C 10

```
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
```

```
**FILE**ID**LQUERY                                       C  7




LL            QQQQQQ    UU      UU  EEEEEEEEEE  RRRRRRRR    YY      YY
LL            QQQQQQ    UU      UU  EEEEEEEEEE  RRRRRRRR    YY      YY
LL          QQ      QQ  UU      UU  EE          RR      RR  YY      YY
LL          QQ      QQ  UU      UU  EE          RR      RR  YY      YY
LL          QQ      QQ  UU      UU  EE          RR      RR    YY  YY
LL          QQ      QQ  UU      UU  EE          RR      RR    YY  YY
LL          QQ      QQ  UU      UU  EEEEEEEE    RRRRRRRR        YY
LL          QQ      QQ  UU      UU  EEEEEEEE    RRRRRRRR        YY
LL          QQ  QQ  QQ  UU      UU  EE          RR  RR          YY
LL          QQ  QQ  QQ  UU      UU  EE          RR  RR          YY
LL          QQ    QQ    UU      UU  EE          RR    RR        YY          ....
LL          QQ    QQ    UU      UU  EE          RR    RR        YY          ....
LLLLLLLLLL    QQQQ  QQ  UUUUUUUUUU  EEEEEEEEEE  RR      RR      YY          ....
LLLLLLLLLL    QQQQ  QQ  UUUUUUUUUU  EEEEEEEEEE  RR      RR      YY          ....

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
```

```
0 %TITLE 'EDT$LQUERY - do /QUERY processing'
0 MODULE EDT$LQUERY (                                   ! Do /QUERY processing
0               IDENT = 'V04-000'                               ! File: LQUERY.BLI Edit: REM1013
0               ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     EDT -- The DEC Standard Editor
1 !
1 ! ABSTRACT:
1 !
1 !       This module handles the user interface of the /QUERY
1 !       option on various line mode commands, such as SUBSTITUTE.
1 !
1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
1 !
1 ! AUTHOR: Bob Kushlis, CREATION DATE: February 3, 1978
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  DJS 02-FEB-1981.  This module was created by
1 !       extracting the routine EDT$$PROC_QRYQAL  from the module EXEC.BLI.
1 ! 1-002 - Regularize headers.  JBS 20-Mar-1981
1 ! 1-003 - Use the ASSERT macro and fix up some comments mangled by converting QUERY
1 !           to EDT$$PROC_QRYQAL .  JBS 01-Jun-1981
1 ! 1-004 - Revise journaling.  JBS 22-Jun-1981
1 ! 1-005 - Use new message codes.  JBS 04-Aug-1981
1 ! 1-006 - Make empty responses illegal.  JBS 16-Aug-1981
1 ! 1-007 - Prompt from a global.  JBS 23-Oct-1981
1 ! 1-008 - Use heap storage instead of the stack to hold the constructed
1 !           line.  JBS 27-Jan-1982
1 ! 1-009 - Add a missing dot.  JBS 28-Jan-1982
1 ! 1-010 - Add EDT$$G_JOU_VALID.  JBS 09-Apr-1982
```

```
;       0058  1 ! 1-011 - Refresh the screen after a query.  JBS 05-Jul-1982
;       0059  1 ! 1-012 - If the journal file ends at a query response, treat the response
;       0060  1 !          as "Q", to return to command level.  Note that the "Q" must
;       0061  1 !          be journaled.  JBS 10-Jul-1982
;       0062  1 ! 1-013 - Added logic to maintain EDT$$G_TIN_OBUFPOS durring /RECOVERY mode.
;       0063  1 !          REM 10-Oct-1983
;       0064  1 !--
;       0065  1
```

```
0066  1 %SBTTL 'Declarations'
0067  1 !
0068  1 ! TABLE OF CONTENTS:
0069  1 !
0070  1
0071  1 REQUIRE 'EDTSRC:TRAROUNAM';
0510  1
0511  1 FORWARD ROUTINE
0512  1     EDT$$PROC_QRYQAL;                          ! Process the EDT$$PROC_QRYQAL  qualifier
0513  1
0514  1 !
