```
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPP          PPP   AAA         AAA  SSS          RRR         RRR        TTT        LLL
PPP          PPP   AAA         AAA  SSS          RRR         RRR        TTT        LLL
PPP          PPP   AAA         AAA  SSS          RRR         RRR        TTT        LLL
PPP          PPP   AAA         AAA  SSS          RRR         RRR        TTT        LLL
PPP          PPP   AAA         AAA  SSS          RRR         RRR        TTT        LLL
PPP          PPP   AAA         AAA  SSS          RRR         RRR        TTT        LLL
PPPPPPPPPPPP       AAA         AAA     SSSSSSSSS     RRRRRRRRRRR        TTT        LLL
PPPPPPPPPPPP       AAA         AAA     SSSSSSSSS     RRRRRRRRRRR        TTT        LLL
PPPPPPPPPPPP       AAA         AAA     SSSSSSSSS     RRRRRRRRRRR        TTT        LLL
PPP                AAAAAAAAAAAAAAA           SSS   RRR   RRR          TTT        LLL
PPP                AAAAAAAAAAAAAAA           SSS   RRR   RRR          TTT        LLL
PPP                AAAAAAAAAAAAAAA           SSS   RRR   RRR          TTT        LLL
PPP                AAA         AAA           SSS   RRR      RRR       TTT        LLL
PPP                AAA         AAA           SSS   RRR      RRR       TTT        LLL
PPP                AAA         AAA           SSS   RRR      RRR       TTT        LLL
PPP                AAA         AAA  SSSSSSSSSSS   RRR         RRR     TTT        LLLLLLLLLLLLLLL
PPP                AAA         AAA  SSSSSSSSSSS   RRR         RRR     TTT        LLLLLLLLLLLLLLL
PPP                AAA         AAA  SSSSSSSSSSS   RRR         RRR     TTT        LLLLLLLLLLLLLLL
PPP                AAA         AAA  SSSSSSSSSSS   RRR         RRR     TTT        LLLLLLLLLLLLLLL
```

FILEID**PASWRIUNS

```
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR     IIIIII    UU      UU  NN      NN    SSSSSSSS
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR     IIIIII    UU      UU  NN      NN    SSSSSSSS
PP      PP   AA      AA  SS        WW      WW  RR      RR     II      UU      UU  NN      NN  SS
PP      PP   AA      AA  SS        WW      WW  RR      RR     II      UU      UU  NN      NN  SS
PP      PP   AA      AA  SS        WW      WW  RR      RR     II      UU      UU  NNNN    NN  SS
PP      PP   AA      AA  SS        WW      WW  RR      RR     II      UU      UU  NNNN    NN  SS
PPPPPPPP     AA      AA    SSSSSS  WW      WW  RRRRRRRR       II      UU      UU  NN  NN  NN    SSSSSS
PPPPPPPP     AA      AA    SSSSSS  WW      WW  RRRRRRRR       II      UU      UU  NN  NN  NN    SSSSSS
PP           AAAAAAAAAA          SS  WW  WW  WW  RR  RR       II      UU      UU  NN    NNNN          SS
PP           AAAAAAAAAA          SS  WW  WW  WW  RR  RR       II      UU      UU  NN    NNNN          SS
PP           AA      AA          SS  WWWW  WWWW  RR    RR     II      UU      UU  NN      NN          SS      ....
PP           AA      AA          SS  WWWW  WWWW  RR    RR     II      UU      UU  NN      NN          SS      ....
PP           AA      AA  SSSSSSSS    WW      WW  RR      RR   IIIIII    UUUUUUUUUU  NN      NN  SSSSSSSS    ....
PP           AA      AA  SSSSSSSS    WW      WW  RR      RR   IIIIII    UUUUUUUUUU  NN      NN  SSSSSSSS    ....

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 MODULE PAS$WRITE_UNSIGNED ( %TITLE 'Write an unsigned integer'
;     2          0002  0                 IDENT = '1-002'                 ! File: PASWRIUNS.B32 Edit: SBL1002
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**********************************************************************
;     7          0007  1 !*                                                                    *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                           *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.            *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                             *
;    11          0011  1 !*                                                                    *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  1 !*  TRANSFERRED.                                                       *
;    18          0018  1 !*                                                                    *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  1 !*  CORPORATION.                                                       *
;    22          0022  1 !*                                                                    *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
;    25          0025  1 !*                                                                    *
;    26          0026  1 !*                                                                    *
;    27          0027  1 !**********************************************************************
;    28          0028  1 !
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 ! FACILITY:     Pascal Language Support
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module contains a procedure which writes an unsigned integer
;    36          0036  1 !       to a textfile.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  User mode - AST reentrant
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.  SBL 1-April-1981
;    45          0045  1 ! 1-002 - Make total-width a longword.  SBL 30-June-1982
;    46          0046  1 !--
;    47          0047  1
```

```
;    49          0048  1 %SBTTL 'Declarations'
;    50          0049  1 !
;    51          0050  1 ! PROLOGUE DEFINITIONS:
;    52          0051  1 !
;    53          0052  1
;    54          0053  1 REQUIRE 'RTLIN:PASPROLOG';                    ! Externals, linkages, PSECTs, structures
;    55          0117  1
;    56          0118  1 !
;    57          0119  1 ! TABLE OF CONTENTS:
;    58          0120  1 !
;    59          0121  1
;    60          0122  1 FORWARD ROUTINE
;    61          0123  1     PASSWRITE_UNSIGNED: NOVALUE,               ! Write to textfile
;    62          0124  1     PASSWRITEV_UNSIGNED: NOVALUE;              ! Write to string
;    63          0125  1
;    64          0126  1 !
;    65          0127  1 ! MACROS:
;    66          0128  1 !
;    67          0129  1 !       NONE
;    68          0130  1 !
;    69          0131  1 ! EQUATED SYMBOLS:
;    70          0132  1 !
;    71          0133  1 !       NONE
;    72          0134  1 !
;    73          0135  1 ! FIELDS:
;    74          0136  1 !
;    75          0137  1 !       NONE
;    76          0138  1 !
;    77          0139  1 ! OWN STORAGE:
;    78          0140  1 !
;    79          0141  1 !       NONE
;    80          0142  1 !
```

```
;    82          0143  1 %SBTTL 'PAS$WRITE_UNSIGNED - Write unsigned integer to textfile'
;    83          0144  1 GLOBAL ROUTINE PAS$WRITE_UNSIGNED (
;    84          0145  1         PFV: REF $PAS$PFV_FILE_VARIABLE,                ! File variable
;    85          0146  1         INTEGER,                                        ! Value to write
;    86          0147  1         TOTAL_WIDTH: SIGNED,                            ! Total field width
;    87          0148  1         ERROR                                           ! Error unwind address
;    88          0149  1     ): NOVALUE =
;    89          0150  1
;    90          0151  1 !++
;    91          0152  1 ! FUNCTIONAL DESCRIPTION:
;    92          0153  1 !
;    93          0154  1 !       This procedure writes an unsigned integer to the specified textfile.
;    94          0155  1 !
;    95          0156  1 ! CALLING SEQUENCE:
;    96          0157  1 !
;    97          0158  1 !       CALL PAS$WRITE_UNSIGNED (PFV.mr.r, INTEGER.rlu.v, TOTAL_WIDTH.rl.v
;    98          0159  1 !                                [,ERROR.j.r])
;    99          0160  1 !
;   100          0161  1 ! FORMAL PARAMETERS:
;   101          0162  1 !
;   102          0163  1 !       PFV             - The Pascal File Variable (PFV) passed by reference.
;   103          0164  1 !                         The structure of the PFV is defined in PASPFV.REQ.
;   104          0165  1 !
;   105          0166  1 !       INTEGER         - The integer to write.
;   106          0167  1 !
;   107          0168  1 !       TOTAL_WIDTH     - Optional.  Total field width.
;   108          0169  1 !
;   109          0170  1 !       ERROR           - Optional.  Address to unwind to if an error occurs.
;   110          0171  1 !
;   111          0172  1 ! IMPLICIT INPUTS:
;   112          0173  1 !
;   113          0174  1 !       NONE
;   114          0175  1 !
;   115          0176  1 ! IMPLICIT OUTPUTS:
;   116          0177  1 !
;   117          0178  1 !       NONE
;   118          0179  1 !
;   119          0180  1 ! ROUTINE VALUE:
;   120          0181  1 !
;   121          0182  1 !       NONE
;   122          0183  1 !
;   123          0184  1 ! SIDE EFFECTS:
;   124          0185  1 !
;   125          0186  1 !       If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
;   126          0187  1 !
;   127          0188  1 ! SIGNALLED ERRORS:
;   128          0189  1 !
;   129          0190  1 !       LINTOOLON - line too long
;   130          0191  1 !       NEGWIDDIG - negative width or digits specification not allowed
;   131          0192  1 !
;   132          0193  1 !--
;   133          0194  1
;   134          0195  2     BEGIN
;   135          0196  2
;   136          0197  2     LOCAL
;   137          0198  2         FCB: REF $PAS$FCB_CONTROL_BLOCK,                ! File control block
;   138          0199  2         FIELD_WIDTH,                                    ! Total width
```

```
;  139          0200  2          INT_DIGITS,                             ! Number of digits
;  140          0201  2          STRING: VECTOR [12, BYTE],              ! String for result
;  141          0202  2          DESCR: BLOCK [8, BYTE],                 ! String descriptor
;  142          0203  2          CTRSTR_DESCR: BLOCK [8, BYTE],          ! FAO Control string descriptor
;  143          0204  2          PFV_ADDR: VOLATILE,                     ! Enable argument
;  144          0205  2          UNWIND_ACT: VOLATILE,                   ! Enable argument
;  145          0206  2          ERROR_ADDR: VOLATILE;                   ! Enable argument
;  146          0207  2
;  147          0208  2      BUILTIN
;  148          0209  2          ACTUALCOUNT;
;  149          0210  2
;  150          0211  2      ENABLE
;  151          0212  2          PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  152          0213  2
;  153          0214  2      !+
;  154          0215  2      ! Get ERROR parameter, if present.
;  155          0216  2      !-
;  156          0217  2
;  157          0218  2      IF ACTUALCOUNT () GEQU 4
;  158          0219  2      THEN
;  159          0220  2          ERROR_ADDR = .ERROR;            ! Set unwind address
;  160          0221  2
;  161          0222  2      PFV_ADDR = PFV [PFV$R_PFV];         ! Set PFV address
;  162          0223  2
;  163          0224  2      !+
;  164          0225  2      ! Validate PFV and get PFV.
;  165          0226  2      !-
;  166          0227  2
;  167          0228  2      PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
;  168          0229  2
;  169          0230  2      !+
;  170          0231  2      ! Set unwind action to unlock file.
;  171          0232  2      !-
;  172          0233  2
;  173          0234  2      UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
;  174          0235  2
;  175          0236  2      !+
;  176          0237  2      ! Do common initialization.
;  177          0238  2      !-
;  178          0239  2
;  179          0240  2      PAS$$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
;  180          0241  2
;  181          0242  2      !+
;  182          0243  2      ! Check for invalid width.
;  183          0244  2      !-
;  184          0245  2
;  185          0246  2      IF .TOTAL_WIDTH LSS 0
;  186          0247  2      THEN
;  187          0248  2          $PAS$IO_ERROR (PAS$_NEGWIDDIG,0);
;  188          0249  2
;  189          0250  2      !+
;  190          0251  2      ! Convert integer to text
;  191          0252  2      !-
;  192          0253  2
;  193          0254  2      DESCR [DSC$W_LENGTH] = 12;
;  194          0255  2      DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
;  195          0256  2      DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
```

```
;  196          0257  2     DESCR [DSC$A_POINTER] = STRING;
;  197          0258  2     CTRSTR_DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
;  198          0259  2     CTRSTR_DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  199          0260  2     CTRSTR_DESCR [DSC$W_LENGTH] = %CHARCOUNT ('!UL');
;  200          0261  2     CTRSTR_DESCR [DSC$A_POINTER] = UPLIT BYTE ('!UL');
;  201          0262  2
;  202        P 0263  2     $FAO (CTRSTR_DESCR,          ! Control string descriptor
;  203        P 0264  2           DESCR [DSC$W_LENGTH], ! Returned length
;  204        P 0265  2           DESCR,                ! Result descriptor
;  205          0266  2           .INTEGER);            ! Value to convert (Conversion can't fail)
;  206          0267  2
;  207          0268  2
;  208          0269  2     !+
;  209          0270  2     ! Get desired field width and width of converted string.
;  210          0271  2     !-
;  211          0272  2
;  212          0273  2     FIELD_WIDTH = .TOTAL_WIDTH;
;  213          0274  2     INT_DIGITS = .DESCR [DSC$W_LENGTH];
;  214          0275  2
;  215          0276  2     !+
;  216          0277  2     ! See if field will fit in record.
;  217          0278  2     !-
;  218          0279  2
;  219          0280  2     FIELD_WIDTH = MAX (.FIELD_WIDTH, .DESCR [DSC$W_LENGTH]);
;  220          0281  3     BEGIN
;  221          0282  3     LOCAL
;  222          0283  3         EXTRA;                   ! Extra characters past end of line
;  223          0284  3     EXTRA = (.FCB [FCB$A_RECORD_CUR] + .FIELD_WIDTH) - .FCB [FCB$A_RECORD_END];
;  224          0285  3     IF .EXTRA GTR 0
;  225          0286  3     THEN
;  226          0287  3         $PAS$IO_ERROR (PAS$_LINTOOLON,1,.EXTRA);
;  227          0288  2     END;
;  228          0289  2
;  229          0290  2     !+
;  230          0291  2     ! Move leading blanks, if any.
;  231          0292  2     !-
;  232          0293  2
;  233          0294  2     IF     D_WIDTH - .DESCR [DSC$W_LENGTH] GTR 0
;  234          0295  2     THEN
;  235          0296  2         FCB [FCB$A_RECORD_CUR] = CH$FILL (%C' ', .FIELD_WIDTH - .DESCR [DSC$W_LENGTH], .FCB [FCB$A_RECORD_CU
;  236          0297  2
;  237          0298  2     !+
;  238          0299  2     ! Now move value
;  239          0300  2     !-
;  240          0301  2
;  241          0302  2     FCB [FCB$A_RECORD_CUR] = CH$MOVE (.DESCR [DSC$W_LENGTH], .DESCR [DSC$A_POINTER], .FCB [FCB$A_RECORD_CUR]
;  242          0303  2
;  243          0304  2     !+
;  244          0305  2     ! Call WRITE epilogue routine to move the last character written to the
;  245          0306  2     ! user's buffer and to unlock the file variable.
;  246          0307  2     !-
;  247          0308  2
;  248          0309  2     PAS$$END_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
;  249          0310  2
;  250          0311  2     RETURN;
;  251          0312  2
;  252          0313  1     END;                                    ! End of routine PAS$WRITE_UNSIGNED
```

```
                                                        .TITLE  PAS$WRITE_UNSIGNED Write an unsigned integer
                                                        .IDENT  \1-002\

                                                        .PSECT  _PAS$CODE,NOWRT,  SHR,  PIC,2

                                    4C  55  21  00000 P.AAA:  .ASCII  \!UL\                                                 ;

                                                        .EXTRN  PAS$WRITE_UNSIGNED
                                                        .EXTRN  PAS$WRITEV_UNSIGNED
                                                        .EXTRN  PAS$$IO_HANDLER
                                                        .EXTRN  PAS$$VALIDATE_PFV
                                                        .EXTRN  PAS$$INIT_WRITE
                                                        .EXTRN  PAS$$SIGNAL, PAS$K_NEGWIDDIG
                                                        .EXTRN  SYS$FAO, PAS$K_LINTOOLON
                                                        .EXTRN  PAS$$END_WRITE

                                          03FC 00000    .ENTRY  PAS$WRITE_UNSIGNED, Save R2,R3,R4,R5,R6,R7,-          ; 0144
                                                                R8,R9
                                59 00000000G   00  9E 00002    MOVAB   PAS$$SIGNAL, R9                                 ;
                                5E             24  C2 00009    SUBL2   #36, SP                                         ;
                                               7E  D4 0000C    CLRL    ERROR_ADDR                                      ; 0195
                                         04    AE  7C 0000E    CLRQ    UNWIND_ACT                                      ;
                                6D       00B2  CF  DE 00011    MOVAL   6$, (FP)                                        ;
                                04             6C  91 00016    CMPB    (AP), #4                                        ; 0218
                                               04  1F 00019    BLSSU   1$                                              ;
                                         6E      10  AC  D0 0001B    MOVL    ERROR, ERROR_ADDR                       ; 0220
                                         56      04  AC  D0 0001F 1$:  MOVL    PFV, R6                                 ; 0222
                                   08    AE          56  D0 00023    MOVL    R6, PFV_ADDR                            ;
                                            00000000G  00  16 00027    JSB     PAS$$VALIDATE_PFV                       ; 0228
                                   04    AE          01  D0 0002D    MOVL    #1, UNWIND_ACT                          ; 0234
                                            00000000G  00  16 00031    JSB     PAS$$INIT_WRITE                         ; 0240
                                                  0C   AC  D5 00037    TSTL    TOTAL_WIDTH                             ; 0246
                                                       0A  18 0003A    BGEQ    2$                                      ;
                                                       7E  D4 0003C    CLRL    -(SP)                                   ; 0248
                                         7E          00G  8F  9A 0003E    MOVZBL  #PAS$K_NEGWIDDIG, -(SP)                 ;
                                         69               02  FB 00042    CALLS   #2, PAS$$SIGNAL                         ;
                                                          04  00045    RET                                             ;
                                   14    AE  010E000C     8F  D0 00046 2$:  MOVL    #17694732, DESCR                        ; 0254
                                   18    AE        1C     AE  9E 0004E    MOVAB   STRING, DESCR+4                         ; 0257
                                   0C    AE  010E0003     8F  D0 00053    MOVL    #17694723, CTRSTR_DESCR                 ; 0260
                                   10    AE        9F     AF  9E 0005B    MOVAB   P.AAA, CTRSTR_DESCR+4                   ; 0261
                                               08   AC  DD 00060    PUSHL   INTEGER                                 ; 0266
                                               18   AE  9F 00063    PUSHAB  DESCR                                   ;
                                               1C   AE  9F 00066    PUSHAB  DESCR                                   ;
                                               18   AE  9F 00069    PUSHAB  CTRSTR_DESCR                            ;
                             00000000G         00       04  FB 0006C    CALLS   #4, SYS$FAO                             ;
                                               52       0C  AC  D0 00073    MOVL    TOTAL_WIDTH, FIELD_WIDTH        ; 0273
                                               58       14  AE  3C 00077    MOVZWL  DESCR, R8                       ; 0274
                                               50           58  D0 0007B    MOVL    R8, INT_DIGITS                  ;
                                               50           52  D0 0007E    MOVL    FIELD_WIDTH, R0                 ; 0280
                                               58           50  D1 00081    CMPL    R0, R8                          ;
                                                            03  18 00084    BGEQ    3$                              ;
                                               50           58  D0 00086    MOVL    R8, R0                          ;
                                               52           50  D0 00089 3$:  MOVL    R0, FIELD_WIDTH                 ;
                                50             52       EC  A7  C1 0008C    ADDL3   -20(FCB), FIELD_WIDTH, R0       ; 0284
```

```
                                50        F0   A7  C2 00091          SUBL2   -16(FCB), EXTRA                                  ;
                                                0C  15 00095          BLEQ    4$                                              ; 0285
                                                50  DD 00097          PUSHL   EXTRA                                           ; 0287
                                                01  DD 00099          PUSHL   #1
                                7E        00G   8F  9A 0009B          MOVZBL  #PAS$K_LINTOOLON, -(SP)
                                69              03  FB 0009F          CALLS   #3, PAS$$SIGNAL
                                                    04 000A2          RET
                                58              52  D1 000A3 4$:      CMPL    FIELD_WIDTH, R8                                 ; 0294
                                                0E  15 000A6          BLEQ    5$
                                52              58  C2 000A8          SUBL2   R8, R2                                          ; 0296
          52            20      6E              00  2C 000AB          MOVC5   #0, (SP), #32, R2, @-20(FCB)
                                            EC  B7     000B0
                            EC  A7              53  D0 000B2          MOVL    R3, -20(FCB)
                    EC  B7  18  BE              58  28 000B6 5$:      MOVC3   R8, @DESCR+4, @-20(FCB)                         ; 0302
                            EC  A7              53  D0 000BC          MOVL    R3, -20(FCB)
                                    00000000G   00  16 000C0          JSB     PAS$$END_WRITE                                  ; 0309
                                                    04 000C6          RET                                                     ; 0313
                                                  0000 000C7 6$:      .WORD   Save nothing                                    ; 0195
                                50        08    AC  D0 000C9          MOVL    8(AP), R0
                                50        04    A0  D0 000CD          MOVL    4(R0), R0
                                          D8    A0  9F 000D1          PUSHAB  ERROR_ADDR
                                          DC    A0  9F 000D4          PUSHAB  UNWIND_ACT
                                          E0    A0  9F 000D7          PUSHAB  PFV_ADDR
                                                03  DD 000DA          PUSHL   #3
                                                5E  DD 000DC          PUSHL   SP
                                7E        04    AC  7D 000DE          MOVQ    4(AP), -(SP)
                        00000000G       00      03  FB 000E2          CALLS   #3, PAS$$IO_HANDLER
                                                    04 000E9          RET

; Routine Size:  234 bytes,    Routine Base:  _PAS$CODE + 0003

;  253          0314  1
;  254          0315  1 !<BLF/PAGE>
```

```
;  256          0316  1 %SBTTL 'PAS$WRITEV_UNSIGNED - Write unsigned to string'
;  257          0317  1 GLOBAL ROUTINE PAS$WRITEV_UNSIGNED (
;  258          0318  1         MAX_LENGTH: WORD,                          ! Maximum length of string
;  259          0319  1         STRING_LINE: REF VECTOR [, WORD],          ! String to write to
;  260          0320  1         VALUE,                                     ! Value to write
;  261          0321  1         TOTAL_WIDTH: SIGNED,                       ! Total field width
;  262          0322  1         ERROR                                      ! Error unwind address
;  263          0323  1     ) : NOVALUE =
;  264          0324  1
;  265          0325  1 !++
;  266          0326  1 ! FUNCTIONAL DESCRIPTION:
;  267          0327  1 !
;  268          0328  1 !       This procedure writes an un signed integer to the specified string.
;  269          0329  1 !
;  270          0330  1 ! CALLING SEQUENCE:
;  271          0331  1 !
;  272          0332  1 !       CALL PAS$WRITEV_UNSIGNED (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
;  273          0333  1 !          VALUE.rlu.v, TOTAL_WIDTH.r[.v [, ERROR.j.r])
;  274          0334  1 !
;  275          0335  1 ! FORMAL PARAMETERS:
;  276          0336  1 !
;  277          0337  1 !       MAX_LENGTH      - The maximum length of STRING_LINE.
;  278          0338  1 !
;  279          0339  1 !       STRING_LINE     - A varying string to which the output will be appended.
;  280          0340  1 !
;  281          0341  1 !       VALUE           - The unsigned integer to write.
;  282          0342  1 !
;  283          0343  1 !       TOTAL_WIDTH     - The width of the field to write.
;  284          0344  1 !
;  285          0345  1 !       ERROR           - Optional.  If specified, the address to unwind to
;  286          0346  1 !                         in case of an error.
;  287          0347  1 !
;  288          0348  1 ! IMPLICIT INPUTS:
;  289          0349  1 !
;  290          0350  1 !       NONE
;  291          0351  1 !
;  292          0352  1 ! IMPLICIT OUTPUTS:
;  293          0353  1 !
;  294          0354  1 !       NONE
;  295          0355  1 !
;  296          0356  1 ! ROUTINE VALUE:
;  297          0357  1 !
;  298          0358  1 !       NONE
;  299          0359  1 !
;  300          0360  1 ! SIDE EFFECTS:
;  301          0361  1 !
;  302          0362  1 !       NONE
;  303          0363  1 !
;  304          0364  1 ! SIGNALLED ERRORS:
;  305          0365  1 !
;  306          0366  1 !       See PAS$WRITE_UNSIGNED
;  307          0367  1 !
;  308          0368  1 !--
;  309          0369  1
;  310          0370  2     BEGIN
;  311          0371  2
;  312          0372  2     LOCAL
```

```
;  313          0373  2          PFV: $PAS$PFV_FILE_VARIABLE,    ! Pascal File Variable
;  314          0374  2          ARG_LIST: VECTOR [4, LONG],     ! Argument list
;  315          0375  2          PFV_ADDR: VOLATILE,             ! Enable argument
;  316          0376  2          UNWIND_ACT: VOLATILE,           ! Enable argument
;  317          0377  2          ERROR_ADDR: VOLATILE;           ! Enable argument
;  318          0378  2
;  319          0379  2      BUILTIN
;  320          0380  2          ACTUALCOUNT;                    ! Count of arguments
;  321          0381  2
;  322          0382  2      ENABLE
;  323          0383  2          PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  324          0384  2
;  325          0385  2      !+
;  326          0386  2      ! Get ERROR parameter, if present.
;  327          0387  2      !-
;  328          0388  2
;  329          0389  2      IF ACTUALCOUNT () GEQU 5
;  330          0390  2      THEN
;  331          0391  2          ERROR_ADDR = .ERROR;            ! Set unwind address
;  332          0392  2
;  333          0393  2      PFV_ADDR = PFV [PFV$R_PFV];         ! Set PFV address
;  334          0394  2
;  335          0395  2      !+
;  336          0396  2      ! Set up ARG_LIST.
;  337          0397  2      !-
;  338          0398  2
;  339          0399  2      ARG_LIST [0] = 3;                   ! Three arguments
;  340          0400  2      ARG_LIST [1] = PFV [PFV$R_PFV];     ! PFV address
;  341          0401  2      ARG_LIST [2] = .VALUE;              ! Unsigned integer to write
;  342          0402  2      ARG_LIST [3] = .TOTAL_WIDTH;        ! Field width
;  343          0403  2
;  344          0404  2      !+
;  345          0405  2      ! Call PAS$$DO_WRITEV to do the work, giving it the address of
;  346          0406  2      ! PAS$WRITE_UNSIGNED to call.
;  347          0407  2      !-
;  348          0408  2
;  349          0409  2      PAS$$DO_WRITEV (PFV [PFV$R_PFV], .MAX_LENGTH, STRING_LINE [0], ARG_LIST,
;  350          0410  2          PAS$WRITE_UNSIGNED);
;  351          0411  2
;  352          0412  2      RETURN;
;  353          0413  2
;  354          0414  1      END;                                ! End of routine PAS$WRITEV_UNSIGNED


                                                 .EXTRN  PAS$$DO_WRITEV

                                    007C 00000   .ENTRY  PAS$WRITEV_UNSIGNED, Save R2,R3,R4,R5,R6    ; 0317
                          5E            28  C2 00002   SUBL2  #40, SP                                ;
                                        7E  D4 00005   CLRL   ERROR_ADDR                             ; 0370
                              04        AE  7C 00007   CLRQ   UNWIND_ACT                             ;
                          6D      0039  CF  DE 0000A   MOVAL  2$, (FP)                               ;
                          05            6C  91 0000F   CMPB   (AP), #5                               ; 0389
                                        04  1F 00012   BLSSU  1$                                     ;
                          6E        14  AC  D0 00014   MOVL   ERROR, ERROR_ADDR                      ; 0391
                    08    AE        1C  AE  9E 00018 1$:  MOVAB  PFV, PFV_ADDR                       ; 0393
                    0C    AE            03  D0 0001D   MOVL   #3, ARG_LIST                           ; 0399
```

```
                          10   AE      1C   AE  9E 00021          MOVAB   PFV, ARG_LIST+4                          ; 0400
                          14   AE      0C   AC  7D 00026          MOVQ    VALUE, ARG_LIST+8                       ; 0401
                               55    FEE7   CF  9E 0002B          MOVAB   PAS$WRITE_UNSIGNED, R5                  ; 0409
                               54      0C   AE  9E 00030          MOVAB   ARG_LIST, R4                            ;
                               56      1C   AE  9E 00034          MOVAB   PFV, R6                                 ;
                               53      08   AC  D0 00038          MOVL    STRING_LINE, R3                         ;
                               52      04   AC  3C 0003C          MOVZWL  MAX_LENGTH, R2                          ;
                                  00000000G   00  16 00040          JSB     PAS$$DO_WRITEV                          ;
                                              04 00046          RET                                             ; 0414
                                            0000 00047 2$:      .WORD   Save nothing                            ; 0370
                               50      08   AC  D0 00049          MOVL    8(AP), R0                               ;
                               50      04   A0  D0 0004D          MOVL    4(R0), R0                               ;
                                       D4   A0  9F 00051          PUSHAB  ERROR_ADDR                              ;
                                       D8   A0  9F 00054          PUSHAB  UNWIND_ACT                              ;
                                       DC   A0  9F 00057          PUSHAB  PFV_ADDR                                ;
                                            03  DD 0005A          PUSHL   #3                                      ;
                                            5E  DD 0005C          PUSHL   SP                                      ;
                               7E      04   AC  7D 0005E          MOVQ    4(AP), -(SP)                            ;
                     00000000G   00           03  FB 00062          CALLS   #3, PAS$$IO_HANDLER                     ;
                                              04 00069          RET                                             ;

; Routine Size:  106 bytes,    Routine Base:  _PAS$CODE + 00ED

;  355          0415  1
;  356          0416  1 !<BLF/PAGE>
```

```
;  358          0417  1 END                                   ! End of module PAS$WRITE_UNSIGNED
;  359          0418  1
;  360          0419  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _PAS$CODE                        343  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                                 Library Statistics
;                                                 -------- Symbols --------      Pages       Processing
;       File                                      Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1               9776         7         0         581       00:01.0
;  _$255$DUA28:[PASRTL.OBJ]PASLIB.L32;1             427        96        22          33       00:00.4



;                                       COMMAND QUALIFIERS

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

;  361          0420  0
; Size:         340 code + 3 data bytes
; Run Time:        00:08.3
; Elapsed Time:    00:19.3
; Lines/CPU Min:    3021
; Lexemes/CPU-Min: 16280
; Memory Used:  101 pages
; Compilation Complete
```

PASWRIREH LIS

PASWRITEL LIS

PASWRIREG LIS

PATDEF MDL

PASWRISTR LIS

PATCH

PASWRIREF LIS

PATCH MAP

SRMDEF MDL

PASWRIRFH LIS

PASWRIVAR LIS

PASWRIRED LIS

PASWRIUNS LIS

PASWRIINT LIS

PASWRIOCT LIS

PASWRIRFG LIS

BSTRUC REQ

PASWRIRFF LIS

CHRKEY REQ

PASWRIRFD LIS