```
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     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  DDDDDDDDDDDD           TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
```

```
EX

Mo
--
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
SY
LB
LI
```

```
**FILE**ID**GETXLATE                                J 6

  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  XX      XX  LL            AAAAAA    TTTTTTTTTT  EEEEEEEEEE
  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  XX      XX  LL            AAAAAA    TTTTTTTTTT  EEEEEEEEEE
GG          EE              TT      XX      XX  LL          AA      AA      TT      EE
GG          EE              TT      XX      XX  LL          AA      AA      TT      EE
GG          EE              TT        XX  XX    LL          AA      AA      TT      EE
GG          EE              TT        XX  XX    LL          AA      AA      TT      EE
GG          EEEEEEEE        TT          XX      LL          AA      AA      TT      EEEEEEEE
GG          EEEEEEEE        TT          XX      LL          AA      AA      TT      EEEEEEEE
GG  GGGGGG  EE              TT        XX  XX    LL          AAAAAAAAAA      TT      EE
GG  GGGGGG  EE              TT        XX  XX    LL          AAAAAAAAAA      TT      EE
GG      GG  EE              TT      XX      XX  LL          AA      AA      TT      EE              ....
GG      GG  EE              TT      XX      XX  LL          AA      AA      TT      EE              ....
  GGGGGG    EEEEEEEEEE      TT      XX      XX  LLLLLLLLLL  AA      AA      TT      EEEEEEEEEE      ....
  GGGGGG    EEEEEEEEEE      TT      XX      XX  LLLLLLLLLL  AA      AA      TT      EEEEEEEEEE      ....

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
```

```
EDTS
V04-
; 1
; 1
; 1
; 1
; 1
; 1

; Ra

; 1
; 1
```

```
0001  0 %TITLE 'EDT$GETXLATE - Handles the nokeypad XLATE command'
0002  0 MODULE EDT$GETXLATE (                               ! nokeypad XLATE command
0003  0                 IDENT = 'V04-000'                           ! File: GETXLATE.B32 Edit: JBS1007
0004  0                 ) =
0005  1 BEGIN
0006  1 !
0007  1 !****************************************************************************
0008  1 !*                                                                          *
0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0011  1 !*  ALL RIGHTS RESERVED.                                                    *
0012  1 !*                                                                          *
0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0018  1 !*  TRANSFERRED.                                                            *
0019  1 !*                                                                          *
0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0022  1 !*  CORPORATION.                                                            *
0023  1 !*                                                                          *
0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0026  1 !*                                                                          *
0027  1 !*                                                                          *
0028  1 !****************************************************************************
0029  1 !
0030  1
0031  1 !++
0032  1 ! FACILITY:     EDT -- The DEC Standard Editor
0033  1 !
0034  1 ! ABSTRACT:
0035  1 !
0036  1 !       Handles the nokeypad XLATE command by setting up descriptor
0037  1 !       and passing string to user supplied XLATE.  String
0038  1 !       returned from XLATE is placed in the command buffer
0039  1 !       for execution.
0040  1 !
0041  1 !
0042  1 ! ENVIRONMENT:  Runs on Vax/Vms
0043  1 !
0044  1 ! AUTHOR: Shelly T. Solomon, CREATION DATE: 12-Jul-1982
0045  1 !
0046  1 ! MODIFIED BY:
0047  1 !
0048  1 ! 1-001 - Original.  STS 14-Aug-1982
0049  1 ! 1-002 - Return status of 1 rather that success status. STS 16-Aug-1982
0050  1 ! 1-003 - Save command buffer and allocate our own. STS 25-Aug-1982
0051  1 ! 1-004 - Unpack descriptor correctly. STS 13-Oct-1982
0052  1 ! 1-005 - Don't forget to deallocate heap storage.  JBS 14-Jun-1983
0053  1 ! 1-006 - Don't try to execute a 0-length returned string.  JBS 14-Jun-1983
0054  1 ! 1-007 - Don't print status just because no returned string.  JBS 17-Jun-1983
0055  1 !--
0056  1
```

```
;    58          0057  1 %SBTTL 'Declarations'
;    59          0058  1 !
;    60          0059  1 ! TABLE OF CONTENTS:
;    61          0060  1 !