0515  1 ! INCLUDE FILES:
0516  1 !
0517  1
0518  1 REQUIRE 'EDTSRC:EDTREQ';
0653  1
0654  1 !
0655  1 ! MACROS:
0656  1 !
0657  1 !       NONE
0658  1 !
0659  1 ! EQUATED SYMBOLS:
0660  1 !
0661  1
0662  1 LITERAL
0663  1     QUERY_LEN = 80,
0664  1     LINE_LEN = LIN_FIXED_SIZE + 255;
0665  1
0666  1 !
0667  1 ! OWN STORAGE:
0668  1 !
0669  1 !       NONE
0670  1 !
0671  1 ! EXTERNAL REFERENCES:
0672  1 !
0673  1 !       In the routine
```

```
1 %SBTTL 'EDT$$PROC_QRYQAL  - do /QUERY processing'
1
1 GLOBAL ROUTINE EDT$$PROC_QRYQAL (                      ! Do /QUERY processing
1     WPTR,                                               ! Used to reconstruct line for SUBSTITUTE
1     WEND                                                ! Used to reconstruct line for SUBSTITUTE
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       EDT$$PROC_QRYQAL  processing routine.  This routine is called before operating on
1 !       each line by those commands which can take a /QUERY qualifier.
1 !
1 !       The options bits are checked to see if the /QUERY qualifier was used.
1 !       If not the routine returns success immediately.  If it was specified,
1 !       then the user is prompted for verification.  The actions for each
1 !       possible answer are:
1 !
1 !               Y - Return a 1 to indcate operation should be performed.
1 !               N - Return a 0 to indicate operation should not be performed.
1 !               Q - The global flag EDT$$G_EXE_QRYQUIT  is set to stop further processing.
1 !               A - The /QUERY option bit is cleared so no more queries are done.
1 !
1 !       If the answer is not one of the above, then a message is displayed and the
1 !       user is prompted again.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !  WPTR                 Used to reconstruct line for SUBSTITUTE, 0 otherwise
1 !
1 !  WEND                 Used to reconstruct line for SUBSTITUTE, 0 otherwise
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       EDT$$G_RCOV_MOD
1 !       EDT$$T_LN_BUF
1 !       EDT$$G_LN_LEN
1 !       EDT$$A_WK_LN
1 !       EDT$$G_EXE_QRYQUIT
1 !       EDT$$G_EXE_SBITS
1 !       EDT$$T_PMT_QUERY
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       EDT$$G_JOU_VALID
1 !       EDT$$G_TIN_OBUFPOS
1 !
1 ! ROUTINE VALUE:
1 !
1 !       1 = do the operation
1 !       0 = don't do the operation
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
```

