    64          0064  1 !               file name in WANSINAME, by spliting the long name between
;    65          0065  1 !               the HDR1 FILE_ID and the HDR4 lables.
;    66          0066  1 !
;    67          0067  1 !       V02-016 DMW00080        David Michael Walp       2-Mar-1982
;    68          0068  1 !               Added check for variable records greater then 9995
;    69          0069  1 !
;    70          0070  1 !       V02-015 DMW00074        David Michael Walp       5-Feb-1982
;    71          0071  1 !               Changed StatBlk to no-op
;    72          0072  1 !
;    73          0073  1 !       V02-014 DMW00066        David Michael Walp       8-Jan-1982
;    74          0074  1 !               Added HDR1 access character support
;    75          0075  1 !
;    76          0076  1 !       V02-013 DMW00061        David Michael Walp       7-Dec-1981
;    77          0077  1 !               Rename TRANSLATION_TABLE to ANSI_A_BAD
;    78          0078  1 !
;    79          0079  1 !       V02-012 DMW00051        David Michael Walp      10-Nov-1981
;    80          0080  1 !               Change SELECTONE to CASE, return length of ASCNAME
;    81          0081  1 !               ( not just a flag )
;    82          0082  1 !
;    83          0083  1 !       V02-011 DMW00046        David Michael Walp      28-Oct-1981
;    84          0084  1 !               Added ANSI "a" character support for file name, reformat
;    85          0085  1 !
;    86          0086  1 !       V02-010 DMW00042        David Michael Walp       2-Oct-1981
;    87          0087  1 !               Added support binary zero input dates
;    88          0088  1 !
;    89          0089  1 !       V02-009 DMW00029        David Michael Walp      10-Aug-1980
;    90          0090  1 !               Added stream file support and removed V2.0 HDR2 temp support
;    91          0091  1 !
;    92          0092  1 !       V02-008 DMW0004         David Michael Walp      11-Nov-1980
;    93          0093  1 !               New BLISS compiler, FUNCTION declaration changed from
;    94          0094  1 !               BBLOCK to BLOCK.  Old compiler used to give a longword
;    95          0095  1 !               with a declartion of "BBLOCK [1]".
;    96          0096  1 !
;    97          0097  1 !       V02-007 REFORMAT        Maria del C. Nasr        30-Jun-1980
;    98          0098  1 !
;    99          0099  1 !       V02-006 MCN0015         Maria del C. Nasr        10-Jun-1980
;   100          0100  1 !               Write form control attribute in HDR2 too, and clean
;   101          0101  1 !               undefined attributes of HDR3.
;   102          0102  1 !
;   103          0103  1 !       A0005   MCN0008         Maria del C. Nasr        22-Feb-1980   16:36
;   104          0104  1 !               Temporary support of RMS attributes in HDR2
;   105          0105  1 !
;   106          0106  1 !       A0004   MCN0005         Maria del C. Nasr        05-Nov-1979   11:30
;   107          0107  1 !               Eliminate [VCB$V_INTCHG] (interchange) switch
;   108          0108  1 !
;   109          0109  1 !       A0003   MCN0003         Maria del C. Nasr        16-Oct-1979   14:28
;   110          0110  1 !               Add HDR3 processing
;   111          0111  1 !
;   112          0112  1 !**
;   113          0113  1
;   114          0114  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
```

```
;  115          0115  1
;  116          0116  1 REQUIRE 'SRC$:MTADEF.B32';
;  117          0500  1
;  118          0501  1 FORWARD ROUTINE
;  119          0502  1     WANSINAME       :  COMMON_CALL NOVALUE,     ! write ANSI 'a' filename
;  120          0503  1     WBINDATE        :  NOVALUE,                 ! write 64_bit date
;  121          0504  1     WBLOCKSZ        :  COMMON_CALL NOVALUE,     ! write block size
;  122          0505  1     WEXPIREDT       :  NOVALUE,                 ! write expiration date
;  123          0506  1     WRECATTR        :  COMMON_CALL NOVALUE,     ! write record attributes
;  124          0507  1     WRITE_ATTRIBUTE :  COMMON_CALL NOVALUE;     ! write attributes, main
;  125          0508  1
;  126          0509  1 EXTERNAL
;  127          0510  1     CURRENT_UCB     : REF BBLOCK,               ! address of UCB
;  128          0511  1     HDR1            : REF BBLOCK,               ! address of HDR1 label
;  129          0512  1     HDR2            : REF BBLOCK,               ! address of HDR2 label
;  130          0513  1     HDR3            : REF BBLOCK,               ! address of HDR3 label
;  131          0514  1     HDR4            : REF BBLOCK,               ! address of HDR4 label
;  132          0515  1     NMBLOCK         : BBLOCK [10];              ! name block
;  133          0516  1
;  134          0517  1 EXTERNAL ROUTINE
;  135          0518  1     GET_RECORD,                                 ! get current record tape is reading
;  136          0519  1     SYS$FAO         : ADDRESSING_MODE (ABSOLUTE); ! change to hexadecimal
;  137          0520  1
;  138          0521  1 OWN
;  139          0522  1     MRS             : WORD,                     ! maximum record length
;  140          0523  1     VFCHDR          : WORD,                     ! length of VFC header
;  141          0524  1     BLOCKSZ;                                    ! block size
;  142          0525  1
;  143          0526  1 BIND
;  144          0527  1     CVT5 = DESCRIPTOR ('!5ZW'),
;  145          0528  1     CVT8 = DESCRIPTOR ('!8XL');
;  146          0529  1
;  147          0530  1 GLOBAL
;  148          0531  1     ANSI_NAME_SIZE  : SIGNED BYTE;
;  149          0532  1
```

```
;  151     0533  1 GLOBAL ROUTINE WRITE_ATTRIBUTE : COMMON_CALL NOVALUE =
;  152     0534  1
;  153     0535  1 !++
;  154     0536  1 !
;  155     0537  1 ! FUNCTIONAL DESCRIPTION:
;  156     0538  1 !       This routine writes attributes into HDR1 and HDR2
;  157     0539  1 !
;  158     0540  1 ! CALLING SEQUENCE:
;  159     0541  1 !       WRITE_ATTRIBUTE()
;  160     0542  1 !
;  161     0543  1 ! INPUT PARAMETERS:
;  162     0544  1 !       none
;  163     0545  1 !
;  164     0546  1 ! IMPLICIT INPUTS:
;  165     0547  1 !       IO_PACKET - address of current IO request packet
;  166     0548  1 !
;  167     0549  1 ! OUTPUT PARAMETERS:
;  168     0550  1 !       none
;  169     0551  1 !
;  170     0552  1 ! IMPLICIT OUTPUTS:
;  171     0553  1 !       none
;  172     0554  1 !
;  173     0555  1 ! ROUTINE VALUE:
;  174     0556  1 !       none
;  175     0557  1 !
;  176     0558  1 ! SIDE EFFECTS:
;  177     0559  1 !       none
;  178     0560  1 !
;  179     0561  1 !--
;  180     0562  1
;  181     0563  2     BEGIN
;  182     0564  2
;  183     0565  2     EXTERNAL REGISTER                               ! here so main expection
;  184     0566  2         COMMON_REG;                                 !  handler can touch the VCB
;  185     0567  2
;  186     0568  2     EXTERNAL
;  187     0569  2         ATR_TABLE    : BLOCKVECTOR [ATR$C_MAX_CODE, 1],  ! attribute table
;  188     0570  2         IO_PACKET    : REF BBLOCK,                  ! addr of current IRP
;  189     0571  2         ANSI_A_GOOD : VECTOR [ 256, BYTE ];         ! translation table
;  190     0572  2
;  191     0573  2     MACRO EQLLEN = 0, 24, 8, 0%;
;  192     0574  2     MACRO MAX    = 0,  0, 8, 0%;
;  193     0575  2
;  194     0576  2     LOCAL
;  195     0577  2         ACCESS_INPUT,                               ! set for whether access cahr specified
;  196     0578  2         CHAR    : VECTOR [4,BYTE],                  ! accessibility char to write to label
;  197     0579  2         CURRENT_RECORD,                             ! Record tape drive is reading
;  198     0580  2         STATUS,
;  199     0581  2         CHAR_INPUT : BYTE,                          ! Access char from ATR block
;  200     0582  2         DESCR    : VECTOR [2,LONG],                 ! descriptor of output for FAO
;  201     0583  2         MVL      : REF BBLOCK,                      ! MVL of volume
;  202     0584  2         PACKET   : REF BBLOCK,                      ! IO packet address
;  203     0585  2         ABD      : REF BBLOCKVECTOR [, ABD$C_LENGTH],     ! addr of desc vector
;  204     0586  2         ORB      : REF BBLOCK,                      ! Object's rights block address
;  205     0587  2         BITS,                                       ! number of bits set
;  206     0588  2         P,                                          ! pointer to attribute
;  207     0589  2         CODE,                                       ! attribute code
```

