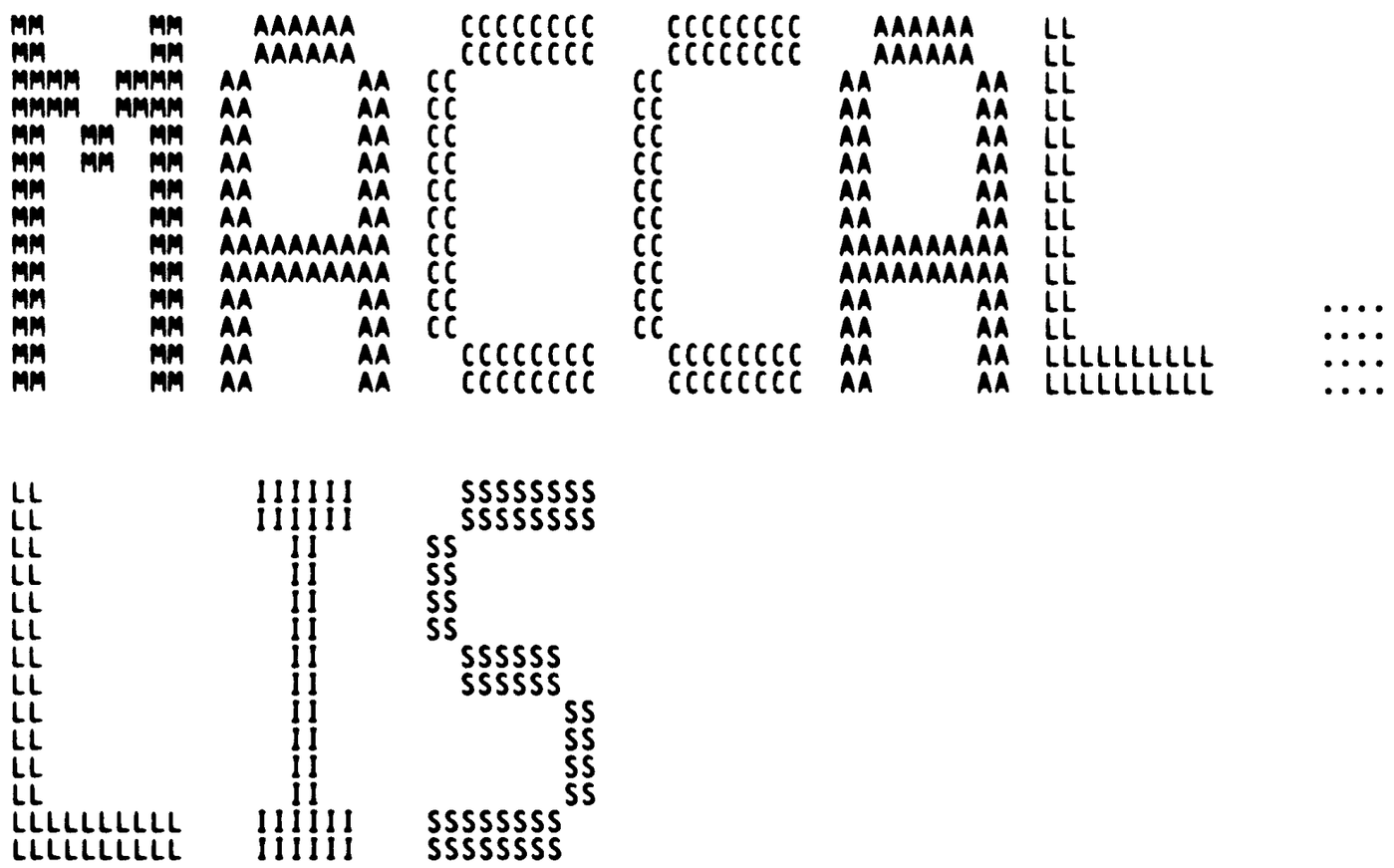
```
EEEEEEEEEEEEEEE  DDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEE              DDD         DDD       TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEE              DDD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

FILEID**MACCAL