;    62          0061  1
;    63          0062  1 REQUIRE 'EDTSRC:TRAROUNAM';
;    64          0501  1
;    65          0502  1 FORWARD ROUTINE
;    66          0503  1     EDT$$GET_XLATE;
;    67          0504  1
;    68          0505  1 !
;    69          0506  1 ! LINKAGES
;    70          0507  1 !
;    71          0508  1
;    72          0509  1 LINKAGE
;    73          0510  1     IOCALL = CALL (REGISTER = 1);
;    74          0511  1
;    75          0512  1 !
;    76          0513  1 ! INCLUDE FILES:
;    77          0514  1 !
;    78          0515  1
;    79          0516  1 REQUIRE 'EDTSRC:EDTREQ';
;    80          0651  1
;    81          0652  1 REQUIRE 'EDTSRC:SYSSYM';
;    82          0682  1
;    83          0683  1 !
;    84          0684  1 ! MACROS:
;    85          0685  1 !
;    86          0686  1 !       NONE
;    87          0687  1 !
;    88          0688  1 ! EQUATED SYMBOLS:
;    89          0689  1 !
;    90          0690  1 !       NONE
;    91          0691  1 !
;    92          0692  1 ! OWN STORAGE:
;    93          0693  1 !
;    94          0694  1 !       NONE
;    95          0695  1 !
;    96          0696  1 ! EXTERNAL REFERENCES:
;    97          0697  1 !
;    98          0698  1 ! in the routine
;    99          0699  1 !
```

```
1 %SBTTL 'EDT$$GETXLATE - nokeypad XLATE command'
1
1 GLOBAL ROUTINE EDT$$GET_XLATE (                        ! Nokeypad XLATE command
1     STRING,                                            ! Address of string to be XLATEd
1     LENGTH                                             ! Length of that string
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !     This is the routine called to handle the nokeypad XLATE command
1 !        It sets up a descriptor with the passed string and calls the
1 !        user supplied XLATE.  The returned string is placed in the
1 !        command buffer to be processed.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !        STRING  The string following the XLATE keyword to be passed to
1 !                the XLATE routine.
1 !
1 !        LENGTH  Length of above string
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !        EDT$$A_CMD_BUF  address of the command buffer
1 !
1 !        EDT$$A_CMD_END  address of the command buffer end
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !        NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !        1 = OK, 0 = error in command; message printed
1 !
1 ! SIDE EFFECTS:
1 !
1 !        Adds commands into the command buffer
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL ROUTINE
2         EDT$$CHM_PAREXE,                               ! Parse and execute a command
2         EDT$$MSG_BELL,                                 ! Output a message with bell
2         EDT$$ALO_HEAP,                                 ! Allocate heap storage
2         EDT$$DEA_HEAP : NOVALUE,                       ! Deallocate heap storage
2         STR$FREE1_DX;                                  ! Free a string
2
2     EXTERNAL
2         EDT$$A_XLATE_ROUT,                             ! Address of XLATE routine
2         EDT$$G_XLATE_ENV,                              ! Environment address for XLATE
2         EDT$$T_CMD_BUF,                                ! Command buffer
2         EDT$$A_CMD_BUF,                                ! Current location in command buffer
2         EDT$$A_CMD_END;                                ! End of command buffer
```

```
;  158       0757  2
;  159       0758  2      MESSAGES ((INVSUBCOM, INSMEM, COMEXHXLA, PASSTATUS));
;  160       0759  2
;  161       0760  2      LOCAL
;  162       0761  2          SAVE_CMD_BUF,
;  163       0762  2          SAVE_CMD_END,
;  164       0763  2          NEW_CMD_BUF : REF VECTOR [, BYTE],
;  165       0764  2          CMD_DESC : BLOCK [8, BYTE],
;  166       0765  2          STATUS;
;  167       0766  2
;  168       0767  2      CMD_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  169       0768  2      CMD_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
;  170       0769  2      CMD_DESC [DSC$A_POINTER] = 0;
;  171       0770  2      CMD_DESC [DSC$W_LENGTH] = 0;
;  172       0771  2      STRING_DESC (CMD_DESC, LENGTH, .STRING);    ! build descriptor
;  173       0772  2  !+
;  174       0773  2  ! The following call to the XLATE routine is a general routine call
;  175       0774  2  ! using the linkage defined as IOCALL.  A call is being made to the
;  176       0775  2  ! routine whose address is contained in the variable EDT$$A_XLATE_ROUT.