```
;  208          0590  2          COUNT;                                  ! number of char in attribute
;  209          0591  2
;  210          0592  2
;  211          0593  2      PACKET          = .IO_PACKET;
;  212          0594  2      ABD             = .BBLOCK[.PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
;  213          0595  2      MRS             = 0;                        ! default record size
;  214          0596  2      BLOCKSZ         = 512;                      ! default record size
;  215          0597  2      VFCHDR          = 0;                        ! initialize VFC header count
;  216          0598  2      ANSI_NAME_SIZE  = -1;                       ! no ANSI file name
;  217          0599  2      ACCESS_INPUT    = MTA$K_NOCHAR;             ! Assume no access char passed
;  218          0600  2
;  219          0601  2      INCRU I FROM ABD$C_ATTRIB TO .PACKET[IRP$W_BCNT] - 1 DO
;  220          0602  3          BEGIN
;  221          0603  3
;  222          0604  3          ! set up a pointer to the attribute and set up the size and
;  223          0605  3          ! find the attribute code
;  224          0606  3
;  225          0607  3          P     = .ABD[.I, ABD$W_TEXT] + ABD[.I, ABD$W_TEXT];
;  226          0608  3          COUNT = .ABD[.I, ABD$W_COUNT];
;  227          0609  3          CODE  = .(.P)<0, 8>;
;  228          0610  3          P     = .P + 1;
;  229          0611  3
;  230          0612  3          ! test for table length
;  231          0613  3
;  232          0614  3          IF .CODE GTRU ATR$C_MAX_CODE THEN ERR_EXIT(SS$_BADATTRIB);
;  233          0615  3
;  234          0616  3          ! check attribute length
;  235          0617  3
;  236          0618  3          IF .ATR_TABLE[(.CODE-1),EQLLEN]
;  237          0619  3          THEN
;  238          0620  4              BEGIN
;  239          0621  4              IF .COUNT NEQU .ATR_TABLE[(.CODE-1),MAX]
;  240          0622  4              THEN ERR_EXIT(SS$_BADATTRIB);
;  241          0623  4              END
;  242          0624  3          ELSE
;  243          0625  4              BEGIN
;  244          0626  4                  IF .COUNT GTRU .ATR_TABLE[(.CODE-1),MAX]
;  245          0627  4                      THEN ERR_EXIT(SS$_BADATTRIB);
;  246          0628  3              END;
;  247          0629  3
;  248          0630  3          ! call the correct procedure
;  249          0631  3
;  250          0632  3          CASE .CODE FROM 1 TO ATR$C_MAX_CODE OF
;  251          0633  3              SET
;  252          0634  3
;  253          0635  3              [ ATR$C_RECATTR     ] : WRECATTR  ( .P, .COUNT );
;  254          0636  3              [ ATR$C_FILNAM      ] : CH$MOVE   ( .COUNT, .P, NMBLOCK );
;  255          0637  3              [ ATR$C_FILTYP      ] : CH$MOVE   ( .COUNT, .P, NMBLOCK + 6 );
;  256          0638  3              [ ATR$C_FILVER      ] : CH$MOVE   ( .COUNT, .P, NMBLOCK + 8 );
;  257          0639  3              [ ATR$C_EXPDAT      ] : WEXPIREDT ( .P, .COUNT );
;  258          0640  3              [ ATR$C_BLOCKSIZE   ] : WBLOCKSZ  ( .P, .COUNT );
;  259          0641  3              [ ATR$C_ASCNAME     ] : WANSINAME ( .P, .COUNT );
;  260          0642  3              [ ATR$C_CREDATE     ] : WBINDATE  ( .P, .COUNT,
;  261          0643  3                                                  BBLOCK[.HDR1,HD1$T_CREATEDT]);
;  262          0644  3              [ ATR$C_EXPDATE     ] : WBINDATE  ( .P, .COUNT,
;  263          0645  3                                                  BBLOCK[.HDR1,HD1$T_EXPIREDT]);
;  264          0646  4              [ ATR$C_HDR1_ACC    ] : BEGIN
```

```
;  265     0647  4                        CHAR_INPUT = .ANSI_A_GOOD [ .(.P)<0,8> ];
;  266     0648  4                        ACCESS_INPUT = MTA$K_CHARVALID;
;  267     0649  3                    END;
;  268     0650  3
;  269     0651  3            ! These are either read only attributes or unimplemented ones
;  270     0652  3
;  271     0653  3            [ ATR$C_HEADER,
;  272     0654  3              ATR$C_USERLABEL,
;  273     0655  3              ATR$C_ENDLBLAST,
;  274     0656  3              ATP$C_UIC_RO      ] : ERR_EXIT ( SS$_BADATTRIB );
;  275     0657  3
;  276     0658  3            ! The following are valid for DISK but not TAPE, so we will drop
;  277     0659  3            !  them on the floor ( NO OP ) for device independence sake
;  278     0660  3            ! If these values are asked for during read attributes we will
;  279     0661  3            !  dummy them up
;  280     0662  3            ! INRANGE statement is used for speed and code size sake
;  281     0663  3
;  282     0664  3 !          [ 1,
;  283     0665  3 !            2,
;  284     0666  3 !            ATR$C_UCHAR,
;  285     0667  3 !            ATR$C_STATBLK,
;  286     0668  3 !            ATR$C_ASCDATES,   !!!!!!!!!!!!!! This One Should be Fixed !!!!!!!
;  287     0669  3 !            ATR$C_ALCONTROL,
;  288     0670  3 !            ATR$C_REVDATE,
;  289     0671  3 !            ATR$C_BAKDATE,
;  290     0672  3 !            ATR$C_UIC,
;  291     0673  3 !            ATR$C_FPRO,
;  292     0674  3 !            ATR$C_RPRO,
;  293     0675  3 !            ATR$C_ACLEVEL,
;  294     0676  3 !            ATR$C_SEMASK,
;  295     0677  3 !            ATR$C_DIRSEQ,
;  296     0678  3 !            ATR$C_BACKLINK,
;  297     0679  3 !            ATR$C_JOURNAL     ] : ;
;  298     0680  3                [ INRANGE           ] : ;
;  299     0681  3                TES;
;  300     0682  2            END;
;  301     0683  2
;  302     0684  2        ! Call the accessibility system service to determine the accessibility char
;  303     0685  2        ! to write on the label.
;  304     0686  2        ! First keep the record that the UCB is reading. The accessibility
;  305     0687  2        ! routine can not move the tape from under us! Thus we will compare
;  306     0688  2        ! this to the field after the call and if the tape was moved we punt
;  307     0689  2        ! the operation.
;  308     0690  2
;  309     0691  2        MVL = .CURRENT_VCB[VCB$L_MVL];
;  310     0692  2        ORB = .CURRENT_UCB[UCB$L_ORB];
;  311     0693  2        CURRENT_RECORD = KERNEL_CALL(GET_RECORD, .CURRENT_UCB);
;  312   P 0694  2        CHAR = $MTACCESS(LBLNAM = .HDR1,
;  313   P 0695  2                         UIC = .ORB[ORB$L_OWNER],
;  314   P 0696  2                         STD_VERSION = .MVL[MVL$B_STDVER],
;  315   P 0697  2                         ACCESS_CHAR = .CHAR_INPUT,
;  316   P 0698  2                         ACCESS_SPEC = .ACCESS_INPUT,
;  317     0699  2                         TYPE = MTA$K_OUTHDR1);
;  318     0700  2
;  319     0701  2        STATUS = KERNEL_CALL(GET_RECORD, .CURRENT_UCB);
;  320     0702  2        IF .CURRENT_RECORD NEQ .STATUS
;  321     0703  2            THEN ERR_EXIT(SS$_TAPEPOSLOST);
```

```
;  322     0704  2
;  323     0705  2      ! Write the char to the label.
;  324     0706  2
;  325     0707  2      HDR1[HD1$B_FILACCESS] = .CHAR[0];
;  326     0708  2
;  327     0709  2      ! after both blocksize and maximum record size attributes have been
;  328     0710  2      ! processed then determine the value of reclen in HDR2
;  329     0711  2
;  330     0712  2      IF .MRS EQLU 0
;  331     0713  2      THEN
;  332     0714  3          (CH$MOVE(HD2$S_BLOCKLEN, HDR2[HD2$T_BLOCKLEN], HDR2[HD2$T_RECLEN]))
;  333     0715  2      ELSE
;  334     0716  3          BEGIN
;  335     0717  3
;  336     0718  3          ! count overhead for ANSI 'D' record
;  337     0719  3
;  338     0720  3          IF .HDR2[HD2$B_RECFORMAT] EQL 'D' THEN MRS = .MRS + 4 + .VFCHDR;
;  339     0721  3
;  340     0722  3          ! FAO output descriptor
;  341     0723  3
;  342     0724  3          DESCR[0] = HD2$S_RECLEN;
;  343     0725  3          DESCR[1] = HDR2[HD2$T_RECLEN];
;  344     0726  3          SYS$FAO(CVT5, 0, DESCR, .MRS);
;  345     0727  2          END;
;  346     0728  2
;  347     0729  2      ! now check
;  348     0730  2      !     that the block size is valid                                                    ;
;  349     0731  2      ! and that the record size is not bigger then the block size
;  350     0732  2      ! and that variable records are not larger then can be supported ( RCW )
;  351     0733  2
;  352     0734  3      IF     ( .BLOCKSZ LSSU 18 )                                                         ;
;  353     0735  3         OR ( .MRS GTRU .BLOCKSZ )
;  354     0736  3         OR ( ( .MRS GTRU (9999 - 4) ) AND ( .HDR2 [ HD2$B_RECFORMAT ] EQL 'D' ))
;  355     0737  2      THEN ERR_EXIT(SS$_BADATTRIB);
;  356     0738  2
;  357     0739  1      END;


                                                .TITLE  WTATTR
                                                .IDENT  \V04-000\

                                                .PSECT  $CODE$,NOWRT,2

                                57  5A  35  21  00000 P.AAB:   .ASCII  \!5ZW\                                ;
                                      00000004  00004 P.AAA:   .LONG   4
                                      00000000' 00008          .ADDRESS P.AAB
                                4C  58  38  21  0000C P.AAD:   .ASCII  \!8XL\                                ;
                                      00000004  00010 P.AAC:   .LONG   4
                                      00000000' 00014          .ADDRESS P.AAD