```
1
2        BEGIN
2
2        EXTERNAL ROUTINE
2            EDT$$FMT_MSG,                             ! Place the text of a message in the format buffer
2            EDT$$RD_CMDLN,                            ! Read a command line
2            EDT$$RD_JOUTXT,                           ! Read a text record from the journal file
2            EDT$$TI_FLUSHJOUFI : NOVALUE,             ! Write current journal buffer on journal file
2            EDT$$TI_BUFCH : NOVALUE,                  ! Store a character in the journal buffer
2            EDT$$CNV_UPC,                             ! Convert to upper case
2            EDT$$TY_CURLN,                            ! Display the current line
2            EDT$$ALO_HEAP,                            ! Allocate heap storage
2            EDT$$DEA_HEAP : NOVALUE,                  ! Deallocate heap storage
2            EDT$$FMT_CRLF;                            ! Terminate a formatted line
2
2        EXTERNAL
2            EDT$$G_RCOV_MOD,
2            EDT$$T_LN_BUF : VECTOR [255, BYTE],
2            EDT$$G_LN_LEN,
2            EDT$$A_WK_LN : REF LIN_BLOCK,
2            EDT$$G_EXE_QRYQUIT,                       ! Quit flag  for /QUERY operations.
2            EDT$$G_EXE_SBITS,                         ! The options switches.
2            EDT$$T_PMT_QUERY : VECTOR [, BYTE],       ! Counted ASCII string for /QUERY prompt
2            EDT$$G_JOU_VALID,                         ! 1 = journal record is valid
2            EDT$$G_TIN_OBUFPOS,                       ! Position in journal output buffer
2            EDT$$G_FMT_BOT;                           ! We are printing at the bottom of the screen
2
2        MESSAGES ((PLSANSYNQ, INSMEM));
2
2        LOCAL
2            QUERY_RESP : REF BLOCK [CH$ALLOCATION (QUERY_LEN)],
2            QUERY_RESP_COMPLETE,                      ! 1 = response is complete
2            RET_VAL,                                  ! Return value
2            LEN,
2            SW_LINE,                                  ! Save EDT$$A_WK_LN
2            IN,                                       ! Input char ptr
2            OUT,                                      ! Output char ptr
2            WLEN,                                     ! Length of rest of line
2            L_WPTR,                                   ! Local copy of WPTR
2            T_LINE : REF LIN_BLOCK;                   ! Current line with substitutions
2
2     !+
2     ! Check for the /QUERY bit.  If it is clear then return 1.
2     !-
2
2        IF .EDT$$G_EXE_SBITS<OPB_QUERY>
2        THEN
3            BEGIN
3     !+
3     ! Display the line so the user can see what he is verifying.
3     !-
3
4            IF (.WPTR EQL 0)
3            THEN
3                EDT$$TY_CURLN ()
3            ELSE
3     !+
3 ! During a SUBSTITUTE command, the current line is in various
3 ! pieces which are here reconstructed so that any substitution
3 ! already made on the line are shown.
3 !-
4               BEGIN
4 !+
4 ! Allocate enough space for a maximum-length line.
4 !-
4
4               IF EDT$$ALO_HEAP (%REF (LINE_LEN), T_LINE)
4               THEN
5                   BEGIN
5 !+
5 ! Initialize the description for the line to be constructed.
5 !-
5                   EDT$$CPY_MEM (LIN_FIXED_SIZE, .EDT$$A_WK_LN, .T_LINE);
5 !+
5 ! Copy the line up to the last substitution.
5 !-
5                   IN = CH$PTR (EDT$$T_LN_BUF);
5                   OUT = CH$PTR (T_LINE [LIN_TEXT]);
5
5                   DECR I FROM .EDT$$G_LN_LEN - 1 TO 0 DO
5                       CH$WCHAR_A (CH$RCHAR_A (IN), OUT);
5
5 !+
5 ! Copy the current line from the last match to the end.
5 !-
5                   WLEN = CH$DIFF (.WEND, .WPTR);
5
5                   IF ((.EDT$$G_LN_LEN + .WLEN) GTR 255) THEN WLEN = MAX (0, 255 - .EDT$$G_LN_LEN);
5
5                   L_WPTR = .WPTR;
5
5                   DECR I FROM .WLEN - 1 TO 0 DO
5                       CH$WCHAR_A (CH$RCHAR_A (L_WPTR), OUT);
5
5 !+
5 ! Fixup the description of the fake current line.
5 !-
5                   T_LINE [LIN_LENGTH] = .EDT$$G_LN_LEN + .WLEN;
5 !+
5 ! Type the line.
5 !-
5                   SW_LINE = .EDT$$A_WK_LN;          ! Save the current line description
5                   EDT$$A_WK_LN = .T_LINE;           ! Make the constructed line the current one
5                   EDT$$TY_CURLN ();                 ! Format and output this line
5                   EDT$$A_WK_LN = .SW_LINE;          ! Restore the current line description
5 !+
5 ! Deallocate the heap storage used to hold the line.
5 !-
5                   EDT$$DEA_HEAP (%REF (LINE_LEN), T_LINE);
5                   END
4               ELSE
5                   BEGIN
5 !+
5 ! There is not enough heap storage to print the line.  Don't do this operation
```

```
5 ! and stop the whole command.  Also, give an appropriate error message.
