```
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
```

```
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR     IIIIII      OOOOOO      CCCCCCCC  TTTTTTTTTT
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR     IIIIII      OOOOOO      CCCCCCCC  TTTTTTTTTT
PP      PP   AA    AA   SS        WW      WW  RR      RR     II        OO      OO  CC            TT
PP      PP   AA    AA   SS        WW      WW  RR      RR     II        OO      OO  CC            TT
PP      PP   AA    AA   SS        WW      WW  RR      RR     II        OO      OO  CC            TT
PP      PP   AA    AA   SS        WW      WW  RR      RR     II        OO      OO  CC            TT
PPPPPPPP     AA    AA     SSSSSS  WW      WW  RRRRRRRR       II        OO      OO  CC            TT
PPPPPPPP     AA    AA     SSSSSS  WW      WW  RRRRRRRR       II        OO      OO  CC            TT
PP           AAAAAAAAAA         SS WW  WW  WW  RR  RR         II        OO      OO  CC            TT
PP           AAAAAAAAAA         SS WW  WW  WW  RR  RR         II        OO      OO  CC            TT
PP           AA    AA           SS WWWW  WWWW  RR    RR       II        OO      OO  CC            TT
PP           AA    AA           SS WWWW  WWWW  RR    RR       II        OO      OO  CC            TT
PP           AA    AA   SSSSSSSS   WW      WW  RR      RR   IIIIII      OOOOOO      CCCCCCCC      TT
PP           AA    AA   SSSSSSSS   WW      WW  RR      RR   IIIIII      OOOOOO      CCCCCCCC      TT


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_OCT ( %TITLE 'Write a value in base 8'
;     2          0002  0                  IDENT = '1-002'                    ! File: PASWRIOCT.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 procedures which write a value in
;    36          0036  1 !       base 8 to a textfile or string.
;    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     PAS$WRITE_OCT: NOVALUE,                  ! Write to textfile
;    62          0124  1     PAS$WRITEV_OCT: NOVALUE;                 ! Write to string
;    63          0125  1
;    64          0126  1 ! MACROS:
;    65          0127  1 !
;    66          0128  1 !       NONE
;    67          0129  1 !
;    68          0130  1 ! EQUATED SYMBOLS:
;    69          0131  1 !
;    70          0132  1 !       NONE
;    71          0133  1 !
;    72          0134  1 ! FIELDS:
;    73          0135  1 !
;    74          0136  1 !       NONE
;    75          0137  1 !
;    76          0138  1 ! OWN STORAGE:
;    77          0139  1 !
;    78          0140  1 !       NONE
;    79          0141  1 !
;    80          0142  1
;    81          0143  1 !! If this is for a V2 system, redefine OTS$CVT_L_TO as PAS$CVT_L_TO.
;    82        L 0144  1 %IF %VARIANT
;    83        U 0145  1 %THEN
;    84        U 0146  1 UNDECLARE
;    85        U 0147  1     OTS$CVT_L_TO;
;    86        U 0148  1 EXTERNAL ROUTINE
;    87        U 0149  1     PAS$CVT_L_TO;
;    88        U 0150  1 BIND ROUTINE
;    89        U 0151  1     OTS$CVT_L_TO = PAS$CVT_L_TO;
;    90          0152  1 %FI
```

```
;    92          0153  1 %SBTTL 'PAS$WRITE_OCT - Write a value in octal'
;    93          0154  1 GLOBAL ROUTINE PAS$WRITE_OCT (
;    94          0155  1         PFV: REF $PAS$PFV_FILE_VARIABLE,          ! File variable
;    95          0156  1         NBITS,                                    ! Size of value in bits
;    96          0157  1         VALUE,                                    ! Address of value
;    97          0158  1         TOTAL_WIDTH: SIGNED,                      ! Total field width
;    98          0159  1         ERROR,                                    ! Error unwind address
;    99          0160  1         MIN_DIGITS: SIGNED                        ! Minimum number of digits
;   100          0161  1     ): NOVALUE =
;   101          0162  1
;   102          0163  1 !++
;   103          0164  1 ! FUNCTIONAL DESCRIPTION:
;   104          0165  1 !
;   105          0166  1 !       This procedure writes a base 8 representation of a value to the
;   106          0167  1 !       specified textfile.
;   107          0168  1 !