                                                .PSECT  $LOCKEDD1$,NOEXE,2

                                          00000 MRS:     .BLKB   2
                                          00002 VFCHDR:  .BLKB   2
                                          00004 BLOCKSZ: .BLKB   4
                                          00008 ANSI_NAME_SIZE::
                                                .BLKB   1
```

```
                                CVT5=           P.AAA
                                CVT8=           P.AAC
                                        .EXTRN  CURRENT_UCB, HDR1
                                        .EXTRN  HDR2, HDR3, HDR4
                                        .EXTRN  NMBLOCK, GET_RECORD
                                        .EXTRN  SYS$FAO, ATR_TABLE
                                        .EXTRN  IO_PACKET, ANSI_A_GOOD
                                        .EXTRN  SYS$CMKRNL, SYS$MTACCESS

                                        .PSECT  $CODE$,NOWRT,2

                        07FC 00000      .ENTRY  WRITE_ATTRIBUTE, Save R2,R3,R4,R5,R6,R7,R8,-; 0533
                                                R9,R10
                5E              08  C2 00002       SUBL2   #8, SP
                50      0000G   CF  D0 00005       MOVL    IO_PACKET, PACKET                 0593
                5A        2C    B0  D0 0000A       MOVL    @44(PACKET), ABD                  0594
0000'           CF      0200    8F  3C 0000E       MOVZWL  #512, BLOCKSZ                     0596
                        0000'   CF  D4 00015       CLRL    MRS                               0595
0000'           CF              01  8E 00019       MNEGB   #1, ANSI_NAME_SIZE                0598
                                7E  D4 0001E       CLRL    ACCESS_INPUT                      0599
                7E        32    A0  3C 00020       MOVZWL  50(PACKET), -(SP)                 0601
                                6E  D7 00024       DECL    (SP)
                57              05  D0 00026       MOVL    #5, I
                            010C    30 00029       BSBW    20$
                50              6A47 7E 0002C 1$:  MOVAQ   (ABD)[I], R0                     0607
                58                60 3C 00030       MOVZWL  (R0), P
                58                50 C0 00033       ADDL2   R0, P
                            02  AA47   7F 00036       PUSHAQ  2(ABD)[I]                     0608
                59                9E 3C 0003A       MOVZWL  @(SP)+, COUNT
                56                88 9A 0003D       MOVZBL  (P)+, CODE                    0609
                30                56 D1 00040       CMPL    CODE, #48                     0614
                                  02 1B 00043       BLEQU   2$
                                  34 BF 00045       CHMU    #52
                          0E  0000GCF46 E9 00047 2$:  BLBC    ATR_TABLE-1[CODE], 3$         0618
                          0000GCF46     DF 0004D      PUSHAL  ATR_TABLE-4[CODE]             0621
        59        9E        08      00  ED 00052      CMPZV   #0, #8, @(SP)+, COUNT
                                    10  13 00057      BEQL    5$
                                    0C  11 00059      BRB     4$                            0622
                          0000GCF46     DF 0005B 3$:  PUSHAL  ATR_TABLE-4[CODE]             0626
        59        9E        08      00  ED 00060      CMPZV   #0, #8, @(SP)+, COUNT
                                    02  1E 00065      BGEQU   5$
                                    34  BF 00067 4$:  CHMU    #52                           0627
                  2F        01      56  CF 00069 5$:  CASEL   CODE, #1, #47                 0632
      0060      00C9      00C9      00C9      0006D 6$:  .WORD   19$-6$,-
      0082      007A      0072      006A      00075      19$-6$,-
      00C7      008C      00C7      00C9      0007D      19$-6$,-
      0096      00C7      00C9      00C9      00085      7$-6$,-
      00C9      00A8      00C9      00A0      0008D      8$-6$,-
      00C9      00C9      00C9      00C9      00095      9$-6$,-
      00C9      00C9      00C7      00C9      0009D      10$-6$,-
      00C9      00C9      00B8      00C9      000A5      11$-6$,-
      00C9      00C9      00C9      00C9      000AD      19$-6$,-
      00C9      00C9      00C9      00C9      000B5      18$-6$,-
      00C9      00C9      00C9      00C9      000BD      12$-6$,-
      00C9      00C9      00C9      00C9      000C5      18$-6$,-
                                                         19$-6$,-
```

```
                                                                  19$-6$,-                                ;
                                                                  18$-6$,-                                ;
                                                                  13$-6$,-                                ;
                                                                  14$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  15$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  18$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  17$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$,-                                ;
                                                                  19$-6$                                  ;
                         7E          58  7D 000CD 7$:    MOVQ     P, -(SP)                                ; 0635
              0000V      CF          02  FB 000D0        CALLS    #2, WRECATTR                            ;
                                     5F  11 000D5        BRB      19$                                     ;
   0000G  CF             68          59  28 000D7 8$:    MOVC3    COUNT, (P), NMBLOCK                     ; 0636
                                     57  11 000DD        BRB      19$                                     ;
   0000G  CF             68          59  28 000DF 9$:    MOVC3    COUNT, (P), NMBLOCK+6                   ; 0637
                                     4F  11 000E5        BRB      19$                                     ;
   0000G  CF             68          59  28 000E7 10$:   MOVC3    COUNT, (P), NMBLOCK+8                   ; 0638
                                     47  11 000ED        BRB      19$                                     ;
                         7E          58  7D 000EF 11$:   MOVQ     P, -(SP)                                ; 0639
              0000V      CF          02  FB 000F2        CALLS    #2, WEXPIREDT                           ;
                                     3D  11 000F7        BRB      19$                                     ;
                         7E          58  7D 000F9 12$:   MOVQ     P, -(SP)                                ; 0640
              0000V      CF          02  FB 000FC        CALLS    #2, WBLOCKSZ                            ;
                                     33  11 00101        BRB      19$                                     ;
                         7E          58  7D 00103 13$:   MOVQ     P, -(SP)                                ; 0641
              0000V      CF          02  FB 00106        CALLS    #2, WANSINAME                           ;
                                     29  11 0010B        BRB      19$                                     ;
          7E  0000G      CF          29  C1 0010D 14$:   ADDL3    #41, HDR1, -(SP)                        ; 0643
                                     06  11 00113        BRB      16$                                     ;
          7E  0000G      CF          2F  C1 00115 15$:   ADDL3    #47, HDR1, -(SP)                        ; 0645
                         7E          58  7D 0011B 16$:   MOVQ     P, -(SP)                                ;
```

```
                0000V  CF              03  FB 0011E             CALLS   #3, WBINDATE                                    ;
                                       11  11 00123             BRB     19$                                             ;
                       50              68  9A 00125 17$:       MOVZBL  (P), R0                                         ; 0647
                       6E      0000GCF40  90 00128             MOVB    ANSI_A_GOOD[R0], CHAR_INPUT                     ;
                    08    AE           01  D0 0012E             MOVL    #1, ACCESS_INPUT                                ; 0648
                                       02  11 00132             BRB     19$                                             ; 0632
                                       34  BF 00134 18$:       CHMU    #52                                             ; 0656
                                       57  D6 00136 19$:       INCL    I                                               ; 0601
                    04    AE           57  D1 00138 20$:       CMPL    I, 4(SP)                                        ;
                                       03  1A 0013C             BGTRU   21$                                             ;
                                   FEEB  31 0013E             BRW     1$                                              ;
                       52        34    AB  D0 00141 21$:       MOVL    52(CURRENT_VCB), MVL                            ; 0691
                       50      0000G   CF  D0 00145             MOVL    CURRENT_UCB, R0                                 ; 0692
                       53        1C    A0  D0 0014A             MOVL    28(R0), ORB                                     ;
                                       50  DD 0014E             PUSHL   R0                                              ; 0693
                                       01  DD 00150             PUSHL   #1                                              ;
                                       5E  DD 00152             PUSHL   SP                                              ;
                                     0000G CF  9F 00154         PUSHAB  GET_RECORD                                      ;
                00000000G              9F      04  FB 00158     CALLS   #4, @#SYS$CMKRNL                                ;
                       54                      50  D0 0015F     MOVL    R0, CURRENT_RECORD                              ;
                                               03  DD 00162     PUSHL   #3                                              ; 0699
                                         0C    AE  DD 00164     PUSHL   ACCESS_INPUT                                    ;
                       7E                08    AE  9A 00167     MOVZBL  CHAR_INPUT, -(SP)                               ;
                       7E                22    A2  9A 0016B     MOVZBL  34(MVL), -(SP)                                  ;
                                               63  DD 0016F     PUSHL   (ORB)                                           ;
                                             0000G CF  DD 00171 PUSHL   HDR1                                            ;
                00000000G              00      06  FB 00175     CALLS   #6, SYS$MTACCESS                                ;
                       52                      50  D0 0017C     MOVL    R0, CHAR                                        ;
                                             0000G CF  DD 0017F PUSHL   CURRENT_UCB                                     ; 0701
                                               01  DD 00183     PUSHL   #1                                              ;
                                               5E  DD 00185     PUSHL   SP                                              ;
                                             0000G CF  9F 00187 PUSHAB  GET_RECORD                                      ;
                    00000000G                  9F      04  FB 0018B CALLS   #4, @#SYS$CMKRNL                        ;
                                               50          54  D1 00192 CMPL    CURRENT_RECORD, STATUS                  ; 0702
                                                           04  13 00195 BEQL    22$                                     ;
                                             0224  8F      BF 00197     CHMU    #548                                    ; 070?
                                               50  0000G   CF  D0 0019B 22$:   MOVL    HDR1, R0                        ; 0707
                                                35    A0          52  90 001A0 MOVB    CHAR, 53(R0)                    ;
                                               56  0000G   CF  D0 001A4 MOVL    HDR2, R6                                ; 0714
                                               57  0000'   CF  3C 001A9 MOVZWL  MRS  R7                                 ; 0712
                                                           08  12 001AE BNEQ    23$                                     ;
                0A    A6                 05    A6          05  28 001B0 MOVC3   #5, 5(R6), 10(R6)                       ; 0714
                                                           34  11 001B6 BRB     25$                                     ;
                                               44  8F        04    A6  91 001B8 23$:   CMPB    4(R6), #68              ; 0720
                                                           0F  12 001BD BNEQ    24$                                     ;
                                               50  0000'   CF  3C 001BF MOVZWL  VFCHDR, R0                             ;
                                               51    04  A740  9E 001C4 MOVAB   4(R7)[R0], R1                          ;
                                             0000'   CF          51  B0 001C9 MOVW    R1, MRS                            ;
                                               0C    AE          05  D0 001CE 24$:   MOVL    #5, DESCR               ; 0724
                                               10    AE    0A    A6  9E 001D2 MOVAB   10(R6), DESCR+4                 ; 0725
                                               7E          0000'   CF  3C 001D7 MOVZWL  MRS, -(SP)                     ; 0726
                                                             10    AE  9F 001DC PUSHAB  DESCR                          ;
                                                             7E      D4 001DF CLRL    -(SP)                            ;
                                                           FE07    CF  9F 001E1 PUSHAB  CVT5                           ;
                    00000000G                  9F      04  FB 001E5     CALLS   #4, @#SYS$FAO                       ;
                                               12  0000'   CF  D1 001EC 25$:   CMPL    BLOCKSZ, #18                    ; 0734
                                                           20  1F 001F1 BLSSU   26$                                     ;
```