5 !-
5                   EDT$$FMT_MSG (EDT$_INSMEM);      ! Give an appropriate error message
5                   EDT$$G_EXE_QRYQUIT = 1;          ! Stop the command
5                   RETURN (0);                      ! Don't do this substitution
4                   END;
4
3               END;
3
3 !+
3 ! Allocate space to hold the response to the query.
3 !-
3
4           IF ( NOT EDT$$ALO_HEAP (%REF (QUERY_LEN), QUERY_RESP))
3           THEN
4               BEGIN
4 !+
4 ! There is not enough storage to accept the response.  Don't do this
4 ! operation and stop the whole command.  Also, give an appropriate error message.
4 !-
4               EDT$$FMT_MSG (EDT$_INSMEM);          ! Give an appropriate message
4               EDT$$G_EXE_QRYQUIT = 1;              ! Stop the command
4               RETURN (0);                          ! Don't do this substitution
3               END;
3
3           QUERY_RESP_COMPLETE = 0;
3
3           WHILE ( NOT .QUERY_RESP_COMPLETE) DO
4               BEGIN
4 !+
4 ! Get the line from either the terminal or the journal file.
4 !-
4
4               IF .EDT$$G_RCOV_MOD
4               THEN
5                   BEGIN
5
6                   IF ( NOT EDT$$RD_JOUTXT (.QUERY_RESP, LEN))
5                   THEN
6                       BEGIN
6 !+
6 ! We have reached the end of the journal file.  Fake a "Q" response so that
6 ! we will terminate this command without making any more changes to the buffer.
6 !-
6                       LEN = 1;
6                       CH$WCHAR (%C'Q', .QUERY_RESP);
6 !+
6 ! We must journal the fake response, in case we do another /RECOVER during this session.
6 !-
6                       EDT$$TI_BUFCH (CH$RCHAR (.QUERY_RESP));
6                       EDT$$G_JOU_VALID = 1
6                       END
5                   ELSE
5                       EDT$$G_TIN_OBUFPOS = MIN(.len, 1)
5                   END
4               ELSE
5                   BEGIN
```

```
:   332         0902  5 !+
:   333         0903  5 ! Make sure the journal buffer has been written to the journal file,
:   334         0904  5 ! since we are about to wait for terminal input.
:   335         0905  5 !-
:   336         0906  5                 EDT$$TI_FLUSHJOUFI (%C'T');
:   337         0907  5 !+
:   338         0908  5 ! If all the text is being concentrated at the bottom of the screen, then be sure we are prompting
:   339         0909  5 ! on a blank line.
:   340         0910  5 !-
:   341         0911  5
:   342         0912  5                 IF .EDT$$G_FMT_BOT THEN EDT$$FMT_CRLF ();
:   343         0913  5
:   344         0914  5                 EDT$$RD_CMDLN (EDT$$T_PMT_QUERY [1], .EDT$$T_PMT_QUERY [0], .QUERY_RESP, LEN, QUERY_LEN);
:   345         0915  5 !+
:   346         0916  5 ! Make sure the response is journaled.  Only the first character of the response is journaled
:   347         0917  5 ! because only the first character is important.
:   348         0918  5 !-
:   349         0919  5
:   350         0920  5                 IF (.LEN GEQ 1) THEN EDT$$TI_BUFCH (CH$RCHAR (.QUERY_RESP));
:   351         0921  5
:   352         0922  5                 EDT$$G_JOU_VALID = 1;
:   353         0923  4                 END;
:   354         0924  4
:   355         0925  4 !+
:   356         0926  4 ! Check out the answer.