;  177       0776  2  ! The comtents of the variable EDT$$G_XLATE_ENV is first put into
;  178       0777  2  ! register 1 for the call.  (This is done to fill the requirement of
;  179       0778  2  ! callable EDT where the user can specify an XLATE routine and some
;  180       0779  2  ! high-level languages require an environment address to be passed in
;  181       0780  2  ! register 1.)  The remaing parameter in the list is the actual
;  182       0781  2  ! parameter passed to the XLATE routine.
;  183       0782  2  !-
;  184       0783  2      STATUS = IOCALL (.EDT$$A_XLATE_ROUT, .EDT$$G_XLATE_ENV, CMD_DESC);
;  185       0784  2  !+
;  186       0785  2  ! If the returned status is bad print the indicated message.
;  187       0786  2  !-
;  188       0787  2
;  189       0788  3      IF ( NOT .STATUS)
;  190       0789  2      THEN
;  191       0790  3          BEGIN
;  192       0791  3          EDT$$MSG_BELL (.STATUS);
;  193       0792  3          STR$FREE1_DX (CMD_DESC);
;  194       0793  3          RETURN (0);
;  195       0794  2          END;
;  196       0795  2
;  197       0796  2  !+
;  198       0797  2  ! Returned status is good.  If there is a command string returned
;  199       0798  2  ! execute it.
;  200       0799  2  ! -
;  201       0800  2
;  202       0801  3      IF (.CMD_DESC [DSC$W_LENGTH] GTRU 0)
;  203       0802  2      THEN
;  204       08C3  3          BEGIN
;  205       0804  3  !+
;  206       0805  3  ! Make sure there's enough room in the command buffer for the
;  207       0806  3  ! resulting command.
;  208       0807  3  !-
;  209       0808  3
;  210       0809  4          IF (.CMD_DESC [DSC$W_LENGTH] GTRU 256)
;  211       0810  3          THEN
;  212       0811  4              BEGIN
;  213       0812  4              EDT$$MSG_BELL (EDT$_COMEXHXLA);
;  214       0813  4              STR$FREE1_DX (CMD_DESC);
;  215          0814  4                 RETURN (0);
;  216          0815  3                 END;
;  217          0816  3
;  218          0817  3  !+
;  219          0818  3  ! Copy the returned string into 256 bytes of heap storage.
;  220          0819  3  !-
;  221          0820  3
;  222          0821  3             IF EDT$$ALO_HEAP (%REF (256), NEW_CMD_BUF)       !
;  223          0822  3             THEN
;  224          0823  3                 CH$MOVE (.CMD_DESC [DSC$W_LENGTH], .CMD_DESC [DSC$A_POINTER], .NEW_CMD_BUF)
;  225          0824  3             ELSE
;  226          0825  4                 BEGIN
;  227          0826  4                 EDT$$MSG_BELL (EDT$_INSMEM);
;  228          0827  4                 RETURN (0);
;  229          0828  3                 END;
;  230          0829  3
;  231          0830  3  !+
;  232          0831  3  ! Save the current command buffer pointers.
;  233          0832  3  !-
;  234          0833  3             SAVE_CMD_BUF = .EDT$$A_CMD_BUF;
;  235          0834  3             SAVE_CMD_END = .EDT$$A_CMD_END;
;  236          0835  3  !+
;  237          0836  3  ! Point to the new command buffer.
;  238          0837  3  !-
;  239          0838  3             EDT$$A_CMD_BUF = .NEW_CMD_BUF;
;  240          0839  3             EDT$$A_CMD_END = .CMD_DESC [DSC$W_LENGTH] + .EDT$$A_CMD_BUF;
;  241          0840  3  !+
;  242          0841  3  ! Now parse and execute the XLATE-returned string.
;  243          0842  3  !-
;  244          0843  3             STATUS = EDT$$CHM_PAREXE (1);
;  245          0844  3  !+
;  246          0845  3  ! Restore the command buffer.
;  247          0846  3  !-
;  248          0847  3             EDT$$A_CMD_BUF = .SAVE_CMD_BUF;
;  249          0848  3             EDT$$A_CMD_END = .SAVE_CMD_END;
;  250          0849  3  !+
;  251          0850  3  ! Free XLATE's command buffer.