```
    0000' CF     0000' CF          10          00 ED 001F3          CMPZV   #0, #16, MRS, BLOCKSZ          ; 0735
                                                 15 1A 001FC          BGTRU   26$                            ;
                                270B 8F    0000' CF B1 001FE          CMPW    MRS, #9995                     ; 0736
                                                 0E 1B 00205          BLEQU   27$                            ;
                                     50    0000G CF D0 00207          MOVL    HDR2, R0                       ;
                                  44 8F      04 A0 91 0020C          CMPB    4(R0), #68                     ;
                                                 02 12 00211          BNEQ    27$                            ;
                                                 34 BF 00213 26$:     CHMU    #52                            ; 0737
                                                    04 00215 27$:     RET                                    ; 0739

; Routine Size:  534 bytes,    Routine Base:  $CODE$ + 0018
```

```
;  359     0740  1 ROUTINE WEXPIREDT (PTER, COUNT) :  NOVALUE =
;  360     0741  1
;  361     0742  1 !++
;  362     0743  1 !
;  363     0744  1 ! FUNCTIONAL DESCRIPTION:
;  364     0745  1 !       This routine puts the expiration attribute in HDR1
;  365     0746  1 !
;  366     0747  1 ! CALLING SEQUENCE:
;  367     0748  1 !       WEXPIREDT(ARG1,ARG2)
;  368     0749  1 !
;  369     0750  1 ! INPUT PARAMETERS:
;  370     0751  1 !       ARG1 - address of attribute(ddmmmyy)
;  371     0752  1 !       ARG2 - number of characters in attribute
;  372     0753  1 !
;  373     0754  1 ! IMPLICIT INPUTS:
;  374     0755  1 !       none
;  375     0756  1 !
;  376     0757  1 ! OUTPUT PARAMETERS:
;  377     0758  1 !       none
;  378     0759  1 !
;  379     0760  1 ! IMPLICIT OUTPUTS:
;  380     0761  1 !       expiration date in HDR1 updated
;  381     0762  1 !
;  382     0763  1 ! ROUTINE VALUE:
;  383     0764  1 !       none
;  384     0765  1 !
;  385     0766  1 ! SIDE EFFECTS:
;  386     0767  1 !       none
;  387     0768  1 !
;  388     0769  1 ! USER ERRORS:
;  389     0770  1 !       SS$_BADATTRIB - attribute length must equal 7 or illegal date
;  390     0771  1 !
;  391     0772  1 !--
;  392     0773  1
;  393     0774  2     BEGIN
;  394     0775  2
;  395     0776  2     EXTERNAL
;  396     0777  2         ZERO_JDATE : VECTOR [ 6, BYTE ];! zero Julian Date of " 00000"
;  397     0778  2
;  398     0779  2     EXTERNAL ROUTINE
;  399     0780  2         CONVDATE_R2J;                   ! convert date from reg to JULIAN(ANSI)
;  400     0781  2
;  401     0782  2     LOCAL
;  402     0783  2         DATE    : VECTOR [12, BYTE];    ! work area for date with dashes
;  403     0784  2
;  404     0785  3     IF (.(.PTER) EQL 0) AND (.(.PTER+4) EQL 0)
;  405     0786  2     THEN
;  406     0787  2
;  407     0788  2         ! if expiration is binary zero, then no date thus stuff zero date
;  408     0789  2
;  409     0790  3         (CH$MOVE ( 6, ZERO_JDATE, BBLOCK[.HDR1, HD1$T_EXPIREDT] ))
;  410     0791  3
;  411     0792  2     ELSE
;  412     0793  3         BEGIN
;  413     0794  3
;  414     0795  3         ! now format DD-MMM-YYYY from DDMMMYY
;  415     0796  3
```

```
;  416          0797  3          (DATE)<0, 16> = .(.PTER);                ! day
;  417          0798  3          (DATE + 2)<0, 8> = '-';                  ! dash
;  418          0799  3          (DATE + 3)<0, 24> = .(.PTER + 2);        ! month
;  419          0800  3          (DATE + 6)<0, 24> = '-19'                ! century
;  420          0801  3          (DATE + 9)<0, 16> = .(.PTER + 5);        ! year
;  421          0802  3
;  422          0803  3          IF NOT CONVDATE_R2J(DATE, BBLOCK[.HDR1, HD1$T_EXPIREDT])
;  423          0804  3          THEN
;  424          0805  3              ERR_EXIT(SS$_BADATTRIB);
;  425          0806  3
;  426          0807  2          END;
;  427          0808  1      END;


                                                        .EXTRN  ZERO_JDATE, CONVDATE_R2J

                                        007C 00000 WEXPIREDT:
                                                        .WORD   Save R2,R3,R4,R5,R6                     ; 0740
                                    5E                  0C  C2 00002        SUBL2   #12, SP
                                    56          04      AC  D0 00005        MOVL    PTER, R6            ; 0785
                                                        66  D5 00009        TSTL    (R6)
                                                        12  12 0000B        BNEQ    1$
                                                04      A6  D5 0000D        TSTL    4(R6)
                                                        0D  12 00010        BNEQ    1$
                                    50          0000G   CF  D0 00012        MOVL    HDR1, R0            ; 0790
                                2F  A0          0000G   CF      06  28 00017    MOVC3   #6, ZERO_JDATE, 47(R0)
                                                            04  0001E        RET
                                    6F                  66  B0 0001F 1$:    MOVW    (R6), DATE          ; 0797
                                      02        AE      2D  90 00022        MOVB    #45, DATE+2         ; 0798
                          03    AE          18          00      02  A6  F0 00026    INSV    2(R6), #0, #24, DATE+3  ; 0799
                                                        09  AE  9F 0002D    PUSHAB  DATE+9              ; 0801
                                  0039312D  9F              01  FB 00030    CALLS   #1, @#^X0039312D
                                    51          05      A6  D0 00037        MOVL    5(R6), R1
                                    60                  51  B0 0003B        MOVW    R1, (R0)
                          06    AE          18          00      51  F0 0003E    INSV    R1, #0, #24, DATE+6
                                7E          0000G   CF      2F  C1 00044    ADDL3   #47, HDR1, -(SP)    ; 0803
                                                    04  AE  9F 0004A        PUSHAB  DATE
                                            0000G   CF      02  FB 0004D    CALLS   #2, CONVDATE_R2J
                                            02              50  E8 00052    BLBS    R0, 2$
                                                            34  BF 00055    CHMU    #52                 ; 0805
                                                                04 00057 2$:    RET                     ; 0808