```
;  1          0001  0 %TITLE 'EDT$MACCAL - macro call'
;  2          0002  0 MODULE EDT$MACCAL (                     ! Macro call
;  3          0003  0                 IDENT = 'V04-000'               ! File: MACCAL.BLI Edit: JBS1008
;  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 !       Macro call
; 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: 6-AUG-1979
; 41          0041  1 !
; 42          0042  1 ! MODIFIED BY:
; 43          0043  1 !
; 44          0044  1 ! 1-001 - Original.  DJS 18-FEB-1981.  This module was created by
; 45          0045  1 !       extracting routine EDT$$MAC_CALL  from module EDTCTR.
; 46          0046  1 ! 1-002 - Regularize headers and fix file and module name.  JBS 04-Mar-1981
; 47          0047  1 ! 1-003 - Remove L_LINE.  JBS 01-Oct-1981
; 48          0048  1 ! 1-004 - Replace C_LINE. TMV 7-Dec-81
; 49          0049  1 ! 1-005 - Add an entry point so that EDT$$EXE_CMD_NOOVERLAY can
; 50          0050  1 !         get this mdoule back into memory after a macro has been executed.  JBS 10-Mar-1982
; 51          0051  1 ! 1-006 - Save and restore the command line over a macro call, so another
; 52          0052  1 !         command can be after the macro name.  JBS 03-Jun-1982
; 53          0053  1 ! 1-007 - See if a control C was found and reset command buffer. STS 16-Jul-1982
; 54          0054  1 ! 1-008 - Improve the appearance of the listing.  JBS 14-Jun-1983
; 55          0055  1 !--
; 56          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$$MAC_CALL : NOVALUE,
;    67          0504  1     EDT$$LOAD_MACCAL : NOVALUE;
;    68          0505  1
;    69          0506  1 !
;    70          0507  1 ! INCLUDE FILES:
;    71          0508  1 !
;    72          0509  1
;    73          0510  1 REQUIRE 'EDTSRC:EDTREQ';
;    74          0645  1
;    75          0646  1 !
;    76          0647  1 ! MACROS:
;    77          0648  1 !
;    78          0649  1 !       NONE
;    79          0650  1 !
;    80          0651  1 ! EQUATED SYMBOLS:
;    81          0652  1 !
;    82          0653  1 !       NONE
;    83          0654  1 !
;    84          0655  1 ! OWN STORAGE:
;    85          0656  1 !
;    86          0657  1 !       NONE
;    87          0658  1 !
;    88          0659  1 ! EXTERNAL REFERENCES:
;    89          0660  1 !
;    90          0661  1 !       In the routine
```

```
;    92          0662  1 %SBTTL 'EDT$$MAC_CALL  - macro call'
;    93          0663  1
;    94          0664  1 GLOBAL ROUTINE EDT$$MAC_CALL (                  ! Macro call
;    95          0665  1     MAC                                         ! Address of the macro
;    96          0666  1     ) : NOVALUE =
;    97          0667  1
;    98          0668  1 !++
;    99          0669  1 ! FUNCTIONAL DESCRIPTION:
;   100          0670  1 !
;   101          0671  1 !       Call a macro.  A macro is a buffer which contains line-mode
;   102          0672  1 !       commands.
;   103          0673  1 !
;   104          0674  1 ! FORMAL PARAMETERS:
;   105          0675  1 !
;   106          0676  1 !  MAC                  Address of the macro
;   107          0677  1 !
;   108          0678  1 ! IMPLICIT INPUTS:
;   109          0679  1 !
;   110          0680  1 !       EDT$$A_CUR_BUF
;   111          0681  1 !       EDT$$A_MAC_BUF
;   112          0682  1 !       EDT$$A_CMD_BUF
;   113          0683  1 !       EDT$$G_CMD_LEN
;   114          0684  1 !       EDT$$T_CMD_BUF
;   115          0685  1 !       EDT$$A_CMD_END
;   116          0686  1 !
;   117          0687  1 ! IMPLICIT OUTPUTS:
;   118          0688  1 !
;   119          0689  1 !       EDT$$A_ UR_BUF
;   120          0690  1 !       EDT$$A_MAC_BUF
;   121          0691  1 !       EDT$$A_CMD_BUF
;   122          0692  1 !       EDT$$A_CMD_END
;   123          0693  1 !       EDT$$G_CC_DONE
;   124          0694  1 !
;   125          0695  1 ! ROUTINE VALUE:
;   126          0696  1 !
;   127          0697  1 !       NONE
;   128          0698  1 !
;   129          0699  1 ! SIDE EFFECTS:
;   130          0700  1 !
;   131          0701  1 !       Saves and restores the command line.
;   132          0702  1 !
;   133          0703  1 !--
;   134          0704  1
;   135          0705  2     BEGIN
;   136          0706  2
;   137          0707  2     EXTERNAL ROUTINE
;   138          0708  2         EDT$$EXE_CMD_NOOVERLAY,             ! Same as EDT$$EXE_CMD but no overlay analysis
;   139          0709  2         EDT$$RD_CURLN : NOVALUE,
;   140          0710  2         EDT$$TOP_BUF : NOVALUE,
;   141          0711  2         EDT$$ALO_HEAP,                      ! Allocate heap storage
;   142          0712  2         EDT$$DEA_HEAP : NOVALUE,            ! Deallocate heap storage
;   143          0713  2         EDT$$FMT_MSG : NOVALUE;             ! Format a message
;   144          0714  2
;   145          0715  2     EXTERNAL
;   146          0716  2         EDT$$G_CC_DONE,
;   147          0717  2         EDT$$A_CUR_BUF,                     ! Current tbcb
;   148          0718  2         EDT$$G_CMD_LEN,
```

