```
EEEEEEEEEEEEEEE  DDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEE              DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEEEEEEEEEEE     DDD         DDD       TTT
EEEEEEEEEEEE     DDD         DDD       TTT
EEEEEEEEEEEE     DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEE              DDD         DDD       TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDD          TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD         TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD         TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD         TTT
```

```
PPPPPPPP    RRRRRRRR      AAAAAA    PPPPPPPP    PPPPPPPP    NN      NN  UU      UU  MM      MM
PPPPPPPP    RRRRRRRR      AAAAAA    PPPPPPPP    PPPPPPPP    NN      NN  UU      UU  MM      MM
PP      PP  RR      RR   AA    AA   PP      PP  PP      PP  NN      NN  UU      UU  MMMM  MMMM
PP      PP  RR      RR   AA    AA   PP      PP  PP      PP  NN      NN  UU      UU  MMMM  MMMM
PP      PP  RR      RR   AA    AA   PP      PP  PP      PP  NNNN    NN  UU      UU  MM  MM  MM
PP      PP  RR      RR   AA    AA   PP      PP  PP      PP  NNNN    NN  UU      UU  MM  MM  MM
PPPPPPPP    RRRRRRRR     AA    AA   PPPPPPPP    PPPPPPPP    NN  NN  NN  UU      UU  MM      MM
PPPPPPPP    RRRRRRRR     AA    AA   PPPPPPPP    PPPPPPPP    NN  NN  NN  UU      UU  MM      MM
PP          RR  RR       AAAAAAAAAA   PP          PP        NN    NNNN  UU      UU  MM      MM
PP          RR  RR       AAAAAAAAAA   PP          PP        NN    NNNN  UU      UU  MM      MM
PP          RR    RR     AA    AA   PP          PP        NN      NN  UU      UU  MM      MM
PP          RR    RR     AA    AA   PP          PP        NN      NN  UU      UU  MM      MM
PP          RR      RR   AA    AA   PP          PP        NN      NN  UUUUUUUUUU  MM      MM
PP          RR      RR   AA    AA   PP          PP        NN      NN  UUUUUUUUUU  MM      MM

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 %TITLE 'EDT$PRAPPNUM - include digits in a token'
;     2          0002  0 MODULE EDT$PRAPPNUM (                          ! Include digits in a token
;     3          0003  0                     IDENT = 'V04-000'                  ! File: PRAPPNUM.BLI Edit: JBS1004
;     4          0004  0                     ) =
;     5          0005  1 BEGIN
;     6          0006  1 !
;     7          0007  1 !**************************************************************************
;     8          0008  1 !*                                                                        *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                   *
;    12          0012  1 !*                                                                        *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    18          0018  1 !*  TRANSFERRED.                                                           *
;    19          0019  1 !*                                                                        *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    22          0022  1 !*  CORPORATION.                                                           *
;    23          0023  1 !*                                                                        *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;    26          0026  1 !*                                                                        *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !**************************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:     EDT -- The DEC Standard Editor
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       Include digits in a token.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: Bob Kushlis, CREATION DATE: December 12, 1978
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.  DJS 25-Feb-1981.  This module was created by
;    45          0045  1 !       extracting routine APPEND_NUM from module PARSER.
;    46          0046  1 ! 1-002 - Regularize headers.  JBS 12-Mar-1981
;    47          0047  1 ! 1-003 - Change alphanumeric test.  JBS 19-Jul-1982
;    48          0048  1 ! 1-004 - Add VT220 support conditional.  JBS 11-Feb-1983
;    49          0049  1 !--
;    50          0050  1
```

```
;    52          0051  1 %SBTTL 'Declarations'
;    53          0052  1 !
;    54          0053  1 ! TABLE OF CONTENTS:
;    55          0054  1 !
;    56          0055  1
;    57          0056  1 REQUIRE 'EDTSRC:TRAROUNAM';
;    58          0495  1
;    59          0496  1 FORWARD ROUTINE
;    60          0497  1     EDT$$PA_APPDIG;
;    61          0498  1
;    62          0499  1 !
;    63          0500  1 ! INCLUDE FILES:
;    64          0501  1 !
;    65          0502  1
;    66          0503  1 REQUIRE 'EDTSRC:EDTREQ';
;    67          0638  1
;    68          0639  1 REQUIRE 'EDTSRC:PARLITS';
;    69          0923  1
;    70          0924  1 LIBRARY 'EDTSRC:SUPPORTS';
;    71          0925  1
;    72          0926  1 !
;    73          0927  1 ! MACROS:
;    74          0928  1 !
;    75          0929  1 !       NONE
;    76          0930  1 !