; Routine Size:  88 bytes,    Routine Base:  $CODE$ + 022E
```

```
;  429     0809  1 ROUTINE WRECATTR (PTER, COUNT) :  COMMON_CALL NOVALUE =
;  430     0810  1
;  431     0811  1 !++
;  432     0812  1 !
;  433     0813  1 ! FUNCTIONAL DESCRIPTION:
;  434     0814  1 !       This routine records user attributes in HDR2 and HDR3
;  435     0815  1 !
;  436     0816  1 ! CALLING SEQUENCE:
;  437     0817  1 !       WRECATTR(ARG1,ARG2)
;  438     0818  1 !
;  439     0819  1 ! INPUT PARAMETERS:
;  440     0820  1 !       ARG1 - address of attribute
;  441     0821  1 !       ARG2 - size of attribute
;  442     0822  1 !
;  443     0823  1 ! IMPLICIT INPUTS:
;  444     0824  1 !       none
;  445     0825  1 !
;  446     0826  1 ! OUTPUT PARAMETERS:
;  447     0827  1 !       none
;  448     0828  1 !
;  449     0829  1 ! IMPLICIT OUTPUTS:
;  450     0830  1 !       HDR2 and HDR3 updated to reflect user attributes
;  451     0831  1 !
;  452     0832  1 ! ROUTINE VALUE:
;  453     0833  1 !       none
;  454     0834  1 !
;  455     0835  1 ! SIDE EFFECTS:
;  456     0836  1 !       none
;  457     0837  1 !
;  458     0838  1 !--
;  459     0839  1
;  460     0840  2     BEGIN
;  461     0841  2
;  462     0842  2     EXTERNAL REGISTER
;  463     0843  2         COMMON_REG;                             !  Set up reference to the VCB
;  464     0844  2
;  465     0845  2     LITERAL
;  466     0846  2         RECTYPE_LEN = 8,
;  467     0847  2         RECATTR_LEN = 6;
;  468     0848  2
;  469     0849  2     BIND
;  470     0850  2         RECTYPE_ANSI = UPLIT BYTE('UFDVDDDS')           : VECTOR [, BYTE],
;  471     0851  2         RECTYPE_FILE11 = UPLIT BYTE(0,1,2,3,4,5,6,0)    : VECTOR [, BYTE],
;  472     0852  2         RECATTR_ASCI = UPLIT BYTE('MA MA ')             : VECTOR [, BYTE],
;  473     0853  2         RECATTR_FILE11 = UPLIT BYTE(0, 1, 2, 8, 9, 10)  : VECTOR [, BYTE];
;  474     0854  2
;  475     0855  2     LOCAL
;  476     0856  2         ATTRIBUTES      : VECTOR [CH$ALLOCATION(32)],   ! store good attributes
;  477     0857  2         X,                                              ! for support of HDR2 attributes
;  478     0858  2         DESCR   : VECTOR [2],                           ! general descriptor
;  479     0859  2         PARTIAL_CNT;                                    ! count of attributes to transfer
;  480     0860  2
;  481     0861  2     MAP
;  482     0862  2         PTER    : REF BBLOCK;                           ! address of attribute
;  483     0863  2
;  484     0864  2     ! capture valid attributes only
;  485     0865  2
```

```
;  486          0866  2      CH$COPY(.COUNT, .PTER, 0, 32, ATTRIBUTES);
;  487          0867  2
;  488          0868  2      ! is record type present
;  489          0869  2
;  490          0870  3      IF .COUNT GEQU ($BYTEOFFSET(FAT$B_RTYPE) + 1)
;  491          0871  2      THEN
;  492          0872  3          BEGIN
;  493          0873  3          HDR2[HD2$B_RECFORMAT] = %C'U';
;  494          0874  3
;  495          0875  3          DECR I FROM RECTYPE_LEN - 1 TO 0 DO
;  496          0876  4              BEGIN
;  497          0877  4
;  498          0878  4              IF .RECTYPE_FILE11[.I] EQL .PTER[FAT$B_RTYPE]
;  499          0879  4              THEN
;  500          0880  4                  HDR2[HD2$B_RECFORMAT] = .RECTYPE_ANSI[.I];
;  501          0881  4
;  502          0882  3              END;
;  503          0883  3
;  504          0884  3          IF .HDR2[HD2$B_RECFORMAT] EQLU 'V'
;  505          0885  3          THEN
;  506          0886  4              BEGIN
;  507          0887  4              HDR2[HD2$B_RECFORMAT] = 'D';
;  508          0888  4
;  509          0889  5              IF .COUNT GEQU ($BYTEOFFSET(FAT$B_VFCSIZE) + 1)
;  510          0890  4              THEN
;  511          0891  4                  VFCHDR = .PTER[FAT$B_VFCSIZE]    ! get VFC header size
;  512          0892  4              ELSE                                 ! if defined,
;  513          0893  4                  VFCHDR = 2;                      ! otherwise, set default value
;  514          0894  4
;  515          0895  3              END;
;  516          0896  3
;  517          0897  3      ! If this tape is being created for interchange do not allow VMS
;  518          0898  3      ! specific values in this field
;  519          0899  3
;  520          0900  3      IF .CURRENT_VCB[VCB$V_INTCHG] AND .HDR2[HD2$B_RECFORMAT] EQLU 'U'
;  521          0901  3      THEN
;  522          0902  3          ERR_EXIT(SS$_BADATTRIB);
;  523          0903  3
;  524          0904  2          END;
;  525          0905  2
;  526          0906  2      ! if form control character is present
;  527          0907  2
;  528          0908  3      IF .COUNT GEQU ($BYTEOFFSET(FAT$B_RATTRIB) + 1)
;  529          0909  2      THEN
;  530          0910  3          BEGIN
;  531          0911  3
;  532          0912  3          INCR I FROM 0 TO RECATTR_LEN - 1 DO
;  533          0913  3
;  534          0914  3              IF .RECATTR_FILE11[.I] EQLU .PTER[FAT$B_RATTRIB]
;  535          0915  3              THEN
;  536          0916  3                  HDR2[HD2$B_FORMCNTRL] = .RECATTR_ASCI[.I];
;  537          0917  3
;  538          0918  2          END;
;  539          0919  2
;  540          0920  2      ! copy record attributes into HDR3 in hexadecimal format...
;  541          0921  2
;  542          0922  2      PARTIAL_CNT = .COUNT - 1;
```

```
;  543          0923  2
;  544          0924  2      IF .PARTIAL_CNT GTR 28
;  545          0925  2      THEN
;  546          0926  2          PARTIAL_CNT = 28;
;  547          0927  2
;  548          0928  2      DESCR[0] = 8;
;  549          0929  2
;  550          0930  2      INCR I FROM 0 TO .PARTIAL_CNT BY 4 DO
;  551          0931  3          BEGIN
;  552          0932  3          DESCR[1] = HDR3[HD3$T_RECATR] + (.I*2);
;  553          0933  3          SYS$FAO(CVT8, 0, DESCR, .(ATTRIBUTES + .I));
;  554          0934  2          END;
;  555          0935  2
;  556          0936  2      ! calculate maximum record size
;  557          0937  2
;  558          0938  2      MRS = .(ATTRIBUTES + $BYTEOFFSET(FAT$W_RSIZE))<0, 16>;
;  559          0939  2
;  560          0940  2      IF .MRS EQL 0
;  561          0941  2      THEN
;  562          0942  2          MRS = .(ATTRIBUTES + $BYTEOFFSET(FAT$W_MAXREC))<0, 16>;
;  563          0943  2
;  564          0944  2      ! Copy attributes to HDR2 too, for temporary support
;  565          0945  2
;  566          0946  2 !     X = 1;
;  567          0947  2 !
;  568          0948  2 !     IF NOT .X
;  569          0949  2 !     THEN
;  570          0950  2 !        BEGIN
;  571          0951  2 !        PARTIAL_CNT = .COUNT;
;  572          0952  2 !
;  573          0953  2 !        IF .PARTIAL_CNT GTRU 20
;  574          0954  2 !        THEN
;  575          0955  2 !            PARTIAL_CNT = 20;
;  576          0956  2 !
;  577          0957  2 !        CH$COPY(.PARTIAL_CNT, .PTER, 0, HD2$S_RECATR1, HDR2[HD2$T_RECATR1]);
;  578          0958  2 !        PARTIAL_CNT = .COUNT - .PARTIAL_CNT;
;  579          0959  2 !        CH$COPY(.PARTIAL_CNT, .PTER + HD2$S_RECATR1, 0, HD2$S_RECATR2,
;  580          0960  2 !            HDR2[HD2$T_RECATR2]);
;  581          0961  2 !        END;
;  582          0962  2
;  583          0963  1      END;


                                53  44  44  44  56  44  46  55  00286 P.AAE:  .ASCII  \UFDVDDDS\                          ;
                                00  06  05  04  03  02  01  00  0028E P.AAF:  .BYTE   0, 1, 2, 3, 4, 5, 6, 0              ;
                                    20  41  4D  20  41  4D  00296 P.AAG:  .ASCII  \MA MA \                                ;
                                    0A  09  08  02  01  00  0029C P.AAH:  .BYTE   0, 1, 2, 8, 9, 10                       ;

                                                                  RECTYPE_ANSI=      P.AAE
                                                                  RECTYPE_FILE11=    P.AAF
                                                                  RECATTR_ASCI=      P.AAG
                                                                  RECATTR_FILE11=    P.AAH


                                                        00FC 00000 WRECATTR:
                                                                   .WORD   Save R2,R3,R4,R5,R6,R7                        ; 0809
```