```
;  149          0719  2          EDT$$T_CMD_BUF,
;  150          0720  2          EDT$$A_MAC_BUF,                 ! Current macro buffer pointer
;  151          0721  2          EDT$$A_CMD_BUF,                 ! Pointer to the current command
;  152          0722  2          EDT$$A_CMD_END;                 ! End of the current command
;  153          0723  2
;  154          0724  2      LOCAL
;  155          0725  2          STATUS,
;  156          0726  2          SAVE_TBCB,
;  157          0727  2          SAVE_MACRO,
;  158          0728  2          SAVE_CMD_BUF,
;  159          0729  2          SAVE_CMD_END,
;  160          0730  2          CMD_TEXT : REF VECTOR [, BYTE],
;  161          0731  2          CMD_LENGTH;
;  162          0732  2
;  163          0733  2      MESSAGES ((INSMEM));
;  164          0734  2  !+
;  165          0735  2  ! Save the current command line.
;  166          0736  2  !-
;  167          0737  2      SAVE_CMD_BUF = .EDT$$A_CMD_BUF;
;  168          0738  2      SAVE_CMD_END = .EDT$$A_CMD_END;
;  169          0739  2      CMD_LENGTH = CH$DIFF (CH$PLUS (.SAVE_CMD_END, 1), .SAVE_CMD_BUF);
;  170          0740  2
;  171          0741  3      IF (.CMD_LENGTH GTR 0)
;  172          0742  2      THEN
;  173          0743  3          BEGIN
;  174          0744  3
;  175          0745  3          IF EDT$$ALO_HEAP (CMD_LENGTH, CMD_TEXT)
;  176          0746  3          THEN
;  177          0747  3              CH$MOVE (.CMD_LENGTH, .SAVE_CMD_BUF, .CMD_TEXT)
;  178          0748  3          ELSE
;  179          0749  4              BEGIN
;  180          0750  4              EDT$$FMT_MSG (EDT$_INSMEM);
;  181          0751  4              RETURN;
;  182          0752  3              END;
;  183          0753  3
;  184          0754  2          END;
;  185          0755  2
;  186          0756  2  !+
;  187          0757  2  ! Point the command processor to the macro without destroying
;  188          0758  2  ! the current buffer or the current macro.
;  189          0759  2  !-
;  190          0760  2      SAVE_TBCB = .EDT$$A_CUR_BUF;
;  191          0761  2      EDT$$A_CUR_BUF = .MAC;
;  192          0762  2      EDT$$TOP_BOF ();
;  193          0763  2      EDT$$A_CUR_BUF = .SAVE_TBCB;
;  194          0764  2      EDT$$RD_CURLN ();
;  195          0765  2      SAVE_MACRO = .EDT$$A_MAC_BUF;
;  196          0766  2      EDT$$A_MAC_BUF = .MAC;
;  197          0767  2  !+
;  198          0768  2  ! Execute the commands in the specified buffer.
;  199          0769  2  !-
;  200          0770  2      STATUS = EDT$$EXE_CMD_NOOVERLAY (INP_MACRO);
;  201          0771  2  !+
;  202          0772  2  ! Restore the former macro.
;  203          0773  2  !-
;  204          0774  2      EDT$$A_MAC_BUF = .SAVE_MACRO;
;  205          0775  2  !+
```

```
;  206          0776  2 ! Restore the former command line contents, if any.
;  207          0777  2 !-
;  208          0778  2     EDT$$A_CMD_BUF = .SAVE_CMD_BUF;
;  209          0779  2     EDT$$A_CMD_END = .SAVE_CMD_END;
;  210          0780  2
;  211          0781  3     IF (.CMD_LENGTH GTR 0)
;  212          0782  2     THEN
;  213          0783  3         BEGIN
;  214          0784  3         CH$MOVE (.CMD_LENGTH, .CMD_TEXT, .SAVE_CMD_BUF);
;  215          0785  3         EDT$$DEA_HEAP (CMD_LENGTH, CMD_TEXT);
;  216          0786  2         END;
;  217          0787  2
;  218          0788  3     IF (.STATUS EQL 2)                          ! if we saw a control C
;  219          0789  2     THEN
;  220          0790  3         BEGIN
;  221          0791  3
;  222          0792  3         IF (.EDT$$A_CMD_END NEQ .EDT$$A_CMD_BUF) THEN EDT$$G_CC_DONE = 1;
;  223          0793  3
;  224          0794  3         EDT$$A_CMD_BUF = EDT$$T_CMD_BUF;
;  225          0795  3         EDT$$A_CMD_END = .EDT$$A_CMD_BUF;
;  226          0796  3         CH$WCHAR (%C'!', .EDT$$A_CMD_END);
;  227          0797  2         END;
;  228          0798  2
;  229          0799  1     END;                                        ! of routine EDT$MAC_CALL