;    77          0931  1 ! EQUATED SYMBOLS:
;    78          0932  1 !
;    79          0933  1 !       NONE
;    80          0934  1 !
;    81          0935  1 ! OWN STORAGE:
;    82          0936  1 !
;    83          0937  1 !       NONE
;    84          0938  1 !
;    85          0939  1 ! EXTERNAL REFERENCES:
;    86          0940  1 !
;    87          0941  1 !       In the routine
```

```
;    89          0942  1 %SBTTL 'EDT$$PA_APPDIG  - include digits in a token'
;    90          0943  1
;    91          0944  1 GLOBAL ROUTINE EDT$$PA_APPDIG                   ! Include digits in a token
;    92          0945  1     =
;    93          0946  1
;    94          0947  1 !++
;    95          0948  1 ! FUNCTIONAL DESCRIPTION:
;    96          0949  1 !
;    97          0950  1 !       Append a string of digits and letters to the current token.  Normally,
;    98          0951  1 !       CL_NAME type tokens are terminated by a digit.  In some cases we want
;    99          0952  1 !       to allow digits in the token.  This routine expands the length of the
;   100          0953  1 !       current token to include all digits and letters which immediately follow.
;   101          0954  1 !
;   102          0955  1 ! FORMAL PARAMETERS:
;   103          0956  1 !
;   104          0957  1 !       NONE
;   105          0958  1 !
;   106          0959  1 ! IMPLICIT INPUTS:
;   107          0960  1 !
;   108          0961  1 !       EDT$$A_CMD_BUF
;   109          0962  1 !       EDT$$C_PA_CH
;   110          0963  1 !       EDT$$G_PA_CURTOKLEN
;   111          0964  1 !       EDT$$G_PA_TOKCLASS
;   112          0965  1 !
;   113          0966  1 ! IMPLICIT OUTPUTS:
;   114          0967  1 !
;   115          0968  1 !       EDT$$G_PA_CURTOKLEN
;   116          0969  1 !
;   117          0970  1 ! ROUTINE VALUE:
;   118          0971  1 !
;   119          0972  1 !       1 = token is a CL_NAME
;   120          0973  1 !       0 = token is not a CL_NAME, no action taken
;   121          0974  1 !
;   122          0975  1 ! SIDE EFFECTS:
;   123          0976  1 !
;   124          0977  1 !       NONE
;   125          0978  1 !
;   126          0979  1 !--
;   127          0980  1
;   128          0981  2     BEGIN
;   129          0982  2
;   130          0983  2     EXTERNAL ROUTINE
;   131          0984  2         EDT$$PA_GETCH : NOVALUE;                ! Get the next character from the input line
;   132          0985  2
;   133          0986  2     EXTERNAL
;   134          0987  2         EDT$$A_CMD_BUF,                         ! Pointer into command buffer.
;   135          0988  2         EDT$$C_PA_CH,                           ! the currently being processed character
;   136          0989  2         EDT$$G_PA_CURTOKLEN,                    ! length of current token
;   137          0990  2
;   138        L 0991  2 %IF SUPPORT_VT220
;   139          0992  2 %THEN
;   140          0993  2         EDT$$B_CHAR_INFO : BLOCKVECTOR [256, 1, BYTE],  ! Information about characters
;   141          0994  2 %FI
;   142          0995  2
;   143          0996  2         EDT$$G_PA_TOKCLASS;                     ! class of current token
;   144          0997  2
;   145          0998  2     IF (.EDT$$G_PA_TOKCLASS NEQ CL_NAME) THEN RETURN (0);
```

```
;  146          0999  2
;  147        L 1000  2 %IF SUPPORT_VT220
;  148          1001  2 %THEN
;  149          1002  2
;  150          1003  4     WHILE ((.EDT$$B_CHAR_INFO [.EDT$$C_PA_CH, 0, 0, 2, 0] NEQ 0)        ! Alphabetic
;  151          1004  3         OR (.EDT$$B_CHAR_INFO [.EDT$$C_PA_CH, 0, 0, 8, 0] EQL %X'F0'))  ! Numeric
;  152          1005  2     DO
;  153        U 1006  2 %ELSE
;  154        U 1007  2
;  155        U 1008  2         WHILE (((.EDT$$C_PA_CH GEQ %C'A') AND (.EDT$$C_PA_CH LEQ %C'Z')) OR     !
;  156        U 1009  2             ((.EDT$$C_PA_CH GEQ %C'a') AND (.EDT$$C_PA_CH LEQ %C'a')) OR        !