;   108          0169  1 ! CALLING SEQUENCE:
;   109          0170  1 !
;   110          0171  1 !       CALL PAS$WRITE_OCT (PFV.mr.r, NBITS.rl.v, VALUE.rz.r, TOTAL_WIDTH.rl.v
;   111          0172  1 !                           [, [ERROR.j.r] [, MIN_DIGITS.rl.v]])
;   112          0173  1 !
;   113          0174  1 ! FORMAL PARAMETERS:
;   114          0175  1 !
;   115          0176  1 !       PFV             - The Pascal File Variable (PFV) passed by reference.
;   116          0177  1 !                         The structure of the PFV is defined in PASPFV.REQ.
;   117          0178  1 !
;   118          0179  1 !       NBITS           - The size of VALUE in bits.
;   119          0180  1 !
;   120          0181  1 !       VALUE           - The address of the value to write.
;   121          0182  1 !
;   122          0183  1 !       TOTAL_WIDTH     - Total field width.
;   123          0184  1 !
;   124          0185  1 !       ERROR           - Optional.  Address to unwind to if an error occurs.
;   125          0186  1 !
;   126          0187  1 !       MIN_DIGITS      - Optional.  The minimum number of digits to appear
;   127          0188  1 !                         in the result.  Defaults to the minimnum number of
;   128          0189  1 !                         digits needed to represent every bit of the value.
;   129          0190  1 !
;   130          0191  1 ! IMPLICIT INPUTS:
;   131          0192  1 !
;   132          0193  1 !       NONE
;   133          0194  1 !
;   134          0195  1 ! IMPLICIT OUTPUTS:
;   135          0196  1 !
;   136          0197  1 !       NONE
;   137          0198  1 !
;   138          0199  1 ! ROUTINE VALUE:
;   139          0200  1 !
;   140          0201  1 !       NONE
;   141          0202  1 !
;   142          0203  1 ! SIDE EFFECTS:
;   143          0204  1 !
;   144          0205  1 !       If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
;   145          0206  1 !
;   146          0207  1 ! SIGNALLED ERRORS:
;   147          0208  1 !
;   148          0209  1 !       NEGWIDDIG - negative field width or digits specification not allowed
```

```
;  149          0210  1 !       LINTOOLON - line too long
;  150          0211  1 !
;  151          0212  1 !--
;  152          0213  1
;  153          0214  2     BEGIN
;  154          0215  2
;  155          0216  2     LOCAL
;  156          0217  2         FCB: REF $PAS$FCB_CONTROL_BLOCK,                ! File control block
;  157          0218  2         ACTUAL_DIGITS,                                  ! Number of digits actually used
;  158          0219  2         ACTUAL_NBITS,                                   ! Value size actually used
;  159          0220  2         DESCR: BLOCK [8, BYTE],                         ! String descriptor
;  160          0221  2         PFV_ADDR: VOLATILE,                             ! Enable argument
;  161          0222  2         UNWIND_ACT: VOLATILE,                           ! Enable argument
;  162          0223  2         ERROR_ADDR: VOLATILE;                           ! Enable argument
;  163          0224  2
;  164          0225  2     LITERAL
;  165          0226  2         M_SIZE_IN_BITS = %X'04';                        ! Flags argument for
;  166          0227  2                                                         ! convert routine
;  167          0228  2     BUILTIN
;  168          0229  2         ACTUALCOUNT;
;  169          0230  2
;  170          0231  2     ENABLE
;  171          0232  2         PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  172          0233  2
;  173          0234  2     !+
;  174          0235  2     ! Get ERROR parameter, if present.
;  175          0236  2     !-
;  176          0237  2
;  177          0238  2     IF ACTUALCOUNT () GEQU 5
;  178          0239  2     THEN
;  179          0240  2         ERROR_ADDR = .ERROR;            ! Set unwind address
;  180          0241  2
;  181          0242  2     PFV_ADDR = PFV [PFV$R_PFV];         ! Set PFV address
;  182          0243  2
;  183          0244  2     !+
;  184          0245  2     ! Validate PFV and get PFV.
;  185          0246  2     !-
;  186          0247  2
;  187          0248  2     PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
;  188          0249  2
;  189          0250  2     !+
;  190          0251  2     ! Set unwind action to unlock file.