:   357         0927  4 !-
:   358         0928  4
:   359         0929  5             IF (.LEN LSS 1)
:   360         0930  4             THEN
:   361         0931  4                 EDT$$FMT_MSG (EDT$_PLSANSYNQ)
:   362         0932  4             ELSE
:   363         0933  5                 BEGIN
:   364         0934  5                 EDT$$CNV_UPC (.QUERY_RESP, 1);
:   365         0935  5
:   366         0936  5                 SELECTONE CH$RCHAR (.QUERY_RESP) OF
:   367         0937  5                     SET
:   368         0938  5
:   369         0939  5                     [%C'Y'] :
:   370         0940  6                         BEGIN
:   371         0941  6                         RET_VAL = 1;
:   372         0942  6                         QUERY_RESP_COMPLETE = 1;
:   373         0943  5                         END;
:   374         0944  5
:   375         0945  5                     [%C'N'] :
:   376         0946  6                         BEGIN
:   377         0947  6                         RET_VAL = 0;
:   378         0948  6                         QUERY_RESP_COMPLETE = 1;
:   379         0949  5                         END;
:   380         0950  5
:   381         0951  5                     [%C'A'] :
:   382         0952  6                         BEGIN
:   383         0953  6                         EDT$$G_EXE_SBITS<OPB_QUERY> = 0;
:   384         0954  6                         RET_VAL = 1;
:   385         0955  6                         QUERY_RESP_COMPLETE = 1;
:   386         0956  5                         END;
:   387         0957  5
:   388         0958  5                     [%C'Q'] :
```

```
;  389          0959 6                       BEGIN
;  390          0960 6                       EDT$$G_EXE_QRYQUIT = 1;
;  391          0961 6                       RET_VAL = 0;
;  392          0962 6                       QUERY_RESP_COMPLETE = 1;
;  393          0963 5                       END;
;  394          0964 5
;  395          0965 5                   [OTHERWISE] :
;  396          0966 5                       EDT$$FMT_MSG (EDT$_PLSANSYNQ);
;  397          0967 5                   TES;
;  398          0968 5
;  399          0969 4               END;
;  400          0970 4
;  401          0971 3           END;
;  402          0972 3
;  403          0973 3 !+
;  404          0974 3 ! Come here when the query response is complete.  RET_VAL contains the
;  405          0975 3 ! value to return.  Deallocate the heap storage used to hold the responses
;  406          0976 3 ! to the query.
;  407          0977 3 !-
;  408          0978 3       EDT$$DEA_HEAP (%REF (QUERY_LEN), QUERY_RESP);
;  409          0979 3       END
;  410          0980 2   ELSE
;  411          0981 2       RET_VAL = 1;
;  412          0982 2
;  413          0983 2   RETURN (.RET_VAL);
;  414          0984 1   END;                                      ! of routine EDT$$PROC_QRYQAL


                                                           .TITLE  EDT$LQUERY EDT$LQUERY - do /QUERY processing
                                                           .IDENT  \V04-000\

                                                           .EXTRN  EDT$$FMT_MSG, EDT$$RD_CMDLN
                                                           .EXTRN  EDT$$RD_JOUTXT, EDT$$TI_FLUSHJOUFI
                                                           .EXTRN  EDT$$TI_BUFCH, EDT$$CNV_UPC
                                                           .EXTRN  EDT$$TY_CURLN, EDT$$ALO_HEAP
                                                           .EXTRN  EDT$$DEA_HEAP, EDT$$FMT_CRLF
                                                           .EXTRN  EDT$$G_RCOV_MOD
                                                           .EXTRN  EDT$$T_LN_BUF, EDT$$G_LN_LEN
                                                           .EXTRN  EDT$$A_WK_LN, EDT$$G_EXE_QRYQUIT
                                                           .EXTRN  EDT$$G_EXE_SBITS
                                                           .EXTRN  EDT$$T_PMT_QUERY
                                                           .EXTRN  EDT$$G_JOU_VALID
                                                           .EXTRN  EDT$$G_TIN_OBUFPOS
                                                           .EXTRN  EDT$$G_FMT_BOT, EDT$_PLSANSYNQ
                                                           .EXTRN  EDT$_INSMEM

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

                                            0FFC 00000     .ENTRY  EDT$$PROC_QRYQAL, Save R2,R3,R4,R5,R6,R7,-  ; 0676
                                                                   R8,R9,R10,R11                               ;
                        5B 00000000G  00   9E 00002        MOVAB   EDT$$DEA_HEAP, R11                          ;
                        5A 00000000G  00   9E 00009        MOVAB   EDT$$ALO_HEAP, R10                          ;
                        59 00000000G  00   9E 00010        MOVAB   EDT$$TY_CURLN, R9                           ;