;  157        U 1010  2             ((.EDT$$C_PA_CH GEQ %C'0') AND (.EDT$$C_PA_CH LEQ %C'9'))) DO
;  158          1011  2 %FI
;  159          1012  2
;  160          1013  3             BEGIN
;  161          1014  3             EDT$$PA_GETCH ();
;  162          1015  3             CH$WCHAR (.EDT$$C_PA_CH, CH$PLUS (.EDT$$A_CMD_BUF, -1));
;  163          1016  3             EDT$$G_PA_CURTOKLEN = .EDT$$G_PA_CURTOKLEN + 1;
;  164          1017  2             END;
;  165          1018  2
;  166          1019  2     RETURN (1);
;  167          1020  1     END;                                        ! of routine EDT$$PA_APPDIG


;                                                                       .TITLE  EDT$PRAPPNUM EDT$PRAPPNUM - include digits in a
                                                                                        token
                                                                        .IDENT  \V04-000\

                                                                        .EXTRN  EDT$$PA_GETCH, EDT$$A_CMD_BUF
                                                                        .EXTRN  EDT$$C_PA_CH, EDT$$G_PA_CURTOKLEN
                                                                        .EXTRN  EDT$$B_CHAR_INFO
                                                                        .EXTRN  EDT$$G_PA_TOKCLASS

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

                                    000C 00000          .ENTRY  EDT$$PA_APPDIG, Save R2,R3                      ; 0944
                 53 00000000G   00   9E 00002          MOVAB   EDT$$C_PA_CH, R3                                ;
                    00000000G   00   D5 00009          TSTL    EDT$$G_PA_TOKCLASS                              ; 0998
                                37   12 0000F          BNEQ    4$                                              ;
                 52             63   D0 00011          MOVL    EDT$$C_PA_CH, R2                                ; 1003
                 50 00000000G00042   9E 00014 1$:      MOVAB   EDT$$B_CHAR_INFO[R2], R0                        ;
                 03             60   93 0001C          BITB    (R0), #3                                        ;
                                06   12 0001F          BNEQ    2$                                              ;
              F0 8F             60   91 00021          CMPB    (R0), #240                                      ; 1004
                                1D   12 00025          BNEQ    3$                                              ;
       00000000G 00             00   FB 00027 2$:      CALLS   #0, EDT$$PA_GETCH                               ; 1014
                 50 00000000G   00   D0 0002E          MOVL    EDT$$A_CMD_BUF, R0                              ; 1015
                 52             63   D0 00035          MOVL    EDT$$C_PA_CH, R2                                ;
              FF 60             52   90 00038          MOVB    R2, -1(R0)                                      ;
                    00000000G   00   D6 0003C          INCL    EDT$$G_PA_CURTOKLEN                             ; 1016
                                D0   11 00042          BRB     1$                                              ; 1003
                 50             01   D0 00044 3$:      MOVL    #1, R0                                          ; 1019
                                     04 00047          RET                                                     ;
                                50   D4 00048 4$:      CLRL    R0                                              ; 1020
                                     04 0004A          RET                                                     ;

; Routine Size:  75 bytes,    Routine Base:  _EDT$CODE + 0000
```

```
;  168          1021  1
;  169          1022  1 !<BLF/PAGE>
```

```
1 END                                                       ! of module EDT$PRAPPNUM
1
0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
_EDT$CODE	75	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	-------- Symbols -------- Total	Loaded	Percent	Pages Mapped	Processing Time
_255DUA28:[EDT.SRC]EDT.L32;1	377	1	0	40	00:00.2
_255DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
_255DUA28:[EDT.SRC]SUPPORTS.L32;1	2	1	50	5	00:00.1

COMMAND QUALIFIERS

```
BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS$:PRAPPNUM/OBJ=OBJ$:PRAPPNUM MSRC$:PRAPPNUM.BLI/UPDATE=(ENH$:PRAP
PNUM)

; Size:           75 code + 0 data bytes
; Run Time:          00:13.8
; Elapsed Time:      00:17.1
; Lines/CPU Min:      4459
; Lexemes/CPU-Min:  13997
; Memory Used:   92 pages
; Compilation Complete
```

0137 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LXCOMNOO LIS
MCGETLIN LIS
MCRIGHT LIS
MCCHANGE LIS
MACCAL LIS
NOOPEN LIS
PRAPPNUM LIS
MSGTXT LIS
PRGETTOK LIS
PRISTOK LIS
LXPRINT LIS
MCTOP LIS
MCDOWN LIS
MCLEFT LIS
MCBOTTOM LIS
PAUDIT LIS
PRGETCHR LIS
LXCOM LIS
MAIN LIS
MCUP LIS