```
                                57      0000'  CF   9E 00002          MOVAB   VFCHDR, R7                                      ;
                                5E             28   C2 00007          SUBL2   #40, SP                                         ;
                                56       04    AC   D0 0000A          MOVL    PTER, R6                                        ; 0866
                20              00       66    08   2C 0000E          MOVC5   COUNT, (R6), #0, #32, ATTRIBUTES                ;
                                         08    AE      00014                                                                  ;
                                         08    AC   D5 00016          TSTL    COUNT                                           ; 0870
                                               45   13 00019          BEQL    5$                                              ;
                                50     0000G   CF   D0 0001B          MOVL    HDR2, R0                                        ; 0873
                                51       04    A0   9E 00020          MOVAB   4(R0), R1                                       ;
                                61       55    8F   90 00024          MOVB    #85, (R1)                                       ;
                                50             07   D0 00028          MOVL    #7, I                                           ; 0875
                                66       BD AF40    91 0002B 1$:       CMPB    RECTYPE_FILE11[I], (R6)                         ; 0878
                                               05   12 00030          BNEQ    2$                                              ;
                                61       AE AF40    90 00032          MOVB    RECTYPE_ANSI[I], (R1)                           ; 0880
                                F1             50   F4 00037 2$:       SOBGEQ  I, 1$                                           ; 0875
                           56   8F             61   91 0003A          CMPB    (R1), #86                                       ; 0884
                                               13   12 0003E          BNEQ    4$                                              ;
                                61       44    8F   90 00040          MOVB    #68, (R1)                                       ; 0887
                                10       08    AC   D1 00044          CMPL    COUNT, #16                                      ; 0889
                                               06   1F 00048          BLSSU   3$                                              ;
                                67       0F    A6   9B 0004A          MOVZBW  15(R6), VFCHDR                                  ; 0891
                                               03   11 0004E          BRB     4$                                              ;
                                67             02   B0 00050 3$:       MOVW    #2, VFCHDR                                      ; 0893
                     08    2C   AB             04   E1 00053 4$:       BBC     #4, 44(CURRENT_VCB), 5$                         ; 0900
                           55   8F             61   91 00058          CMPB    (R1), #85                                       ;
                                               02   12 0005C          BNEQ    5$                                              ;
                                               34   BF 0005E          CHMU    #52                                             ; 0902
                                02       08    AC   D1 00060 5$:       CMPL    COUNT, #2                                       ; 0908
                                               1E   1F 00064          BLSSU   8$                                              ;
                                52       04    AC   D0 00066          MOVL    PTER, R2                                        ; 0914
                                               50   D4 0006A          CLRL    I                                               ;
                           01   A2         8A AF40    91 0006C 6$:       CMPB    RECATTR_FILE11[I], 1(R2)                        ;
                                               0C   12 00072          BNEQ    7$                                              ;
                                51     0000G   CF   D0 00074          MOVL    HDR2, R1                                        ; 0916
                           24   A1     FF76 CF40    90 00079          MOVB    RECATTR_ASCI[I], 36(R1)                         ;
                     E8            50             05   F3 00080 7$:       AOBLEQ  #5, I, 6$                                       ; 0914
                     53    08   AC               01   C3 00084 8$:       SUBL3   #1, COUNT, PARTIAL_CNT                          ; 0922
                                1C             53   D1 00089          CMPL    PARTIAL_CNT, #28                                ; 0924
                                               03   15 0008C          BLEQ    9$                                              ;
                                53             1C   D0 0008E          MOVL    #28, PARTIAL_CNT                                ; 0926
                                6E             08   D0 00091 9$:       MOVL    #8, DESCR                                       ; 0928
                                52             04   CE 00094          MNEGL   #4, I                                           ; 0930
                                               21   11 00097          BRB     11$                                             ;
                           04   AE     0000GDF42    3E 00099 10$:      MOVAW   @HDR3[I], DESCR+4                               ; 0932
                           04   AE             04   C0 000A0          ADDL2   #4, DESCR+4                                     ;
                                     08 AE42        9F 000A4          PUSHAB  ATTRIBUTES[I]                                   ; 0933
                                               9E   DD 000A8          PUSHL   @(SP)+                                          ;
                                     04    AE      9F 000AA          PUSHAB  DESCR                                           ;
                                               7E   D4 000AD          CLRL    -(SP)                                           ;
                                     FCBB      CF   9F 000AF          PUSHAB  CVT8                                            ;
                     000000000G      9F             04   FB 000B3          CALLS   #4, @#SYS$FAO                           ;
                FFD9 52              04             53   F1 000BA 11$:      ACBL    PARTIAL_CNT, #4, I, 10$                 ; 0930
                           FE   A7         0A    AE   B0 000C0          MOVW    ATTRIBUTES+2, MRS                               ; 0938
                                               05   12 000C5          BNEQ    12$                                             ; 0940
                           FE   A7         18    AE   B0 000C7          MOVW    ATTRIBUTES+16, MRS                              ; 0942
                                                    04 000CC 12$:      RET                                                     ; 0963
```

; Routine Size: 205 bytes,   Routine Base: $CODE$ + 02A2

```
;  585     0964  1 ROUTINE WBLOCKSZ (ADDR, LENGTH) :  COMMON_CALL NOVALUE =
;  586     0965  1
;  587     0966  1 !++
;  588     0967  1 !
;  589     0968  1 ! FUNCTIONAL DESCRIPTION:
;  590     0969  1 !       This routine writes the block size
;  591     0970  1 !
;  592     0971  1 ! CALLING SEQUENCE:
;  593     0972  1 !       WBLOCKSZ(ARG1,ARG2)
;  594     0973  1 !
;  595     0974  1 ! INPUT PARAMETERS:
;  596     0975  1 !       ARG1 - address of attribute
;  597     0976  1 !       ARG2 - length of attribute
;  598     0977  1 !
;  599     0978  1 ! IMPLICIT INPUTS:
;  600     0979  1 !       none
;  601     0980  1 !
;  602     0981  1 ! OUTPUT PARAMETERS:
;  603     0982  1 !       none
;  604     0983  1 !
;  605     0984  1 ! IMPLICIT OUTPUTS:
;  606     0985  1 !       block size written to HDR2
;  607     0986  1 !
;  608     0987  1 ! ROUTINE VALUE:
;  609     0988  1 !       none
;  610     0989  1 !
;  611     0990  1 ! SIDE EFFECTS:
;  612     0991  1 !       none
;  613     0992  1 !
;  614     0993  1 ! USER ERRORS:
;  615     0994  1 !       SS$_BADATTRIB - block size attribute neq to 2
;  616     0995  1 !       SS$_NOTINTBLSZ - block size on interchange tape not <=2048
;  617     0996  1 !--
;  618     0997  1
;  619     0998  2     BEGIN
;  620     0999  2
;  621     1000  2     EXTERNAL REGISTER
;  622     1001  2         COMMON_REG;                             !  Set up reference to the VCB
;  623     1002  2
;  624     1003  2     MAP
;  625     1004  2         ADDR    : REF VECTOR [, BYTE];          ! address of attribute
;  626     1005  2
;  627     1006  2     LOCAL
;  628     1007  2         DESCR   : VECTOR [2];
;  629     1008  2
;  630     1009  2     BLOCKSZ = .(.ADDR)<0, 16>;
;  631     1010  2
;  632     1011  2     ! If this tape is being created for interchange do not allow VMS
;  633     1012  2     ! specific values in this field
;  634     1013  2
;  635     1014  2     IF .CURRENT_VCB[VCB$V_INTCHG]
;  636     1015  2     THEN
;  637     1016  2         IF .BLOCKSZ GTRU 2048
;  638     1017  2         THEN
;  639     1018  2             ERR_EXIT (SS$_NOTINTBLSZ);
;  640     1019  2
;  641     1020  2     DESCR[0] = HD2$S_BLOCKLEN;
```

```
;   642          1021  2     DESCR[1] = HDR2[HD2$T_BLOCKLEN];
;   643          1022  2     SYS$FAO(CVT5, 0, DESCR, .BLOCKSZ);
;   644          1023  1     END;


                                                  0004 00000 WBLOCKSZ:
                                                                      .WORD   Save R2                          ; 0964
                          52      0000'   CF   9E 00002               MOVAB   BLOCKSZ, R2                      ;
                          5E              08   C2 00007               SUBL2   #8, SP                           ;
                          62        04    BC   3C 0000A               MOVZWL  @ADDR, BLOCKSZ                   ; 1009
          0D          2C  AB              04   E1 0000E               BBC     #4, 44(CURRENT_VCB), 1$          ; 1014
               00000800   8F              62   D1 00013               CMPL    BLOCKSZ, #2048                   ; 1016
                                          04   1B 0001A               BLEQU   1$                               ;
                                  01D4    8F   BF 0001C               CHMU    #468                             ; 1018
                          6E              05   D0 00020 1$:           MOVL    #5, DESCR                        ; 1020
      04      AE   0000G  CF              05   C1 00023               ADDL3   #5, HDR2, DESCR+4                ; 1021
                                          62   DD 0002A               PUSHL   BLOCKSZ                          ; 1022
                                    04    AE   9F 0002C               PUSHAB  DESCR                            ;
                                          7E   D4 0002F               CLRL    -(SP)                            ;
                                  FC60    CF   9F 00031               PUSHAB  CVT5                             ;
               00000000G  9F              04   FB 00035               CALLS   #4, @#SYS$FAO                    ;
                                               04 0003C               RET                                      ; 1023