;  191          0252  2     !-
;  192          0253  2
;  193          0254  2     UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
;  194          0255  2
;  195          0256  2     !+
;  196          0257  2     ! Do common initialization.
;  197          0258  2     !-
;  198          0259  2
;  199          0260  2     PAS$$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
;  200          0261  2
;  201          0262  2     !+
;  202          0263  2     ! Set initial values for conversion.
;  203          0264  2     !-
;  204          0265  2
;  205          0266  2     ACTUAL_NBITS = .NBITS;
```

```
;  206          0267  2      ACTUAL_DIGITS = (.ACTUAL_NBITS+2)/3;
;  207          0268  2
;  208          0269  2      !+
;  209          0270  2      ! Create result string descriptor with actual width.
;  210          0271  2      !-
;  211          0272  2
;  212          0273  2      DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
;  213          0274  2      DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  214          0275  2      DESCR [DSC$A_POINTER] = .FCB [FCB$A_RECORD_CUR];
;  215          0276  2      IF ACTUALCOUNT () GEQU 6
;  216          0277  2      THEN
;  217          0278  3          BEGIN
;  218          0279  3          ACTUAL_DIGITS = .MIN_DIGITS;
;  219          0280  3          IF .ACTUAL_DIGITS LSS 0
;  220          0281  3          THEN
;  221          0282  3              $PAS$IO_ERROR (PAS$_NEGWIDDIG,0);
;  222          0283  2          END;
;  223          0284  2      IF .TOTAL_WIDTH LSS 0
;  224          0285  2      THEN
;  225          0286  2          $PAS$IO_ERROR (PAS$_NEGWIDDIG,0);
;  226          0287  2      DESCR [DSC$W_LENGTH] = .TOTAL_WIDTH;
;  227          0288  2      !+
;  228          0289  2      ! Will TOTAL_WIDTH truncate the value?
;  229          0290  2      !+
;  230          0291  2      IF .TOTAL_WIDTH LSSU (.ACTUAL_NBITS+2)/3
;  231          0292  2      THEN
;  232          0293  3          BEGIN
;  233          0294  3          !+
;  234          0295  3          ! Change value size and bit offset to cause a truncated result.
;  235          0296  3          !-
;  236          0297  3          ACTUAL_NBITS = .TOTAL_WIDTH*3;
;  237          0298  2          END;
;  238          0299  2
;  239          0300  2      IF .ACTUAL_DIGITS GTR .TOTAL_WIDTH
;  240          0301  2      THEN
;  241          0302  2          ACTUAL_DIGITS = .TOTAL_WIDTH;
;  242          0303  2
;  243          0304  2      !+
;  244          0305  2      ! See if field will fit in record.
;  245          0306  2      !-
;  246          0307  2
;  247          0308  3      BEGIN
;  248          0309  3      LOCAL
;  249          0310  3          EXTRA;                      ! Extra characters past end of line
;  250          0311  3      EXTRA = (.FCB [FCB$A_RECORD_CUR] + .DESCR [DSC$W_LENGTH]) - .FCB [FCB$A_RECORD_END];
;  251          0312  3      IF .EXTRA GTR 0
;  252          0313  3      THEN
;  253          0314  3          $PAS$IO_ERROR (PAS$_LINTOOLON,1,.EXTRA);
;  254          0315  2      END;
;  255          0316  2
;  256          0317  2      !+
;  257          0318  2      ! Do the conversion.  It can't fail.
;  258          0319  2      !-
;  259          0320  2
;  260          0321  2      OTS$CVT_L_TO (.VALUE, DESCR, .ACTUAL_DIGITS, .ACTUAL_NBITS,
;  261          0322  2                    M_SIZE_IN_BITS);
;  262          0323  2
```

```
;  263          0324  2     FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_CUR] + .DESCR [DSC$W_LENGTH];
;  264          0325  2
;  265          0326  2     !+
;  266          0327  2     ! Call WRITE epilogue routine to move the last character written to the
;  267          0328  2     ! user's buffer and to unlock the file variable.