                                                        .TITLE  EDT$MACCAL EDT$MACCAL - macro call
                                                        .IDENT  \V04-000\

                                                        .EXTRN  EDT$$EXE_CMD_NOOVERLAY
                                                        .EXTRN  EDT$$RD_CURLN, EDT$$TOP_BUF
                                                        .EXTRN  EDT$$ALO_HEAP, EDT$$DEA_HEAP
                                                        .EXTRN  EDT$$FMT_MSG, EDT$$G_CC_DONE
                                                        .EXTRN  EDT$$A_CUR_BUF, EDT$$G_CMD_LEN
                                                        .EXTRN  EDT$$T_CMD_BUF, EDT$$A_MAC_BUF
                                                        .EXTRN  EDT$$A_CMD_BUF, EDT$$A_CMD_END
                                                        .EXTRN  EDT$_INSMEM

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

                                    0FFC 00000          .ENTRY  EDT$$MAC_CALL, Save R2,R3,R4,R5,R6,R7,R8,-      ; 0664
                                                                R9,R10,R11
                          5B 00000000G  00   9E 00002   MOVAB   EDT$$A_CUR_BUF, R11                             ;
                          5A 00000000G  00   9E 00009   MOVAB   EDT$$A_CMD_BUF, R10                             ;
                          59 00000000G  00   9E 00010   MOVAB   EDT$$A_CMD_END, R9                              ;
                          5E            08   C2 00017   SUBL2   #8, SP                                        ;
                          57            6A   D0 0001A   MOVL    EDT$$A_CMD_BUF, SAVE_CMD_BUF                  ; 0737
                          56            69   D0 0001D   MOVL    EDT$$A_CMD_END, SAVE_CMD_END                  ; 0738
                    50    56            57   C3 00020   SUBL3   SAVE_CMD_BUF, SAVE_CMD_END, R0               ; 0739
                    04    AE       01   A0   9E 00024   MOVAB   1(R0), CMD_LENGTH                             ;
                                        25   15 00029   BLEQ    2$                                            ; 0741
                                        5E   DD 0002B   PUSHL   SP                                            ; 0745
                                   08   AE   9F 0002D   PUSHAB  CMD_LENGTH                                    ;
                00000000G  00           02   FB 00030   CALLS   #2, EDT$$ALO_HEAP                            ;
                           08           50   E9 00037   BLBC    R0, 1$                                        ;
                BE         67       04   AE   28 0003A   MOVC3   CMD_LENGTH, (SAVE_CMD_BUF), @CMD_TEXT       ; 0747
                                        0E   11 00040   BRB     2$                                            ;
```