; Routine Size:  61 bytes,    Routine Base:  $CODE$ + 036F


;  645          1024  1
```

```
;  647     1025  1 ROUTINE WBINDATE (INPUT, INPLEN, OUTPUT) :  NOVALUE =
;  648     1026  1
;  649     1027  1 !++
;  650     1028  1 !
;  651     1029  1 ! FUNCTIONAL DESCRIPTION:
;  652     1030  1 !       This routine converts a 64 bit input date to an ANSI Julian date
;  653     1031  1 !
;  654     1032  1 ! CALLING SEQUENCE:
;  655     1033  1 !       WBINDATE(ARG1,ARG2,ARG3)
;  656     1034  1 !
;  657     1035  1 ! INPUT PARAMETERS:
;  658     1036  1 !       ARG1 - address of 64 bit date
;  659     1037  1 !       ARG2 - length of date
;  660     1038  1 !       ARG3 - address for Julian date to placed
;  661     1039  1 !
;  662     1040  1 ! IMPLICIT INPUTS:
;  663     1041  1 !       none
;  664     1042  1 !
;  665     1043  1 ! OUTPUT PARAMETERS:
;  666     1044  1 !       ARG3 - address which receives Julian date
;  667     1045  1 !
;  668     1046  1 ! IMPLICIT OUTPUTS:
;  669     1047  1 !       none
;  670     1048  1 !
;  671     1049  1 ! ROUTINE VALUE:
;  672     1050  1 !       none
;  673     1051  1 !
;  674     1052  1 ! SIDE EFFECTS:
;  675     1053  1 !       none
;  676     1054  1 !
;  677     1055  1 ! USER ERRORS:
;  678     1056  1 !       SS$_BADATTRIB - attribute length neq 8
;  679     1057  1 !
;  680     1058  1 !--
;  681     1059  1
;  682     1060  2     BEGIN
;  683     1061  2
;  684     1062  2     LOCAL
;  685     1063  2         REGDATE : VECTOR [12, BYTE],
;  686     1064  2         REGDESC : VECTOR [2];
;  687     1065  2
;  688     1066  2     EXTERNAL ROUTINE
;  689     1067  2         SYS$ASCTIM      : ADDRESSING_MODE (ABSOLUTE),
;  690     1068  2         CONVDATE_R2J;
;  691     1069  2
;  692     1070  2     REGDESC[0] = 12;
;  693     1071  2     REGDESC[1] = REGDATE;
;  694     1072  2     SYS$ASCTIM(0, REGDESC, .INPUT, 0);
;  695     1073  2     CONVDATE_R2J(REGDATE, .OUTPUT);
;  696     1074  1     END;


                                                        .EXTRN  SYS$ASCTIM

                                0000 00000 WBINDATE:
                                                        .WORD   Save nothing                    ; 1025
```

```
                         5E              10  C2 00002        SUBL2   #16, SP                                 ;
                                         0C  DD 00005        PUSHL   #12                                      ; 1070
                      04 AE          08  AE  9E 00007        MOVAB   REGDATE, REGDESC+4                       ; 1071
                                         7E  D4 0000C        CLRL    -(SP)                                    ; 1072
                                     04  AC  DD 0000E        PUSHL   INPUT                                    ;
                                     08  AE  9F 00011        PUSHAB  REGDESC                                  ;
                                         7E  D4 00014        CLRL    -(SP)                                    ;
              00000000G  9F              04  FB 00016        CALLS   #4, @#SYS$ASCTIM                         ;
                                     0C  AC  DD 0001D        PUSHL   OUTPUT                                   ; 1073
                                     0C  AE  9F 00020        PUSHAB  REGDATE                                  ;
                   0000G  CF             02  FB 00023        CALLS   #2, CONVDATE_R2J                         ;
                                             04 00028        RET                                              ; 1074

; Routine Size:  41 bytes,    Routine Base:  $CODE$ + 03AC
```

```
;  698          1075  1 ROUTINE WANSINAME ( POINTER, LENGTH ) :  COMMON_CALL NOVALUE =
;  699          1076  1
;  700          1077  1 !++
;  701          1078  1 !
;  702          1079  1 ! FUNCTIONAL DESCRIPTION:
;  703          1080  1 !       Writes the 17 or less character filename containing ANSI "a" characters
;  704          1081  1 !       into the HDR1. The remainder of a VMS long filename is written into
;  705          1082  1 !       the HDR4 label.
;  706          1083  1 !
;  707          1084  1 ! CALLING SEQUENCE:
;  708          1085  1 !       WANSINAME ( pointer, length )
;  709          1086  1 !
;  710          1087  1 ! INPUT PARAMETERS:
;  711          1088  1 !       pointer - points to file name string
;  712          1089  1 !       length  - length of the file name string
;  713          1090  1 !
;  714          1091  1 ! IMPLICIT INPUTS:
;  715          1092  1 !       none
;  716          1093  1 !
;  717          1094  1 ! OUTPUT PARAMETERS:
;  718          1095  1 !       none
;  719          1096  1 !
;  720          1097  1 ! IMPLICIT OUTPUTS:
;  721          1098  1 !       file name is stuffed into the header
;  722          1099  1 !
;  723          1100  1 ! ROUTINE VALUE:
;  724          1101  1 !       none
;  725          1102  1 !
;  726          1103  1 ! SIDE EFFECTS:
;  727          1104  1 !       none
;  728          1105  1 !
;  729          1106  1 ! USER ERRORS:
;  730          1107  1 !
;  731          1108  1 !--
;  732          1109  1
;  733          1110  2     BEGIN
;  734          1111  2
;  735          1112  2     EXTERNAL REGISTER
;  736          1113  2         COMMON_REG;
;  737          1114  2
;  738          1115  2     EXTERNAL ANSI_A_BAD          : VECTOR [ , BYTE ],
;  739          1116  2              ESC_CHAR            : BYTE;
;  740          1117  2
;  741          1118  2     LOCAL
;  742          1119  2         TEMP_ID                  : VECTOR [ FILE_SPEC_MAX,BYTE ],
;  743          1120  2         RETURN_LENGTH;
;  744          1121  2
;  745          1122  2     ! check length of file name
;  746          1123  2
;  747          1124  2     IF .LENGTH GTRU FILE_SPEC_MAX THEN ERR_EXIT ( SS$_BADFILENAME );
;  748          1125  2
;  749          1126  2     ! Space fill the temporary storage for the FILE_ID
;  750          1127  2
;  751          1128  2     CH$FILL(' ',FILE_SPEC_MAX,TEMP_ID);
;  752          1129  2
;  753          1130  2     ! translate into upper case and check for non-ANSI 'a' characters
;  754          1131  2
```

```
;  755          1132  2     IF 0 NEQ MOVTUC ( LENGTH,                  .POINTER, ESC_CHAR, ANSI_A_BAD,
;  756          1133  2                       %REF ( FILE_SPEC_MAX ), TEMP_ID)
;  757          1134  2     THEN ERR_EXIT ( SS$_BADFILENAME );
;  758          1135  2
;  759          1136  2     ! Move the filename into the headers and padd space left with spaces
;  760          1137  2
;  761          1138  2     CH$MOVE (HD1$S_FILEID, TEMP_ID, HDR1[HD1$T_FILEID]);
;  762          1139  2     CH$MOVE (HD4$S_FILEID_EXT, TEMP_ID[HD1$S_FILEID], HDR4[HD4$T_FILEID_EXT]);
;  763          1140  2
;  764          1141  2     ! Check the length of the file name. If the file name will fit in
;  765          1142  2     ! the HDR1 FILE ID then set the HDR4 length to zero. Else set up
;  766          1143  2     ! the lengths such that the HDR1 FILE ID is filed with the name
;  767          1144  2     ! then the remainder of the name is put in the HDR4 label with the
;  768          1145  2     ! size that is in the HDR4 label only.
;  769          1146  2     ! PLEASE NOTE that the actual implementation of this is different for
;  770          1147  2     ! volumes with a 4 in the VOL1 standard field as opposed to a 3 or less.
;  771          1148  2     ! This is because the new standard allows us to write any kind
;  772          1149  2     ! of data in implementation dependant fields. The old standard did not allow
;  773          1150  2     ! us to do this.
;  774          1151  2
;  775          1152  3     BEGIN
;  776          1153  3         BIND
;  777          1154  3             CVT2 = DESCRIPTOR ('!2ZW');
;  778          1155  3         LOCAL
;  779          1156  3             DESCR : VECTOR [2],
;  780          1157  3             MVL   : REF BBLOCK;
;  781          1158  3         MVL = .CURRENT_VCB[VCB$L_MVL];
;  782          1159  3         IF .LENGTH  LEQU HD1$S_FILEID
;  783          1160  3            THEN
;  784          1161  4            BEGIN
;  785          1162  4            IF .MVL[MVL$B_STDVER] GTR 3
;  786          1163  4                 THEN
;  787          1164  4                     HDR4[HD4$B_FILEID_EXT_SIZE] = 0
;  788          1165  4                 ELSE
;  789          1166  4                     CH$FILL(0,HD4$S_FILEID_EXT_V3,HDR4[HD4$T_FILEID_EXT_V3]);
;  790          1167  4            END
;  791          1168  3            ELSE
;  792          1169  4            BEGIN
;  793          1170  4            IF .MVL[MVL$B_STDVER] GTR 3
;  794          1171  4                 THEN
;  795          1172  4                     HDR4[HD4$B_FILEID_EXT_SIZE] = .LENGTH - HD1$S_FILEID
;  796          1173  4                 ELSE
;  797          1174  5                 BEGIN
;  798          1175  5                    LOCAL LEN;
;  799          1176  5                    LEN = .LENGTH - HD1$S_FILEID;
;  800          1177  5                    DESCR[0] = HD4$S_FILEID_EXT_V3;
;  801          1178  5                    DESCR[1] = HDR4[HD4$T_FILEID_EXT_V3];
;  802          1179  5                    $FAO(CVT2,0,DESCR,.LEN);
;  803          1180  4                 END;
;  804          1181  4            END
;  805          1182  2         END;
;  806          1183  2
;  807          1184  2     ! strip trailing blanks from the string
;  808          1185  2
;  809          1186  2     RETURN_LENGTH = 0;
;  810          1187  2     DECR I FROM (.LENGTH - 1) TO 0 DO
;  811          1188  2         IF TEMP_ID [ .I ] NEQ ' '
```

```
;     2          THEN
;     3              BEGIN
;     3              RETURN_LENGTH = .I + 1;
;     3              EXITLOOP
;     2              END;
;     2
;     2      ! tell the other routines the size of this name
;     2      !
;     2      ANSI_NAME_SIZE = .RETURN_LENGTH;
;     2
;     1      END;                              ! end of Routine


                               57  5A  32  21  003D5 P.AAJ:  .ASCII  \!2ZW\                              ;
                                               003D9         .BLKB   3
                                   00000004    003DC P.AAI:  .LONG   4                                   ;
                                   00000000'   003E0         .ADDRESS P.AAJ                              ;

                                                     CVT2=            P.AAI
                                                             .EXTRN  ANSI_A_BAD, ESC_CHAR

                                         007C  00000 WANSINAME:
                                                             .WORD   Save R2,R3,R4,R5,R6                 ; 1075
                          5E       A8   AE  9E 00002         MOVAB   -88(SP), SP                         ;
                0000004F  8F       08   AC  D1 00006         CMPL    LENGTH, #79                         ; 1124
                                        04  1B 0000E         BLEQU   1$                                  ;
                               0818     8F  BF 00010         CHMU    #2072                               ;
 004F  8F         20      6E            00  2C 00014 1$:     MOVC5   #0, (SP), #32, #79, TEMP_ID         ; 1128
                                 08     AE     0001B                                                     ;
 0000G CF  0000G  CF  04  BC     08     AC  2F 0001D         MOVTUC  LENGTH, @POINTER, ESC_CHAR, ANSI_A_BAD, - ; 1132
                      08  AE   004F     8F     00028                 #79, TEMP_ID                        ;
                                        02  1D 0002D         BVS     2$                                  ;
                                        51  D4 0002F         CLRL    R1                                  ;
                                        51  D5 00031 2$:     TSTL    R1                                  ;
                                        04  13 00033         BEQL    3$                                  ;
                               0818     8F  BF 00035         CHMU    #2072                               ; 1134
                          50  0000G     CF  D0 00039 3$:     MOVL    HDR1, R0                            ; 1138
          04  A0      08  AE            11  28 0003E         MOVC3   #17, TEMP_ID, 4(R0)                 ;
                          56  0000G     CF  D0 00044         MOVL    HDR4, R6                            ; 1139
          05  A6      19  AE            3E  28 00049         MOVC3   #62, TEMP_ID+17, 5(R6)              ;
                          50     34     AB  D0 0004F         MOVL    52(CURRENT_VCB), MVL                ; 1158
                          11     08     AC  D1 00053         CMPL    LENGTH, #17                         ; 1159
                                        10  1A 00057         BGTRU   5$                                  ;
                          03     22     A0  91 00059         CMPB    34(MVL), #3                         ; 1162
                                        05  1B 0005D         BLEQU   4$                                  ;
                                 04     A6  94 0005F         CLRB    4(R6)                               ; 1164
                                        33  11 00062         BRB     7$                                  ;
                                 43     A6  B4 00064 4$:     CLRW    67(R6)                              ; 1166
                                        2E  11 00067         BRB     7$                                  ; 1159
              51      08  AC            11  C3 00069 5$:     SUBL3   #17, LENGTH, R1                     ; 1172
                          03     22     A0  91 0006E         CMPB    34(MVL), #3                         ; 1170
                                        06  1B 00072         BLEQU   6$                                  ;
                      04  A6            51  90 00074         MOVB    R1, 4(R6)                           ; 1172
                                        1D  11 00078         BRB     7$                                  ;
                          50            51  D0 0007A 6$:     MOVL    R1, LEN                             ; 1176
                          6E            02  D0 0007D         MOVL    #2, DESCR                           ; 1177
```

```
                    04   AE      43   A6  9E 00080          MOVAB   67(R6), DESCR+4                 ; 1178
                                     50  DD 00085          PUSHL   LEN                             ; 1179
                                 04  AE  9F 00087          PUSHAB  DESCR                           ;
                                     7E  D4 0008A          CLRL    -(SP)                           ;
                               FF68  CF  9F 0008C          PUSHAB  CVT2                            ;
                00000000G      00       04  FB 00090      CALLS   #4, SYS$FAO                     ;
                                        51  D4 00097 7$:   CLRL    RETURN_LENGTH                   ; 1186
                               50       08  AC  D0 00099   MOVL    LENGTH, I                       ; 1187
                                        10  11 0009D       BRB     9$                              ;
                               52       08 AE40  9E 0009F 8$:   MOVAB   TEMP_ID[I], R2             ; 1188
                               20           52  D1 000A4   CMPL    R2, #32                         ;
                                        06  13 000A7       BEQL    9$                              ;
                               51       01  A0  9E 000A9   MOVAB   1(R0), RETURN_LENGTH           ; 1191
                                        03  11 000AD       BRB     10$                             ; 1190
                               ED           50  F4 000AF 9$:   SOBGEQ  I, 8$                      ; 1188
                          0000'  CF             51  90 000B2 10$:  MOVB    RETURN_LENGTH, ANSI_NAME_SIZE ; 1197
                                                04 000B7       RET                                 ; 1199

; Routine Size:  184 bytes,    Routine Base:  $CODE$ + 03E4
```

```
;  824     1200  1 !++++
;  825     1201  1 !ROUTINE WLABELAST (ATTRIBUTE) :  NOVALUE =
;  826     1202  1 !
;  827     1203  1 !++
;  828     1204  1 !
;  829     1205  1 ! FUNCTIONAL DESCRIPTION:
;  830     1206  1 !       This routine validates and stores the user's supplied AST control block
;  831     1207  1 !
;  832     1208  1 ! CALLING SEQUENCE:
;  833     1209  1 !       WLABELAST(ARG1)
;  834     1210  1 !
;  835     1211  1 ! INPUT PARAMETERS:
;  836     1212  1 !       ARG1    - address of attribute
;  837     1213  1 !
;  838     1214  1 ! IMPLICIT INPUTS:
;  839     1215  1 !       none
;  840     1216  1 !
;  841     1217  1 ! OUTPUT PARAMETERS:
;  842     1218  1 !       none
;  843     1219  1 !
;  844     1220  1 ! IMPLICIT OUTPUTS:
;  845     1221  1 !       none
;  846     1222  1 !
;  847     1223  1 ! ROUTINE VALUE:
;  848     1224  1 !       none
;  849     1225  1 !
;  850     1226  1 ! SIDE EFFECTS:
;  851     1227  1 !       none
;  852     1228  1 !
;  853     1229  1 ! USER ERRORS:
;  854     1230  1 !       none
;  855     1231  1 !
;  856     1232  1 !--
;  857     1233  1 !
;  858     1234  1 !   BEGIN
;  859     1235  1 !
;  860     1236  1 !   EXTERNAL REGISTER
;  861     1237  1 !      COMMON_REG;
;  862     1238  1 !
;  863     1239  1 !   LOCAL
;  864     1240  1 !      AST              : REF BBLOCK,          ! user supplied AST control block
;  865     1241  1 !      FUNCTION         : BLOCK [1],           ! IO function code and modifiers
;  866     1242  1 !      LENGTH,
;  867     1243  1 !      MODE;
;  868     1244  1 !
;  869     1245  1 !   EXTERNAL
;  870     1246  1 !      IO_PACKET        : REF BBLOCK;          ! address of IO request packet
;  871     1247  1 !
;  872     1248  1 !   FUNCTION = .IO_PACKET[IRP$W_FUNC];
;  873     1249  1 !   AST = ..ATTRIBUTE;                  ! parameter is address of attribute
;  874     1250  1 !   MODE = 0;
;  875     1251  1 !   LENGTH = 4;
;  876     1252  1 !   BEGIN
;  877     1253  1 !
;  878     1254  1 !   BUILTIN
;  879     1255  1 !      PROBER;
;  880     1256  1 !
```

```
;  881          1257  1 !     IF  .AST NEQ 0
;  882          1258  1 !        AND
;  883          1259  1 !        ( NOT PROBER(MODE, _ENGTH, .AST)
;  884          1260  1 !        OR
;  885          1261  1 !        .AST[ACB$B_TYPE] NEQ DYN$C_ACB
;  886          1262  1 !        OR
;  887          1263  1 !        NOT .FUNCTION[IO$V_ACCESS])
;  888          1264  1 !     THEN
;  889          1265  1 !        ERR_EXIT(SS$_ILLLBLAST);
;  890          1266  1 !
;  891          1267  1 !     END;
;  892          1268  1
;  893          1269  1      ! won't return any AST other than the currently recorded one
;  894          1270  1      !
;  895          1271  1 !     KERNEL_CALL(COMPLETE_USRLBL, .AST, 0);
;  896          1272  1 !     END;
;  897          1273  1 !----
;  898          1274  1 END
;  899          1275  1
;  900          1276  0 ELUDOM



;                                  PSECT SUMMARY
;
;       Name                     Bytes                          Attributes
;
;  $LOCKEDD1$                       9  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        1180  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        64         0      1000          00:01.9



;                                    COMMAND QUALIFIERS

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

; Size:         1113 code + 76 data bytes
; Run Time:       00:23.7
; Elapsed Time:   00:50.4
; Lines/CPU Min:   3235
```

```
; Lexemes/CPU-Min: 17863
; Memory Used:  187 pages
; Compilation Complete
```

0256 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

WTATTR
LIS

POSFID
LIS

RESACC
LIS

RWVOL
LIS

MTHRTL

STRVIO
LIS

MTHRTL
MAP

POSBE
LIS

RDATTR
LIS

REWSPC
LIS

UUMTHRTL
MAP