                        58 00000000G  00   9E 00017        MOVAB   EDT$$A_WK_LN, R8                            ;
                        5E            10   C2 0001E        SUBL2   #16, SP                                     ;
           03 00000000G  00            01   E0 00021        BBS     #1, EDT$$G_EXE_SBITS, 1$                    ; 0776
                                      01B0   31 00029        BRW     30$                                         ;
```

```
                              56          04  AC  D0 0002C 1$:     MOVL     WPTR, R6                                         ; 0783
                                              06  12 00030         BNEQ     2$                                               ;
                              69              00  FB 00032         CALLS    #0, EDT$$TY_CURLN                                ; 0785
                                            0088  31 00035         BRW      10$                                              ;
                                          04  AE  9F 00038 2$:     PUSHAB   T_LINE                                           ; 0797
                       04     AE        0106  8F  3C 0003B         MOVZWL   #262, 4(SP)                                      ;
                                          04  AE  9F 00041         PUSHAB   4(SP)                                            ;
                              6A              02  FB 00044         CALLS    #2, EDT$$ALO_HEAP                                ;
                              03              50  E8 00047         BLBS     R0, 3$                                           ;
                                            0087  31 0004A         BRW      11$                                              ;
                              57              68  D0 0004D 3$:     MOVL     EDT$$A_WK_LN, R7                                 ; 0803
         04   BE              67              07  28 00050         MOVC3    #7, (R7), @T_LINE                                ;
                              51  00000000G   00  9E 00055         MOVAB    EDT$$T_LN_BUF, IN                                ; 0807
              52       04     AE              07  C1 0005C         ADDL3    #7, T_LINE, OUT                                  ; 0808
                              53  00000000G   00  D0 00061         MOVL     EDT$$G_LN_LEN, R3                                ; 0810
                              50              53  D0 00068         MOVL     R3, I                                            ; 0811
                                              03  11 0006B         BRB      5$                                               ;
                              82              81  90 0006D 4$:     MOVB     (IN)+, (OUT)+                                    ;
                              FA              50  F4 00070 5$:     SOBGEQ   I, 4$                                            ;
              50       08     AC              56  C3 00073         SUBL3    R6, WEND, WLEN                                   ; 0816
              51              53              50  C1 00078         ADDL3    WLEN, R3, R1                                     ; 0818
                 000000FF     8F              51  D1 0007C         CMPL     R1, #255                                         ;
                                              0F  15 00083         BLEQ     7$                                               ;
              51 000000FF     8F              53  C3 00085         SUBL3    R3, #255, R1                                     ;
                                              02  18 0008D         BGEQ     6$                                               ;
                                              51  D4 0008F         CLRL     R1                                               ;
                              50              51  D0 00091 6$:     MOVL     R1, WLEN                                         ;
                              54              56  D0 00094 7$:     MOVL     R6, L_WPTR                                       ; 0820
                              51              50  D0 00097         MOVL     WLEN, I                                          ; 0823
                                              03  11 0009A         BRB      9$                                               ;
                              82              84  90 0009C 8$:     MOVB     (L_WPTR)+, (OUT)+                                ;
                              FA              51  F4 0009F 9$:     SOBGEQ   I, 8$                                            ;
         04   BE              53              50  81 000A2         ADDB3    WLEN, R3, @T_LINE                                ; 0828
                              52              57  D0 000A7         MOVL     R7, SW_LINE                                      ; 0832
                              68          04  AE  D0 000AA         MOVL     T_LINE, EDT$$A_WK_LN                             ; 0833