```
                                00000000G  8F   DD 00042 1$:     PUSHL   #EDT$_INSMEM                                    ; 0750
                    00000000G   00         01   FB 00048        CALLS   #1, EDT$$FMT_MSG
                                           04 0004F            RET                                                     ; 0749
                                52         6B   D0 00050 2$:     MOVL    EDT$$A_CUR_BUF, SAVE_TBCB                       ; 0760
                                6B      04 AC   D0 00053        MOVL    MAC, EDT$$A_CUR_BUF                             ; 0761
                    00000000G   00         00   FB 00057        CALLS   #0, EDT$$TOP_BUF                                ; 0762
                                6B         52   D0 0005E        MOVL    SAVE_TBCB, EDT$$A_CUR_BUF                       ; 0763
                    00000000G   00         00   FB 00061        CALLS   #0, EDT$$RD_CURLN                               ; 0764
                                52 00000000G   00   D0 00068    MOVL    EDT$$A_MAC_BUF, SAVE_MACRO                      ; 0765
                    00000000G   00      04 AC   D0 0006F        MOVL    MAC, EDT$$A_MAC_BUF                             ; 0766
                                           01   DD 00077        PUSHL   #1                                              ; 0770
                    00000000G   00         01   FB 00079        CALLS   #1, EDT$$EXE_CMD_NOOVERLAY
                                58         50   D0 00080        MOVL    R0, STATUS
                    00000000G   00         52   D0 00083        MOVL    SAVE_MACRO, EDT$$A_MAC_BUF                      ; 0774
                                6A         57   D0 0008A        MOVL    SAVE_CMD_BUF, EDT$$A_CMD_BUF                    ; 0778
                                69         56   D0 0008D        MOVL    SAVE_CMD_END, EDT$$A_CMD_END                    ; 0779
                                        04 AE   D5 00090        TSTL    CMD_LENGTH                                      ; 0781
                                           12   15 00093        BLEQ    3$
                    67      00  BE      04 AE   28 00095        MOVC3   CMD_LENGTH, @CMD_TEXT, (SAVE_CMD_BUF)           ; 0784
                                           5E   DD 0009B        PUSHL   SP                                              ; 0785
                                        08 AE   9F 0009D        PUSHAB  CMD_LENGTH
                    00000000G   00         02   FB 000A0        CALLS   #2, EDT$$DEA_HEAP
                                02         58   D1 000A7 3$:     CMPL    STATUS, #2                                      ; 0788
                                           1C   12 000AA        BNEQ    5$
                                6A         69   D1 000AC        CMPL    EDT$$A_CMD_END, EDT$$A_CMD_BUF                  ; 0792
                                           07   13 000AF        BEQL    4$
                    00000000G   00         01   D0 000B1        MOVL    #1, EDT$$G_CC_DONE
                                6A 00000000G   00   9E 000B8 4$:     MOVAB   EDT$$T_CMD_BUF, EDT$$A_CMD_BUF          ; 0794
                                69         6A   D0 000BF        MOVL    EDT$$A_CMD_BUF, EDT$$A_CMD_END                  ; 0795
                                50         69   D0 000C2        MOVL    EDT$$A_CMD_END, R0                              ; 0796
                                60         21   90 000C5        MOVB    #33, (R0)
                                           04 000C8 5$:     RET                                                     ; 0799

; Routine Size:  201 bytes,    Routine Base:  _EDT$CODE + 0000

;  230          0800  1
```

```
;  232          0801  1 %SBTTL 'EDT$$LOAD_MACCAL - load this module into memory'
;  233          0802  1
;  234          0803  1 GLOBAL ROUTINE EDT$$LOAD_MACCAL                    ! Load this module into memory
;  235          0804  1     : NOVALUE =
;  236          0805  1
;  237          0806  1 !++
;  238          0807  1 ! FUNCTIONAL DESCRIPTION:
;  239          0808  1 !
;  240          0809  1 !       This routine has no function.  It exists as an entry point so that
;  241          0810  1 !       EDT$$EXE_CMD_NOOVERLAY can call this module back into memory before
;  242          0811  1 !       returning to it.
;  243          0812  1 !
;  244          0813  1 ! FORMAL PARAMETERS:
;  245          0814  1 !
;  246          0815  1 !       NONE
;  247          0816  1 !
;  248          0817  1 ! IMPLICIT INPUTS:
;  249          0818  1 !
;  250          0819  1 !       NONE
;  251          0820  1 !
;  252          0821  1 ! IMPLICIT OUTPUTS:
;  253          0822  1 !
;  254          0823  1 !       NONE
;  255          0824  1 !
;  256          0825  1 ! ROUTINE VALUE:
;  257          0826  1 !
;  258          0827  1 !       NONE
;  259          0828  1 !
;  260          0829  1 ! SIDE EFFECTS:
;  261          0830  1 !
;  262          0831  1 !       NONE
;  263          0832  1 !
;  264          0833  1 !--
;  265          0834  1
;  266          0835  2     BEGIN
;  267          0836  2     0
;  268          0837  1     END;                                           ! of routine EDT$$LOAD_MACCAL


                                                0000 00000          .ENTRY  EDT$$LOAD_MACCAL, Save nothing          ; 0803
                                                  04 00002          RET                                             ; 0837

; Routine Size:  3 bytes,    Routine Base:  _EDT$CODE + 00C9

;  269          0838  1
;  270          0839  1 !<BLF/PAGE>
```

```
;  272          0840  1 END                                          ! of module EDT$MACCAL
;  273          0841  1
;  274          0842  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _EDT$CODE                        204  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
```

```
;                                     COMMAND QUALIFIERS

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

; Size:          204 code + 0 data bytes
; Run Time:          00:14.5
; Elapsed Time:      00:18.1
; Lines/CPU Min:      3496
; Lexemes/CPU-Min:    9707
; Memory Used:  91 pages
; Compilation Complete
```

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