;  268          0329  2     !-
;  269          0330  2
;  270          0331  2     PAS$$END_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
;  271          0332  2
;  272          0333  2     RETURN;
;  273          0334  2
;  274          0335        END;                                    ! End of routine PAS$WRITE_OCT


                                                                    .TITLE  PAS$WRITE_OCT Write a value in base 8
                                                                    .IDENT  \1-002\

                                                                    .EXTRN  PAS$WRITE_OCT, PAS$WRITEV_OCT
                                                                    .EXTRN  PAS$$IO_HANDLER
                                                                    .EXTRN  PAS$$VALIDATE_PFV
                                                                    .EXTRN  PAS$$INIT_WRITE
                                                                    .EXTRN  PAS$$SIGNAL, PAS$K_NEGWIDDIG
                                                                    .EXTRN  PAS$K_LINTOOLON
                                                                    .EXTRN  OTS$CVT_L_TO, PAS$$END_WRITE

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

                                                01FC 00000          .ENTRY  PAS$WRITE_OCT, Save R2,R3,R4,R5,R6,R7,R8    ; 0154
                                    58 00000000G  00  9E 00002      MOVAB   PAS$$SIGNAL, R8
                                    5E            10  C2 00009      SUBL2   #16, SP
                                                  7E  D4 0000C      CLRL    ERROR_ADDR                                  ; 0214
                                              04  AE  7C 0000E      CLRQ    UNWIND_ACT
                                    6D        00A7  CF  DE 00011      MOVAL   8$, (FP)
                                    05              6C  91 00016      CMPB    (AP), #5                                    ; 0238
                                                    04  1F 00019      BLSSU   1$
                                    6E          14  AC  D0 0001B      MOVL    ERROR, ERROR_ADDR                           ; 0240
                                    56          04  AC  D0 0001F  1$:  MOVL    PFV, R6                                     ; 0242
                                08  AE                56  D0 00023      MOVL    R6, PFV_ADDR
                                       00000000G    00  16 00027      JSB     PAS$$VALIDATE_PFV                           ; 0248
                                04  AE                01  D0 0002D      MOVL    #1, UNWIND_ACT                              ; 0254
                                       00000000G    00  16 00031      JSB     PAS$$INIT_WRITE                             ; 0260
                                    52          08  AC  D0 00037      MOVL    NBITS, ACTUAL_NBITS                         ; 0266
                                    54          02  A2  9E 0003B      MOVAB   2(R2), R4                                   ; 0267
                                    54              03  C6 0003F      DIVL2   #3, R4
                                    55              54  D0 00042      MOVL    R4, ACTUAL_DIGITS
                                0E  AE        010E  8F  B0 00045      MOVW    #270, DESCR+2                               ; 0274
                                10  AE          EC  A7  D0 0004B      MOVL    -20(FCB), DESCR+4                           ; 0275
                                    06              6C  91 00050      CMPB    (AP), #6                                    ; 0276
                                                    06  1F 00053      BLSSU   2$
                                    55          18  AC  D0 00055      MOVL    MIN_DIGITS, ACTUAL_DIGITS                   ; 0279
                                                    06  19 00059      BLSS    3$                                          ; 0280
                                    53          10  AC  D0 0005B  2$:  MOVL    TOTAL_WIDTH, R3                             ; 0284
                                                0A  18 0005F      BGEQ    4$
                                                7E  D4 00061  3$:  CLRL    -(SP)                                       ; 0286
                                    7E          00G 8F  9A 00063      MOVZBL  #PAS$K_NEGWIDDIG, -(SP)
                                    68              02  FB 00067      CALLS   #2, PAS$$SIGNAL
                                                        04 0006A      RET
```

```
                     0C   AE            53  B0 0006B 4$:    MOVW    R3, DESCR                       ; 0287
                          54            53  D1 0006F        CMPL    R3, R4                          ; 0291
                                        04  1E 00072        BGEQU   5$                              ;
                     52                 53  C5 00074        MULL3   #3, R3, ACTUAL_NBITS            ; 0297
                          53            55  D1 00078 5$:    CMPL    ACTUAL_DIGITS, R3               ; 0300
                                        03  15 0007B        BLEQ    6$                              ;