;  252          0851  3  !-
;  253          0852  3             EDT$$DEA_HEAP (%REF (256), NEW_CMD_BUF);
;  254          0853  2             END;
;  255          0854  2
;  256          0855  2  !+
;  257          0856  2  ! We no longer need the returned string.
;  258          0857  2  !-
;  259          0858  2         STR$FREE1_DX (CMD_DESC);
;  260          0859  2         RETURN (1);
;  261          0860  1         END;                                        ! of routine EDT$$GET_XLATE


;                                                                            .TITLE  EDT$GETXLATE EDT$GETXLATE - Handles the nokeypa
                                                                                                  d XLATE comma
                                                                             .IDENT  \V04-000\

                                                                             .EXTRN  EDT$$CHM_PAREXE
                                                                             .EXTRN  EDT$$MSG_BELL, EDT$$ALO_HEAP
                                                                             .EXTRN  EDT$$DEA_HEAP, STR$FREE1_DX
                                                                             .EXTRN  EDT$$A_XLATE_ROUT
```

```
                                                        .EXTRN  EDT$$G_XLATE_ENV
                                                        .EXTRN  EDT$$T_CMD_BUF, EDT$$A_CMD_BUF
                                                        .EXTRN  EDT$$A_CMD_END, EDT$_INVSUBCOM
                                                        .EXTRN  EDT$_INSMEM, EDT$_COMFXHXLA
                                                        .EXTRN  EDT$_PASSTATUS, STR$COPY_R

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

                                        07FC 00000      .ENTRY  EDT$$GET_XLATE, Save R2,R3,R4,R5,R6,R7,R8,- ; 0702
                                                                R9,R10
                     5A 00000000G  00  9E 00002         MOVAB   STR$FREE1_DX, R10                          ;
                     59 00000000G  00  9E 00009         MOVAB   EDT$$MSG_BELL, R9                          ;
                     58 00000000G  00  9E 00010         MOVAB   EDT$$A_CMD_END, R8                         ;
                     57 00000000G  00  9E 00017         MOVAB   EDT$$A_CMD_BUF, R7                         ;
                     5E            10  C2 0001E         SUBL2   #16, SP                                    ;
              08     AE 020E0000   8F  D0 00021         MOVL    #34471936, CMD_DESC                        ; 0770
                            0C     AE  D4 00029         CLRL    CMD_DESC+4                                 ; 0769
                            04     AC  DD 0002C         PUSHL   STRING                                     ; 0771
                            08     AC  9F 0002F         PUSHAB  LENGTH                                     ;
                            10     AE  9F 00032         PUSHAB  CMD_DESC                                   ;
          00000000G  00            03  FB 00035         CALLS   #3, STR$COPY_R                             ;
                     50 00000000G  00  D0 0003C         MOVL    EDT$$A_XLATE_ROUT, R0                      ; 0783
                            08     AE  9F 00043         PUSHAB  CMD_DESC                                   ;
                     51 00000000G  00  D0 00046         MOVL    EDT$$G_XLATE_ENV, R1                       ;
                     60            01  FB 0004D         CALLS   #1, (R0)                                   ;
                     56            50  D0 00050         MOVL    R0, STATUS                                 ;
                     04            56  E8 00053         BLBS    STATUS, 1$                                 ; 0788
                                   56  DD 00056         PUSHL   STATUS                                     ; 0791
                                   13  11 00058         BRB     2$                                         ;
                            08     AE  B5 0005A 1$:     TSTW    CMD_DESC                                   ; 0801
                                   7A  13 0005D         BEQL    6$                                         ;
            0100     8F     08     AE  B1 0005F         CMPW    CMD_DESC, #256                             ; 0809
                                   11  1B 00065         BLEQU   3$                                         ;
                 00000000G         8F  DD 00067         PUSHL   #EDT$_COMEXHXLA                            ; 0812
                     69            01  FB 0006D 2$:     CALLS   #1, EDT$$MSG_BELL                          ;
                            08     AE  9F 00070         PUSHAB  CMD_DESC                                   ; 0813
                     6A            01  FB 00073         CALLS   #1, STR$FREE1_DX                           ;
                                   6B  11 00076         BRB     7$                                         ; 0814