                              69              00  FB 000AE         CALLS    #0, EDT$$TY_CURLN                                ; 0834
                              68              52  D0 000B1         MOVL     SW_LINE, EDT$$A_WK_LN                            ; 0835
                                          04  AE  9F 000B4         PUSHAB   T_LINE                                           ; 0839
                       04     AE        0106  8F  3C 000B7         MOVZWL   #262, 4(SP)                                      ;
                                          04  AE  9F 000BD         PUSHAB   4(SP)                                            ;
                              6B              02  FB 000C0         CALLS    #2, EDT$$DEA_HEAP                                ;
                                          0C  AE  9F 000C3 10$:    PUSHAB   QUERY_RESP                                       ; 0858
                       04     AE          50  8F  9A 000C6         MOVZBL   #80, 4(SP)                                       ;
                                          04  AE  9F 000CB         PUSHAB   4(SP)                                            ;
                              6A              02  FB 000CE         CALLS    #2, EDT$$ALO_HEAP                                ;
                              17              50  E8 000D1         BLBS     R0, 12$                                          ;
                                  00000000G   8F  DD 000D4 11$:    PUSHL    #EDT$_INSMEM                                     ; 0865
                 00000000G    00              01  FB 000DA         CALLS    #1, EDT$$FMT_MSG                                 ;
                 00000000G    00              01  D0 000E1         MOVL     #1, EDT$$G_EXE_QRYQUIT                           ; 0866
                                            00F8  31 000E8         BRW      32$                                              ; 0867
                                              53  D4 000EB 12$:    CLRL     QUERY_RESP_COMPLETE                              ; 0870
                              52          0C  AE  D0 000ED         MOVL     QUERY_RESP, R2                                   ; 0882
                              03              53  E9 000F1 13$:    BLBC     QUERY_RESP_COMPLETE, 14$                         ; 0872
                                            00D5  31 000F4         BRW      29$                                              ;
                              2E  00000000G   00  E9 000F7 14$:    BLBC     EDT$$G_RCOV_MOD, 17$                             ; 0878
                                          08  AE  9F 000FE         PUSHAB   LEN                                              ; 0882
                                              52  DD 00101         PUSHL    R2                                               ;
```

```
  00000000G  00            02  FB 00103          CALLS    #2, EDT$$RD_JOUTXT                    ;
             0A            50  E8 0010A          BLBS     R0, 15$                               ;
         08  AE            01  D0 0010D          MOVL     #1, LEN                               ; 0889
             62       51   8F  90 00111          MOVB     #81, (R2)                             ; 0890
                           50  11 00115          BRB      19$                                   ; 0894
             50       08   AE  D0 00117 15$:     MOVL     LEN, R0                               ; 0898
             01            50  D1 0011B          CMPL     R0, #1                                ;
                           03  15 0011E          BLEQ     16$                                   ;
             50            01  D0 00120          MOVL     #1, R0                                ;
  00000000G  00            50  D0 00123 16$:     MOVL     R0, EDT$$G_TIN_OBUFPOS                ;
                           4C  11 0012A          BRB      21$                                   ; 0880
             7E       54   8F  9A 0012C 17$:     MOVZBL   #84, -(SP)                            ; 0906
  00000000G  00            01  FB 00130          CALLS    #1, EDT$$TI_FLUSHJOUFI                ;
             07  00000000G 00  E9 00137          BLBC     EDT$$G_FMT_BOT, 18$                   ; 0912
  00000000G  00            00  FB 0013E          CALLS    #0, EDT$$FMT_CRLF                     ;
             7E       50   8F  9A 00145 18$:     MOVZBL   #80, -(SP)                            ; 0914
                      0C   AE  9F 00149          PUSHAB   LEN                                   ;
                           52  DD 0014C          PUSHL    R2                                    ;
             7E  00000000G 00  9A 0014E          MOVZBL   EDT$$T_PMT_QUERY, -(SP)               ;
                 00000000G 00  9F 00155          PUSHAB   EDT$$T_PMT_QUERY+1                    ;
  00000000G  00            05  FB 0015B          CALLS    #5, EDT$$RD_CMDLN                     ;