                          55            53  D0 0007D        MOVL    R3, ACTUAL_DIGITS               ; 0302
                          50       0C   AE  3C 00080 6$:    MOVZWL  DESCR, R0                       ; 0311
                          50       EC   A7  C0 00084        ADDL2   -20(FCB), R0                    ;
                          50       F0   A7  C2 00088        SUBL2   -16(FCB), EXTRA                 ;
                                        0C  15 0008C        BLEQ    7$                              ; 0312
                                        50  DD 0008E        PUSHL   EXTRA                           ; 0314
                                        01  DD 00090        PUSHL   #1                              ;
                          7E       00G  8F  9A 00092        MOVZBL  #PAS$K_LINTOOLON, -(SP)         ;
                          68            03  FB 00096        CALLS   #3, PAS$$SIGNAL                 ;
                                        04  00099        RET                                     ;
                                        04  DD 0009A 7$:    PUSHL   #4                              ; 0321
                                        52  DD 0009C        PUSHL   ACTUAL_NBITS                    ;
                                        55  DD 0009E        PUSHL   ACTUAL_DIGITS                   ;
                                   18   AE  9F 000A0        PUSHAB  DESCR                           ;
                                   0C   AC  DD 000A3        PUSHL   VALUE                           ;
                00000000G          00       05  FB 000A6        CALLS   #5, OTS$CVT_L_TO                ;
                          50       0C   AE  3C 000AD        MOVZWL  DESCR, R0                       ; 0324
                     EC   A7            50  C0 000B1        ADDL2   R0, -20(FCB)                    ;
                               00000000G  00  16 000B5        JSB     PAS$$END_WRITE                  ; 0331
                                        04  000BB        RET                                     ; 0335
                                      0000  000BC 8$:    .WORD   Save nothing                    ; 0214
                          50       08   AC  D0 000BE        MOVL    8(AP), R0                       ;
                          50       04   A0  D0 000C2        MOVL    4(R0), R0                       ;
                                   EC   A0  9F 000C6        PUSHAB  ERROR_ADDR                      ;
                                   F0   A0  9F 000C9        PUSHAB  UNWIND_ACT                      ;
                                   F4   A0  9F 000CC        PUSHAB  PFV_ADDR                        ;
                                        03  DD 000CF        PUSHL   #3                              ;
                                        5E  DD 000D1        PUSHL   SP                              ;
                               7E       04  AC  7D 000D3        MOVQ    4(AP), -(SP)            ;
                00000000G      00           03  FB 000D7        CALLS   #3, PAS$$IO_HANDLER     ;
                                            04  000DE        RET                             ;

; Routine Size:  223 bytes,    Routine Base:  _PAS$CODE + 00J0

;  275          0336  1
;  276          0337  1 !<BLF/PAGE>
```

```
;  278          0338  1 %SBTTL 'PAS$WRITEV_OCT - Write value in base 8 to string'
;  279          0339  1 GLOBAL ROUTINE PAS$WRITEV_OCT (
;  280          0340  1         MAX_LENGTH: WORD,                              ! Maximum length of string
;  281          0341  1         STRING_LINE: REF VECTOR [, WORD],              ! String to write to
;  282          0342  1         NBITS,                                         ! Number of bits in VALUE
;  283          0343  1         VALUE: REF VECTOR [, BYTE],                    ! Value to write
;  284          0344  1         TOTAL_WIDTH: SIGNED,                           ! Total field width
;  285          0345  1         ERROR,                                         ! Error unwind address
;  286          0346  1         MIN_DIGITS: SIGNED                             ! Minimum number of digits
;  287          0347  1     ) : NOVALUE =
;  288          0348  1
;  289          0349  1 !++
;  290          0350  1 ! FUNCTIONAL DESCRIPTION:
;  291          0351  1 !
;  292          0352  1 !       This procedure writes a value in base 8 to the specified string.
;  293          0353  1 !
;  294          0354  1 ! CALLING SEQUENCE:
;  295          0355  1 !
;  296          0356  1 !       CALL PAS$WRITEV_OCT (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
;  297          0357  1 !          NBITS.rl.v, VALUE.rz.v, TOTAL_WIDTH.rl.v
;  298          0358  1 !          [, [ERROR.j.r] [,MIN_DIGITS.r[.v]]])
;  299          0359  1 !
;  300          0360  1 ! FORMAL PARAMETERS:
;  301          0361  1 !
;  302          0362  1 !       MAX_LENGTH      - The maximum length of STRING_LINE.
;  303          0363  1 !
;  304          0364  1 !       STRING_LINE     - A varying string to which the output will be appended.
;  305          0365  1 !
;  306          0366  1 !       NBITS           - Size of VALUE in bits.
;  307          0367  1 !
;  308          0368  1 !       VALUE           - The value to write.
;  309          0369  1 !
;  310          0370  1 !       TOTAL_WIDTH     - The width of the field to write.
;  311          0371  1 !
;  312          0372  1 !       ERROR           - Optional.  If specified, the address to unwind to
;  313          0373  1 !                         in case of an error.
;  314          0374  1 !
;  315          0375  1 !       MIN_DIGITS      - Minimum number of digits to write.  Defaults to
;  316          0376  1 !                         the minimum necessary to represent every bit of
;  317          0377  1 !                         the value.
;  318          0378  1 !
;  319          0379  1 ! IMPLICIT INPUTS:
;  320          0380  1 !
;  321          0381  1 !       NONE
;  322          0382  1 !
;  323          0383  1 ! IMPLICIT OUTPUTS:
;  324          0384  1 !
;  325          0385  1 !       NONE
;  326          0386  1 !
;  327          0387  1 ! ROUTINE VALUE:
;  328          0388  1 !
;  329          0389  1 !       NONE
;  330          0390  1 !
;  331          0391  1 ! SIDE EFFECTS:
;  332          0392  1 !
;  333          0393  1 !       NONE
;  334          0394  1 !
```

```
;  335          0395  1 ! SIGNALLED ERRORS:
;  336          0396  1 !
;  337          0397  1 !       See PAS$WRITE_OCT
;  338          0398  1 !
;  339          0399  1 !--
;  340          0400  1
;  341          0401  2     BEGIN
;  342          0402  2
;  343          0403  2     LOCAL
;  344          0404  2         PFV: $PAS$PFV_FILE_VARIABLE,    ! Pascal File Variable
;  345          0405  2         ARG_LIST: VECTOR [7, LONG],     ! Argument list
;  346          0406  2         PFV_ADDR: VOLATILE,             ! Enable argument
;  347          0407  2         UNWIND_ACT: VOLATILE,           ! Enable argument
;  348          0408  2         ERROR_ADDR: VOLATILE;           ! Enable argument
;  349          0409  2
;  350          0410  2     BUILTIN
;  351          0411  2         ACTUALCOUNT;                    ! Count of arguments
;  352          0412  2
;  353          0413  2     ENABLE
;  354          0414  2         PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     : Enable error handler
;  355          0415  2
;  356          0416  2     !+
;  357          0417  2     ! Get ERROR parameter, if present.
;  358          0418  2     !-
;  359          0419  2
;  360          0420  2     IF ACTUALCOUNT () GEQU 6
;  361          0421  2     THEN
;  362          0422  2         ERROR_ADDR = .ERROR;            ! Set unwind address
;  363          0423  2
;  364          0424  2     PFV_ADDR = PFV [PFV$R_PFV];         ! Set PFV address
;  365          0425  2
;  366          0426  2     !+
;  367          0427  2     ! Set up ARG_LIST.
;  368          0428  2     !-
;  369          0429  2
;  370          0430  2     ARG_LIST [0] = 4;                   ! Four arguments
;  371          0431  2     ARG_LIST [1] = PFV [PFV$R_PFV];     ! PFV address
;  372          0432  2     ARG_LIST [2] = .NBITS;              ! Number of bits
;  373          0433  2     ARG_LIST [3] = VALUE [0];           ! Value to write
;  374          0434  2     ARG_LIST [4] = .TOTAL_WIDTH;        ! Field width
;  375          0435  2     IF ACTUALCOUNT () GEQU 7
;  376          0436  2     THEN
;  377          0437  3         BEGIN
;  378          0438  3         ARG_LIST [0] = 6;               ! Add two more arguments
;  379          0439  3         ARG_LIST [5] = 0;               ! Error address
;  380          0440  3         ARG_LIST [6] = .MIN_DIGITS;     ! Minimum digits
;  381          0441  2         END;
;  382          0442  2
;  383          0443  2     !+
;  384          0444  2     ! Call PAS$$DO_WRITEV to do the work, giving it the address of
;  385          0445  2     ! PAS$WRITE_OCT to call.