                            04     AE  9F 00078 3$:     PUSHAB  NEW_CMD_BUF                                ; 0821
              04     AE   0100     8F  3C 0007B         MOVZWL  #256, 4(SP)                                ;
                            04     AE  9F 00081         PUSHAB  4(SP)                                      ;
          00000000G  00            02  FB 00084         CALLS   #2, EDT$$ALO_HEAP                          ;
                     09            50  E9 0008B         BLBC    R0, 4$                                     ;
   04    BE     0C   BE     08     AE  28 0008E         MOVC3   CMD_DESC, @CMD_DESC+4, @NEW_CMD_BUF        ; 0823
                                   0B  11 00095         BRB     5$                                         ;
                 00000000G         8F  DD 00097 4$:     PUSHL   #EDT$_INSMEM                               ; 0826
                     69            01  FB 0009D         CALLS   #1, EDT$$MSG_BELL                          ;
                                   41  11 000A0         BRB     7$                                         ; 0827
                     53            67  D0 000A2 5$:     MOVL    EDT$$A_CMD_BUF, SAVE_CMD_BUF               ; 0833
                     52            68  D0 000A5         MOVL    EDT$$A_CMD_END, SAVE_CMD_END               ; 0834
                     67     04     AE  D0 000A8         MOVL    NEW_CMD_BUF, EDT$$A_CMD_BUF                ; 0838
                     50     08     AE  3C 000AC         MOVZWL  CMD_DESC, R0                               ; 0839
              68     50            67  C1 000B0         ADDL3   EDT$$A_CMD_BUF, R0, EDT$$A_CMD_END         ;
                                   01  DD 000B4         PUSHL   #1                                         ; 0843
          00000000G  00            01  FB 000B6         CALLS   #1, EDT$$CHM_PAREXE                        ;
                     56            50  D0 000BD         MOVL    R0, STATUS                                 ;
                     67            53  D0 000C0         MOVL    SAVE_CMD_BUF, EDT$$A_CMD_BUF               ; 0847
```

```
                                   68             52  D0 000C3          MOVL     SAVE_CMD_END, EDT$$A_CMD_END                    ; 0848
                                           04     AE  9F 000C6          PUSHAB   NEW_CMD_BUF                                     ; 0852
                              04   AE    0100     8F  3C 000C9          MOVZWL   #256, 4(SP)                                     ;
                                           04     AE  9F 000CF          PUSHAB   4(SP)                                           ;
                      00000000G    00             02  FB 000D2          CALLS    #2, EDT$$DEA_HEAP                               ;
                                           08     AE  9F 000D9 6$:      PUSHAB   CMD_DESC                                        ; 0858
                                   6A             01  FB 000DC          CALLS    #1, STR$FREE1_DX                                ;
                                   50             01  D0 000DF          MOVL     #1, R0                                          ; 0859
                                                      04 000E2          RET                                                      ;
                                                  50  D4 000E3 7$:      CLRL     R0                                              ; 0860
                                                      04 000E5          RET                                                      ;

; Routine Size:  230 bytes,    Routine Base:  _EDT$CODE + 0000


;  262           0861  1
;  263           0862  1 !<BLF/PAGE>
;  265          0863  1 END                                                  ! of module EDT$GETXLATE
;  266          0864  1
;  267          0865  0 ELUDOM



;                                    PSECT SUMMARY
;
;       Name                       Bytes                      Attributes
;
;    _EDT$CODE                          230  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                            Library Statistics
;
;                                            -------- Symbols --------      Pages       Processing
;       File                                 Total    Loaded   Percent      Mapped      Time
;
;    _$255$DUA28:[EDT.SRC]EDT.L32;1            377         2         0         40         00:00.2
;    _$255$DUA28:[EDT.SRC]PSECTS.L32;1           2         1        50          7         00:00.1
;    _$255$DUA28:[SYSLIB]STARLET.L32;1        9776         6         0        581         00:04.0



;                                    COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS$:GETXLATE/OBJ=OBJ$:GETXLATE MSRC$:GETXLATE.B32/UPDATE=(ENH$:GETX
;       LATE)

; Size:          230 code + 0 data bytes
; Run Time:         00:20.3
; Elapsed Time:     00:26.7
; Lines/CPU Min:     2554
; Lexemes/CPU-Min:  7937
; Memory Used:  111 pages
; Compilation Complete
```

0134 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY