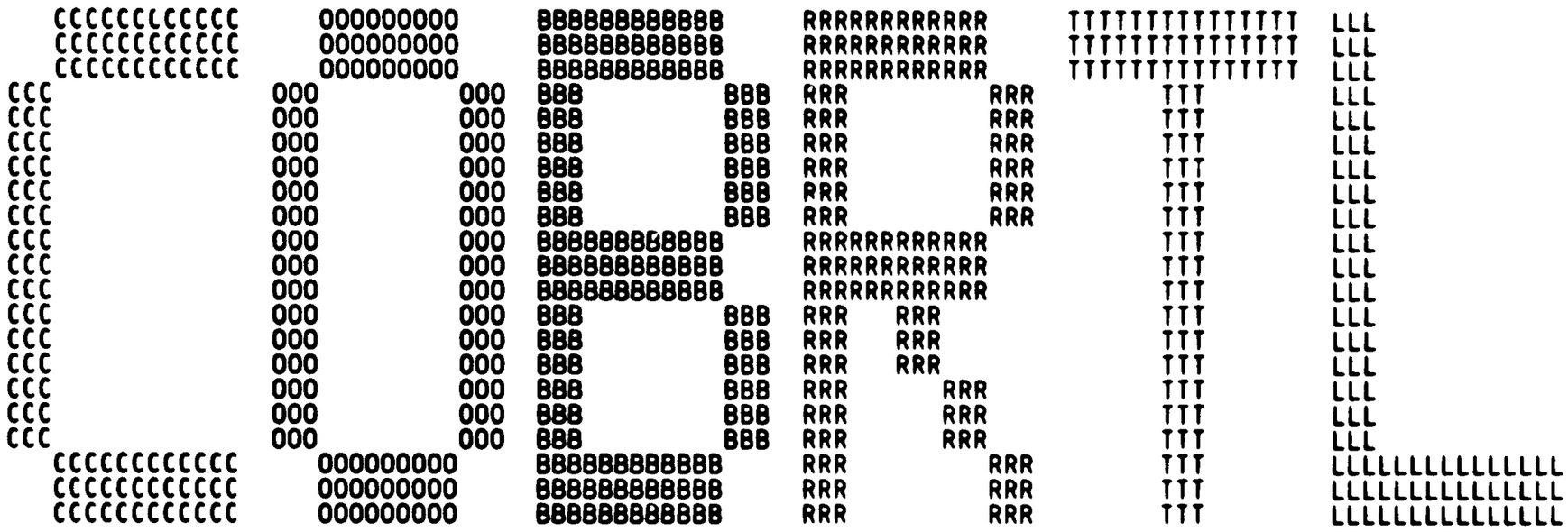
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL

FILEID**COBSETSWI

```
  CCCCCCCC    000000    BBBBBBBB      SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS  WW      WW    IIIIII
  CCCCCCCC    000000    BBBBBBBB      SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS  WW      WW    IIIIII
CC           OO      OO  BB      BB  SS        EE              TT      SS            WW      WW      II
CC           OO      OO  BB      BB  SS        EE              TT      SS            WW      WW      II
CC           OO      OO  BB      BB  SS        EE              TT      SS            WW      WW      II
CC           OO      OO  BB      BB  SS        EE              TT      SS            WW      WW      II
CC           OO      OO  BBBBBBBB      SSSSSS  EEEEEEEE        TT        SSSSSS    WW      WW      II
CC           OO      OO  BBBBBBBB      SSSSSS  EEEEEEEE        TT        SSSSSS    WW      WW      II
CC           OO      OO  BB      BB          SS  EE              TT              SS  WW  WW  WW      II
CC           OO      OO  BB      BB          SS  EE              TT              SS  WW  WW  WW      II
CC           OO      OO  BB      BB          SS  EE              TT              SS  WWWW  WWWW      II          ....
CC           OO      OO  BB      BB          SS  EE              TT              SS  WWWW  WWWW      II          ....
  CCCCCCCC    000000    BBBBBBBB    SSSSSSSS    EEEEEEEEEE      TT      SSSSSSSS    WW      WW    IIIIII        ....
  CCCCCCCC    000000    BBBBBBBB    SSSSSSSS    EEEEEEEEEE      TT      SSSSSSSS    WW      WW    IIIIII        ....

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 COB$SET_SWITCH(
;     2          0002  0                  IDENT = '1-004'          ! file: COBSETSWI.B32 EDIT:LB1004
;     3          0003  0                  ) =
;     4          0004  1 BEGIN
;     5          0005  1
;     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 ! FACILITY:     COBOL SUPPORT
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT
;    34          0034  1 !
;    35          0035  1 !       Sets external switches.
;    36          0036  1 !
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  Vax-11 User Mode
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: MLJ , CREATION DATE: 30-Oct-1979
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original, from COBSWITCH.B32 1-005.  MLJ 30-Oct-1979
;    45          0045  1 ! 1-002 - Signal COB$_SETEXTFAI if $CRELOG fails.  PDG 7-Aug-81
;    46          0046  1 ! 1-003 - Added EDIT phrase so CHECKIN creates a valid audit trail.
;    47          0047  1 !         Also updated copyright date.  LB 9-Aug-81
;    48          0048  1 ! 1-004 - Use logical LNM$C_NAMLENGTH instead of hard-coded 64.
;    49          0049  1 !                                                              LGB 31-OCT-83
;    50          0050  1 !--
```

```
;    52          0051  1 !
;    53          0052  1 ! SWITCHES
;    54          0053  1 !
;    55          0054  1
;    56          0055  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    57          0056  1
;    58          0057  1 !
;    59          0058  1 ! LINKAGES
;    60          0059  1 !
;    61          0060  1 LINKAGE
;    62          0061  1     L = JSB : GLOBAL(PTR=2, CNT=3, DIG=4);
;    63          0062  1 !
;    64          0063  1 ! TABLE OF CONTENTS:
;    65          0064  1 !
;    66          0065  1 FORWARD ROUTINE
;    67          0066  1         GET_DIGIT : L ;
;    68          0067  1 !
;    69          0068  1 ! INCLUDE FILES
;    70          0069  1 !
;    71          0070  1 REQUIRE 'RTLIN:RTLPSECT' ;              ! Macros to declare psects
;    72          0165  1 LIBRARY 'RTLSTARLE';
;    73          0166  1
;    74          0167  1 !
;    75          0168  1 ! MACROS
;    76          0169  1 !
;    77          0170  1 !       NONE
;    78          0171  1 !
;    79          0172  1 ! EQUATED SYMBOLS
;    80          0173  1 !
;    81          0174  1 EXTERNAL LITERAL
;    82          0175  1         COB$_SETEXTFAI;               ! SET external switch failed
;    83          0176  1
;    84          0177  1 !
;    85          0178  1 ! PSECT DECLARATIONS:
;    86          0179  1 !
;    87          0180  1
;    88          0181  1 DECLARE_PSECTS (COB) ;                ! Declare psects for COB$ facility
```

```
;    90          0182  1 GLOBAL ROUTINE COB$SET_SWITCH(SET_MASK,DO_SET)=
;    91          0183  1 !
;    92          0184  1 !++     FUNCTIONAL DESCRIPTION
;    93          0185  1 !
;    94          0186  1 !       This routine sets and clears external switches.
;    95          0187  1 !
;    96          0188  1 !
;    97          0189  1 !
;    98          0190  1 ! FORMAL PARAMETERS:
;    99          0191  1 !
;   100          0192  1 !       SET_MASK                - Bit mask of switches to be set or cleared.
;   101          0193  1 !                                 Bits 0 through 15 correspond to external
;   102          0194  1 !                                 switches numbered 1 through 16.
;   103          0195  1 !
;   104          0196  1 !       DO_SET                  - If true, the indicated switches are set.
;   105          0197  1 !                                 Otherwise, they are cleared.
;   106          0198  1 !
;   107          0199  1 ! IMPLICIT INPUTS:
;   108          0200  1 !
;   109          0201  1 !       Logical name table
;   110          0202  1 !
;   111          0203  1 ! IMPLICIT OUTPUTS:
;   112          0204  1 !
;   113          0205  1 !       New definition for logical name COB$SWITCHES.
;   114          0206  1 !
;   115          0207  1 ! ROUTINE VALUE:
;   116          0208  1 !
;   117          0209  1 !       NONE
;   118          0210  1 !
;   119          0211  1 ! COMPLETION CODES:
;   120          0212  1 !
;   121          0213  1 !       NONE
;   122          0214  1 !
;   123          0215  1 ! SIDE EFFECTS:
;   124          0216  1 !
;   125          0217  1 !       NONE
;   126          0218  1 !
;   127          0219  1 !--
;   128          0220  1 !
;   129          0221  2     BEGIN
;   130          0222  2     LOCAL
;   131          0223  2         BUFFER:         VECTOR[LNM$C_NAMLENGTH,BYTE],   ! Returned equivalence string
;   132          0224  2         BUFDESC:        VECTOR[2],                      ! Descriptor for BUFFER
;   133          0225  2         NAMDESC:        VECTOR[2],                      ! Descriptor for logical name
;   134          0226  2         LENGTH:         WORD,                           ! Length of equivalence string
;   135          0227  2         MASK,                                           ! Local bit mask of switches
;   136          0228  2         STATUS;                                         ! System service status
;   137          0229  2     GLOBAL REGISTER
;   138          0230  2         PTR=2, CNT=3, DIG=4;
;   139          0231  2     LABEL
;   140          0232  2         LOOP;
;   141          0233  2
;   142          0234  2     BUFDESC[0] = LNM$C_NAMLENGTH;                       ! Set up descriptors
;   143          0235  2     BUFDESC[1] = BUFFER;                                !      ...
;   144          0236  2     NAMDESC[0] = 12;                                    !      ...
;   145          0237  2     NAMDESC[1] = UPLIT('COB$SWITCHES');                 !      ...
;   146          0238  2
```

```
;  147          0239  2
;  148          0240  2      MASK = 0;                                       ! Initialize the mask
;  149        P 0241  2      IF $TRNLOG(                                     ! Translate COB$SWITCHES
;  150          0242  3          LOGNAM=NAMDESC, RSLLEN=LENGTH, RSLBUF=BUFDESC)
;  151          0243  2      THEN
;  152          0244  3          BEGIN
;  153          0245  3          PTR = BUFFER;                               ! Set up global registers
;  154          0246  3          CNT = .LENGTH;                              !      ...
;  155          0247  3
;  156          0248  3
;  157          0249  3  LOOP:
;  158          0250  4          BEGIN
;  159          0251  4          WHILE 1 DO
;  160          0252  5              BEGIN LOCAL VAL;
;  161          0253  5              !
;  162          0254  5              ! Skip leading non-digits.
;  163          0255  5              !
;  164          0256  5              WHILE (IF .CNT GTR 0 THEN NOT GET_DIGIT() ELSE LEAVE LOOP) DO 0;
;  165          0257  5              !
;  166          0258  5              ! Remember first digit.
;  167          0259  5              !
;  168          0260  5              VAL = .DIG;
;  169          0261  5              !
;  170          0262  5              ! Scan over digits to next non-digit, computing value in VAL.
;  171          0263  5              !
;  172          0264  5              WHILE (IF .CNT GTR 0 THEN GET_DIGIT() ELSE 0) DO VAL = .VAL * 10 + .DIG;
;  173          0265  5              !
;  174          0266  5              ! Put value into mask, if it is in range 1 to 16.
;  175          0267  5              !
;  176          0268  5              IF (VAL = .VAL - 1) LEQU 15 THEN MASK<.VAL,1> = 1;
;  177          0269  4              END;
;  178          0270  3          END;
;  179          0271  2      END; ! of labelled block LOOP
;  180          0272  2
;  181          0273  2
;  182          0274  2      ! The entire equivalence string for COB$SWITCHES has now been processed,
;  183          0275  2      ! and MASK contains the bit mask of switches that were set.  Modify it
;  184          0276  2      ! according to the routine parameters.
;  185          0277  2      !
;  186          0278  2      IF .DO_SET
;  187          0279  2          THEN MASK = .MASK OR .SET_MASK
;  188          0280  2          ELSE MASK = .MASK AND NOT .SET_MASK;
;  189          0281  2
;  190          0282  2
;  191          0283  2      ! Convert the bit mask to a new equivalence string.
;  192          0284  2      !
;  193          0285  2      PTR = BUFFER;
;  194          0286  2      (.PTR)<0,8> = %C' ';                            ! Use a leading space to be sure the
;  195          0287  2      PTR = .PTR + 1;                                 ! string isn't of zero length.
;  196          0288  2      INCR I FROM 0 TO 15 DO
;  197          0289  3          BEGIN
;  198          0290  3          IF .MASK<.I,1>
;  199          0291  3          THEN
;  200          0292  4              BEGIN LOCAL VAL;
;  201          0293  4              !
;  202          0294  4              ! The switch is set.  Put the decimal ASCII representation into
;  203          0295  4              ! the buffer.  Since we know the value is 1 through 16,
```

```
;  204          0296  4              ! a special-purpose conversion can be done.
;  205          0297  4              !
;  206          0298  4              VAL = .I + 1;
;  207          0299  4              IF .VAL GEQU 10
;  208          0300  4              THEN
;  209          0301  5                  BEGIN
;  210          0302  5                  (.PTR)<0,8> = %C'1';
;  211          0303  5                  PTR = .PTR + 1;
;  212          0304  5                  VAL = .VAL - 10;
;  213          0305  4                  END;
;  214          0306  4              (.PTR)<0,8> = .VAL + %C'0';
;  215          0307  4              PTR = .PTR + 1;
;  216          0308  4              (.PTR)<0,8> = %C' ';
;  217          0309  4              PTR = .PTR + 1;
;  218          0310  3              END;
;  219          0311  2          END;
;  220          0312  2
;  221          0313  2
;  222          0314  2      ! The string has now been constructed.  Redefine the logical name.
;  223          0315  2      !
;  224          0316  2      BUFDESC[0] = .PTR - BUFFER;
;  225          0317  2      BUFDESC[1] = BUFFER;
;  226        P 0318  3      IF NOT (STATUS = $CRELOG(
;  227        P 0319  3          TBLFLG=2,                       ! Process table
;  228        P 0320  3          LOGNAM=NAMDESC,
;  229          0321  3          EQLNAM=BUFDESC))
;  230          0322  2      THEN
;  231          0323  2          SIGNAL(COB$_SETEXTFAI, 0, .STATUS);
;  232          0324  2
;  233          0325  2      1
;  234          0326  1      END;


                                                        .TITLE  COB$SET_SWITCH
                                                        .IDENT  \1-004\

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

         53  45  48  43  54  49  57  53  24  42  4F  43  00000 P.AAA:  .ASCII  \COB$SWITCHES\                          ;

                                                        .EXTRN  COB$_SETEXTFAI, SYS$TRNLOG
                                                        .EXTRN  SYS$CRELOG

                                                00FC 00000      .ENTRY  COB$SET_SWITCH, Save R2,R3,R4,R5,R6,R7,R8,- ; 0182
                                                                        R9,R10,R11                                  ;
                                    5E      FEEC    CE  9E 00002        MOVAB   -276(SP), SP                        ;
                                0C  AE        FF    8F  9A 00007        MOVZBL  #255, BUFDESC                       ; 0234
                                10  AE        14    AE  9E 0000C        MOVAB   BUFFER, BUFDESC+4                   ; 0235
                                04  AE            0C  D0 00011        MOVL    #12, NAMDESC                        ; 0236
                                08  AE        DC    AF  9E 00015        MOVAB   P.AAA, NAMDESC+4                    ; 0237
                                                    56  D4 0001A        CLRL    MASK                                ; 0240
                                                    7E  7C 0001C        CLRQ    -(SP)                               ; 0242
                                                    7E  D4 0001E        CLRL    -(SP)                               ;
                                                18  AE  9F 00020        PUSHAB  BUFDESC                             ;
                                                10  AE  9F 00023        PUSHAB  LENGTH                              ;
                                                18  AE  9F 00026        PUSHAB  NAMDESC                             ;
                            00000000G   00          06  FB 00029        CALLS   #6, SYS$TRNLOG                      ;
```

```
                35          50 E9 00030         BLBC    R0, 4$                          ;
                52     14   AE 9E 00033         MOVAB   BUFFER, PTR                     ; 0245
                53          6E 3C 00037         MOVZWL  LENGTH, CNT                     ; 0246
                            53 D5 0003A 1$:     TSTL    CNT                             ; 0256
                            2A 15 0003C         BLEQ    4$                              ;
                         0000V 30 0003E         BSBW    GET_DIGIT                       ;
                F6          50 E9 00041         BLBC    R0, 1$                          ;
                55          54 D0 00044         MOVL    DIG, VAL                        ; 0260
                            53 D5 00047 2$:     TSTL    CNT                             ; 0264
                            10 15 00049         BLEQ    3$                              ;
                         0000V 30 0004B         BSBW    GET_DIGIT                       ;
                0A          50 E9 0004E         BLBC    R0, 3$                          ;
         50     55          0A C5 00051         MULL3   #10, VAL, R0                    ;
         55     50          54 C1 00055         ADDL3   DIG, R0, VAL                    ;
                            EC 11 00059         BRB     2$                              ;
                            55 D7 0005B 3$:     DECL    VAL                             ; 0268
                0F          55 D1 0005D         CMPL    VAL, #15                        ;
                            D8 1A 00060         BGTRU   1$                              ;
         D4     56          55 E2 00062         BBSS    VAL, MASK, 1$                   ;
                            D2 11 00066         BRB     1$                              ; 0251
                06     08   AC E9 00068 4$:     BLBC    DO_SET, 5$                      ; 0278
                56     04   AC C8 0006C         BISL2   SET_MASK, MASK                  ; 0279
                            04 11 00070         BRB     6$                              ;
                56     04   AC CA 00072 5$:     BICL2   SET_MASK, MASK                  ; 0280
                52     14   AE 9E 00076 6$:     MOVAB   BUFFER, PTR                     ; 0285
                82          20 90 0007A         MOVB    #32, (PTR)+                     ; 0286
                            51 D4 0007D         CLRL    I                               ; 0288
         16     56          51 E1 0007F 7$:     BBC     I, MASK, 9$                     ; 0290
                50     01   A1 9E 00083         MOVAB   1(R1), VAL                      ; 0298
                0A          50 D1 00087         CMPL    VAL, #10                        ; 0299
                            06 1F 0008A         BLSSU   8$                              ;
                82          31 90 0008C         MOVB    #49, (PTR)+                     ; 0302
                50          0A C2 0008F         SUBL2   #10, VAL                        ; 0304
         82     50          30 81 00092 8$:     ADDB3   #48, VAL, (PTR)+                ; 0306
                82          20 90 00096         MOVB    #32, (PTR)+                     ; 0308
         E2     51          0F F3 00099 9$:     AOBLEQ  #15, I, 7$                      ; 0288
                50     14   AE 9E 0009D         MOVAB   BUFFER, R0                      ; 0316
   0C    AE     52          50 C3 000A1         SUBL3   R0, PTR, BUFDESC                ;
            10  AE     14   AE 9E 000A6         MOVAB   BUFFER, BUFDESC+4               ; 0317
                            7E D4 000AB         CLRL    -(SP)                           ; 0321
                       10   AE 9F 000AD         PUSHAB  BUFDESC                         ;
                       0C   AE 9F 000B0         PUSHAB  NAMDESC                         ;
                            02 DD 000B3         PUSHL   #2                              ;
         00000000G  00      04 FB 000B5         CALLS   #4, SYS$CRELOG                  ;
                11          50 E8 000BC         BLBS    STATUS, 10$                     ;
                            50 DD 000BF         PUSHL   STATUS                          ; 0323
                            7E D4 000C1         CLRL    -(SP)                           ;
                    00000000G 8F DD 000C3       PUSHL   #COB$_SETEXTFAI                 ;
         00000000G  00      03 FB 000C9         CALLS   #3, LIB$SIGNAL                  ;
                50          01 D0 000D0 10$:    MOVL    #1, R0                          ; 0326
                               04 000D3         RET                                     ;

; Routine Size:  212 bytes,    Routine Base:  _COB$CODE + 000C
```

```
;  236     0327  1 ROUTINE GET_DIGIT: L=
;  237     0328  1 !
;  238     0329  1 !++     FUNCTIONAL DESCRIPTION
;  239     0330  1 !
;  240     0331  1 !
;  241     0332  1 ! Get next character from string, returning:
;  242     0333  1 !               PTR advanced to next character
;  243     0334  1 !               CNT decremented
;  244     0335  1 !               DIG containing the character less %C'0'
;  245     0336  1 !               Value true if and only if the character was a digit
;  246     0337  1 !
;  247     0338  1 !
;  248     0339  1 ! FORMAL PARAMETERS:
;  249     0340  1 !
;  250     0341  1 !
;  251     0342  1 ! IMPLICIT INPUTS:
;  252     0343  1 !
;  253     0344  1 !       NONE
;  254     0345  1 !
;  255     0346  1 ! IMPLICIT OUTPUTS:
;  256     0347  1 !
;  257     0348  1 !       NONE
;  258     0349  1 !
;  259     0350  1 ! ROUTINE VALUE:
;  260     0351  1 ! COMPLETION CODES:
;  261     0352  1 !
;  262     0353  1 !       NONE
;  263     0354  1 !
;  264     0355  1 ! SIDE EFFECTS:
;  265     0356  1 !
;  266     0357  1 !       NONE
;  267     0358  1 !
;  268     0359  1 !--
;  269     0360  1 !
;  270     0361  2     BEGIN
;  271     0362  2     EXTERNAL REGISTER
;  272     0363  2         PTR, CNT, DIG;
;  273     0364  2     LOCAL
;  274     0365  2         VAL;
;  275     0366  2
;  276     0367  2     VAL = 0;                                    ! Assume false value
;  277     0368  2     DIG = .(.PTR)<0,8>;                         ! Fetch character
;  278     0369  2     PTR = .PTR + 1;                             ! Advance pointer
;  279     0370  2     CNT = .CNT - 1;                             ! Decrement count
;  280     0371  2     DIG = .DIG - %C'0';                         ! Bias character
;  281     0372  2     IF .DIG LEQU 9 THEN VAL = .VAL + 1;         ! Set true value if need be
;  282     0373  2     .VAL                                        ! Return the value
;  283     0374  1     END;


                                50  D4 00000 GET_DIGIT:
                                              CLRL    VAL                                 ; 0367
                          54    82  9A 00002          MOVZBL  (PTR)+, DIG                 ; 0368
                                53  D7 00005          DECL    CNT                         ; 0370
```

```
                          54        30  C2 00007          SUBL2   #48, DIG                          ; 0371
                          09        54  D1 0000A          CMPL    DIG, #9                           ; 0372
                                    02  1A 0000D          BGTRU   1$                                ;
                                    50  D6 0000F          INCL    VAL                               ;
                                        05 00011 1$:       RSB                                       ; 0374

; Routine Size:  18 bytes,    Routine Base:  _COB$CODE + 00E0

;  284          0375  0 END ELUDOM



                                                           .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;  _COB$CODE                     242  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         5         0       581         00:00.8



;                                       COMMAND QUALIFIERS

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

; Size:         230 code + 12 data bytes
; Run Time:       00:04.9
; Elapsed Time:   00:21.5
; Lines/CPU Min:   4563
; Lexemes/CPU-Min: 21127
; Memory Used:  95 pages
; Compilation Complete
```

COBRMSBLO
LIS

COBSPANC
LIS

COBSWITCH
LIS

COBRESTVA
LIS

CONVSHR
MAP

CONVCOMIO
LIS

COBSETSWI
LIS

CONVDEF
MDL

CONVLINK
REQ

COBSPANC2
LIS

CONV

CONVERT
MAP

CONVCALL
LIS

COBVECTOR
LIS

CONVERT
REQ

RECLAIM
MAP

ADDKEY
LIS