;  386          0446  2     !-
;  387          0447  2
;  388          0448  2     PAS$$DO_WRITEV (PFV [PFV$R_PFV], .MAX_LENGTH, STRING_LINE [0], ARG_LIST,
;  389          0449  2         PAS$WRITE_OCT);
;  390          0450  2
;  391          0451  2     RETURN;
```

```
;  392          0452  2
;  393          0453  1     END;                                    ! End of routine PAS$WRITEV_OCT


                                                                    .EXTRN  PAS$$DO_WRITEV

                                                 007C 00000         .ENTRY  PAS$WRITEV_OCT, Save R2,R3,R4,R5,R6     ; 0339
                                      5E            34   C2 00002  SUBL2   #52, SP                                ;
                                                    7E   D4 00005  CLRL    ERROR_ADDR                              ; 0401
                                            04      AE   7C 00007  CLRQ    UNWIND_ACT                              ;
                                      6D      004F  CF   DE 0000A  MOVAL   3$, (FP)                                ;
                                      06            6C   91 0000F  CMPB    (AP), #6                                ; 0420
                                                    04   1F 00012  BLSSU   1$                                      ;
                                      6E      18    AC   D0 00014  MOVL    ERROR, ERROR_ADDR                       ; 0422
                                08    AE      28    AE   9E 00018  1$:     MOVAB   PFV, PFV_ADDR                   ; 0424
                                0C    AE            04   D0 0001D  MOVL    #4, ARG_LIST                            ; 0430
                                10    AE      28    AE   9E 00021  MOVAB   PFV, ARG_LIST+4                         ; 0431
                                14    AE      0C    AC   7D 00026  MOVQ    NBITS, ARG_LIST+8                       ; 0432
                                1C    AE      14    AC   D0 0002B  MOVL    TOTAL_WIDTH, ARG_LIST+16                ; 0434
                                      C7            6C   91 00030  CMPB    (AP), #7                                ; 0435
                                                    0C   1F 00033  BLSSU   2$                                      ;
                                0C    AE            06   D0 00035  MOVL    #6, ARG_LIST                            ; 0438
                                              20    AE   D4 00039  CLRL    ARG_LIST+20                             ; 0439
                                24    AE      1C    AC   D0 0003C  MOVL    MIN_DIGITS, ARG_LIST+24                 ; 0440
                                      55      FEDC  CF   9E 00041  2$:     MOVAB   PAS$WRITE_OCT, R5               ; 0448
                                      54      0C    AE   9E 00046  MOVAB   ARG_LIST, R4                            ;
                                      56      28    AE   9E 0004A  MOVAB   PFV, R6                                 ;
                                      53      08    AC   D0 0004E  MOVL    STRING_LINE, R3                         ;
                                      52      04    AC   3C 00052  MOVZWL  MAX_LENGTH, R2                          ;
                                          00000000G 00   16 00056  JSB     PAS$$DO_WRITEV                          ;
                                                    04   0005C  RET                                               ; 0453
                                                 0000 0005D  3$:     .WORD   Save nothing                         ; 0401
                                      50      08    AC   D0 0005F  MOVL    8(AP), R0                               ;
                                      50      04    A0   D0 00063  MOVL    4(R0), R0                               ;
                                              C8    A0   9F 00067  PUSHAB  ERROR_ADDR                              ;
                                              CC    A0   9F 0006A  PUSHAB  UNWIND_ACT                              ;
                                              D0    A0   9F 0006D  PUSHAB  PFV_ADDR                                ;
                                                    03   DD 00070  PUSHL   #3                                      ;
                                                    5E   DD 00072  PUSHL   SP                                      ;
                                      7E      04    AC   7D 00074  MOVQ    4(AP), -(SP)                            ;
                                  00000000G   00          03   FB 00078  CALLS   #3, PAS$$IO_HANDLER               ;
                                                    04   0007F  RET                                               ;

; Routine Size:  128 bytes,    Routine Base:  _PAS$CODE + 00DF

;  394          0454  1
;  395          0455  1 !<BLF/PAGE>
```

```
;  397          0456  1 END                                        ! End of module PAS$WRITE_OCT
;  398          0457  1
;  399          0458  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _PAS$CODE                       351  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         6         0         581       00:01.0
;  _$255$DUA28:[PASRTL.OBJ]PASLIB.L32;1             427        97        22          33       00:00.4
```

```
;                                  COMMAND QUALIFIERS

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

; Size:          351 code + 0 data bytes
; Run Time:          00:08.8
; Elapsed Time:      00:18.6
; Lines/CPU Min:      3108
; Lexemes/CPU-Min: 15319
; Memory Used:  103 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