                      08   AE  D5 00162          TSTL     LEN                                   ; 0920
                           0A  15 00165          BLEQ     20$                                   ;
             7E            62  9A 00167 19$:     MOVZBL   (R2), -(SP)                           ;
  00000000G  00            01  FB 0016A          CALLS    #1, EDT$$TI_BUFCH                     ;
  00000000G  00            01  D0 00171 20$:     MOVL     #1, EDT$$G_JOU_VALID                  ; 0922
                      08   AE  D5 00178 21$:     TSTL     LEN                                   ; 0929
                           3F  15 0017B          BLEQ     27$                                   ;
                           01  DD 0017D          PUSHL    #1                                    ; 0934
                           52  DD 0017F          PUSHL    R2                                    ;
  00000000G  00            02  FB 00181          CALLS    #2, EDT$$CNV_UPC                      ;
         59  8F            62  91 00188          CMPB     (R2), #89                             ; 0939
                           05  12 0018C          BNEQ     23$                                   ;
             54            01  D0 0018E 22$:     MOVL     #1, RET_VAL                           ; 0941
                           24  11 00191          BRB      26$                                   ; 0942
         4E  8F            62  91 00193 23$:     CMPB     (R2), #78                             ; 0945
                           1C  13 00197          BEQL     25$                                   ;
         41  8F            62  91 00199          CMPB     (R2), #65                             ; 0951
                           09  12 0019D          BNEQ     24$                                   ;
  00000000G  00            02  8A 0019F          BICB2    #2, EDT$$G_EXE_SBITS                  ; 0953
                           E6  11 001A6          BRB      22$                                   ; 0954
         51  8F            62  91 001A8 24$:     CMPB     (R2), #81                             ; 0958
                           0E  12 001AC          BNEQ     27$                                   ;
  00000000G  00            01  D0 001AE          MOVL     #1, EDT$$G_EXE_QRYQUIT                ; 0960
                           54  D4 001B5 25$:     CLRL     RET_VAL                               ; 0961
             53            01  D0 001B7 26$:     MOVL     #1, QUERY_RESP_COMPLETE               ; 0962
                           0D  11 001BA          BRB      28$                                   ; 0936
                 00000000G 8F  DD 001BC 27$:     PUSHL    #EDT$_PLSANSYNQ                       ; 0966
  00000000G  00            01  FB 001C2          CALLS    #1, EDT$$FMT_MSG                      ;
                      FF25  31 001C9 28$:     BRW      13$                                   ; 0872
                      0C   AE  9F 001CC 29$:     PUSHAB   QUERY_RESP                            ; 0978
         04  AE       50   8F  9A 001CF          MOVZBL   #80, 4(SP)                            ;
                      04   AE  9F 001D4          PUSHAB   4(SP)                                 ;
             6B            02  FB 001D7          CALLS    #2, EDT$$DEA_HEAP                     ;
                           03  11 001DA          BRB      31$                                   ; 0776
             54            01  D0 001DC 30$:     MOVL     #1, RET_VAL                           ; 0981
             50            54  D0 001DF 31$:     MOVL     RET_VAL, R0                           ; 0983
```

```
                                           04 001E2          RET                                          ;
                                       50  D4 001E3 32$:     CLRL     R0                                  ; 0984
                                           04 001E5          RET                                          ;

; Routine Size:  486 bytes,    Routine Base:  _EDT$CODE + 0000


;  415          0985  1
;  416          0986  1 !<BLF/PAGE>
```

```
:  418          0987  1 END                                             ! of module EDT$LQUERY
:  419          0988  1
:  420          0989  0 ELUDOM



:                                PSECT SUMMARY
:
:       Name                    Bytes                   Attributes
:
:  _EDT$CODE                        486  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        10         2         40        00:00.2
:  _$255$DUA28:[EDT.SRC]PSECTS.L32;1        2         1        50          7        00:00.1



:                                COMMAND QUALIFIERS

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

; Size:          486 code + 0 data bytes
; Run Time:         00:22.7
; Elapsed Time:     00:27.2
; Lines/CPU Min:     2609
; Lexemes/CPU-Min:   8160
; Memory Used:  152 pages
; Compilation Complete
```

0136 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY