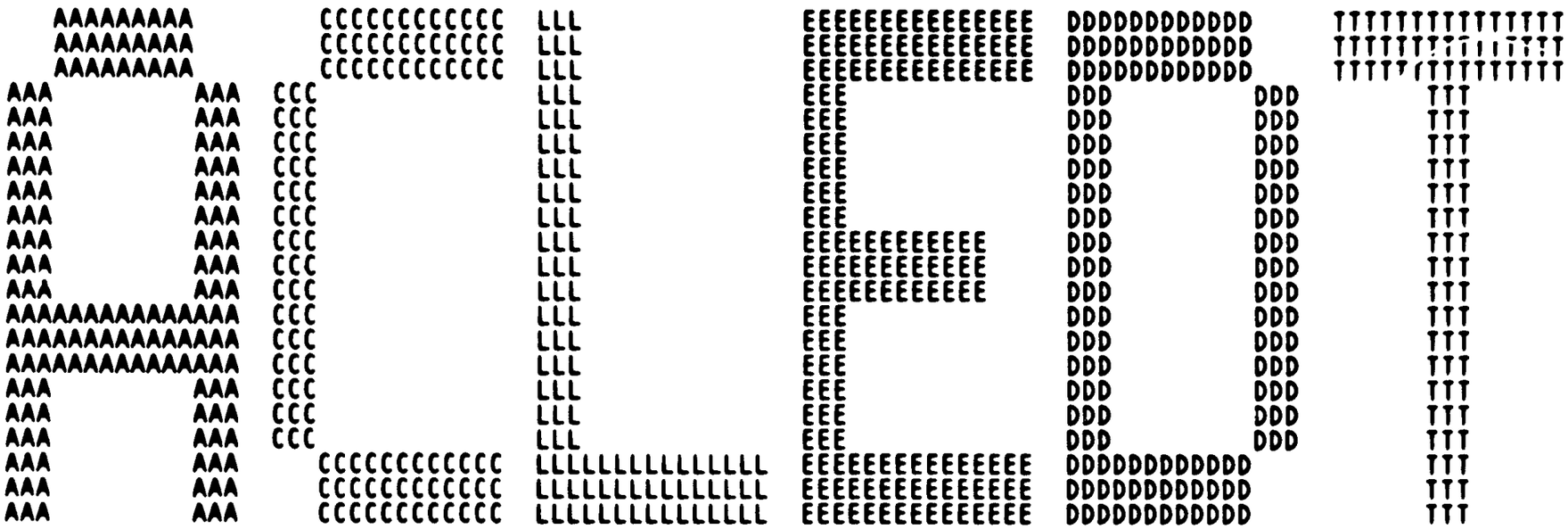
AAAAAAAAA
AAAAAAAAA
AAAAAAAAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
CCCCCCCCCCC
CCCCCCCCCCC
CCCCCCCCCCC
CCC
CCC
CCC
CCC
CCC
CCC
CCC
CCC
CCC
CCC
CCC
CCC
CCC
CCCCCCCCCCC
CCCCCCCCCCC
CCCCCCCCCCC
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLL
LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL
EEEEEEEEEEEEEEE
EEEEEEEEEEEEEEE
EEEEEEEEEEEEEEE
EEE
EEE
EEE
EEE
EEE
EEE
EEEEEEEEEEEEE
EEEEEEEEEEEEE
EEEEEEEEEEEEE
EEE
EEE
EEE
EEE
EEE
EEE
EEEEEEEEEEEEEEE
EEEEEEEEEEEEEEE
EEEEEEEEEEEEEEE
DDDDDDDDDDDD
DDDDDDDDDDDD
DDDDDDDDDDDD
DDD DDD
DDD DDD
DDD DDD
DDD DDD
DDD DDD
DDD DDD
DDD DDD
DDD DDD
DDD DDD
DDD DDD
DDDDDDDDDDDD
DDDDDDDDDDDD
DDDDDDDDDDDD
TTTTTTTTTTTTTTT
TTTTTTTTTTTTTTT
TTTTTTTTTTTTTTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT
TTT


AAAAAA EEEEEEEEEE DDDDDDDD DDDDDDDD EEEEEEEEEE CCCCCCCC OOOOOO DDDDDDDD EEEEEEEEEE
AAAAAA EEEEEEEEEE DDDDDDDD DDDDDDDD EEEEEEEEEE CCCCCCCC OOOOOO DDDDDDDD EEEEEEEEEE
AA AA EE DD DD DD DD EE CC OO OO DD DD EE
AA AA EE DD DD DD DD EE CC OO OO DD DD EE
AA AA EE DD DD DD DD EE CC OO OO DD DD EE
AA AA EE DD DD DD DD EE CC OO OO DD DD EE
AA AA EEEEEEEE DD DD DD DD EEEEEEEE CC OO OO DD DD EEEEEEEE
AA AA EEEEEEEE DD DD DD DD EEEEEEEE CC OO OO DD DD EEEEEEEE
AAAAAAAAAA EE DD DD DD DD EE CC OO OO DD DD EE
AAAAAAAAAA EE DD DD DD DD EE CC OO OO DD DD EE
AA AA EE DD DD DD DD EE CC OO OO DD DD EE
AA AA EE DD DD DD DD EE CC OO OO DD DD EE
AA AA EEEEEEEEEE DDDDDDDD DDDDDDDD EEEEEEEEEE CCCCCCCC OOOOOO DDDDDDDD EEEEEEEEEE
AA AA EEEEEEEEEE DDDDDDDD DDDDDDDD EEEEEEEEEE CCCCCCCC OOOOOO DDDDDDDD 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
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSS

```
;     1          0001  0 MODULE AED$DECODE (
;     2          0002  0                     LANGUAGE (BLISS32),
;     3          0003  0                     IDENT = 'V04-000'
;     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:     Miscellaneous utilities
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       This module contains the routines necessary to read the action
;    37          0037  1 !       definition file and decode the users input based upon the action
;    38          0038  1 !       definitions.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !
;    42          0042  1 !       VAX/VMS operating system, user mode utilities.
;    43          0043  1 !
;    44          0044  1 !--
;    45          0045  1 !
;    46          0046  1 !
;    47          0047  1 ! AUTHOR:       L. Mark Pilant          CREATION DATE:  15-Sep-1982  15:30
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;    51          0051  1 !       V03-005 LMP0213         L. Mark Pilant,         24-Mar-1984  12:23
;    52          0052  1 !               Add support for locking and unlocking the object's ACL.
;    53          0053  1 !
;    54          0054  1 !       V03-004 LMP0193         L. Mark Pilant,         14-Feb-1984  10:04
;    55          0055  1 !               Add support for additional edition actions: delete BOL,
;    56          0056  1 !               session reset, and quit session.
;    57          0057  1 !
```

```
;   58          0058  1 !       V03-003 LMP0172         L. Mark Pilant,         28-Nov-1983  12:11
;   59          0059  1 !               Numerous bug fixes, support for VT2xx terminals, and a
;   60          0060  1 !               session keystroke logger.
;   61          0061  1 !
;   62          0062  1 !       V03-002 LMP0142         L. Mark Pilant,         24-Aug-1983  3:17
;   63          0063  1 !               Change references to ACLEDIT$INI to be ACLEDIT$INIT.
;   64          0064  1 !
;   65          0065  1 !       V03-001 LMP0103         L. Mark Pilant,         21-Apr-1983  12:44
;   66          0066  1 !               Add support for HIDDEN and PROTECTED ACEs.
;   67          0067  1 !
;   68          0068  1 !**
;   69          0069  1
;   70          0070  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;   71          0071  1 LIBRARY 'SYS$LIBRARY:TPAMAC.L32';
;   72          0072  1 REQUIRE 'SRC$:ACLEDTDEF';
```

```
;    74          0525  1 FORWARD ROUTINE
;    75          0526  1         AED_GETKEYINI,                  ! Check for & read definition file
;    76          0527  1         AED_DECODEKEY,                  ! Decode input given definitions
;    77          0528  1         AED_FLUSHKEY,                   ! Flush session buffer & close file
;    78          0529  1
;    79          0530  1 ! TPARSE action routine.
;    80          0531  1
;    81          0532  1         SET_RUBOUT,                     ! Set rubout as the string definition
;    82          0533  1         SET_DEFINITION;                 ! Define a key
;    83          0534  1
;    84          0535  1 EXTERNAL ROUTINE
;    85          0536  1         AED_FILERROR    : NOVALUE,      ! RMS file error reporting
;    86          0537  1         AED_PUTOUTPUT,                  ! General purpose output routine
;    87          0538  1         AED_SET_CURSOR;                 ! Set cursor position & remember
;    88          0539  1
;    89          0540  1 EXTERNAL
;    90          0541  1         KEY_TABLE       : $BBLOCK [8];  ! Key definition table listhead
;    91          0542  1
;    92          0543  1 ! Storage for TPARSE usage.
;    93          0544  1
;    94          0545  1 OWN
;    95          0546  1         KEY_BLOCK       : $BBLOCK [KEY_C_LENGTH],       ! Key definition block
;    96          0547  1         KEY_STRING      : $BBLOCK [DSC$C_S_BLN];        ! Key string descriptor
;    97          0548  1
;    98          0549  1 BIND
;    99          0550  1         KEY_ACTION      = KEY_BLOCK[KEY_B_ACTION] : BYTE,       ! Action code
;   100          0551  1         KEY_FLAGS       = KEY_BLOCK[KEY_B_FLAGS] : BYTE;        ! Needed flags
;   101          0552  1
;   102          0553  1 ! TPARSE state tables to parse the action definition file.
;   103          0554  1
;   104          0555  1 $INIT_STATE     (KEYDEF_STATE, KEYDEF_KEY);
;   105          0556  1
;   106        P 0557  1 $STATE  (SWALLOW_1,
;   107        P 0558  1          (TPA$_BLANK,SWALLOW_1),
;   108        P 0559  1          ('DEFINE')
;   109          0560  1          );
;   110          0561  1
;   111        P 0562  1 $STATE  (SWALLOW_2,
;   112        P 0563  1          (TPA$_BLANK,SWALLOW_2),
;   113        P 0564  1          ('GOLD',,KEY_C_GOLD,KEY_ACTION),
;   114        P 0565  1          ('HELP',,KEY_C_HELP,KEY_ACTION),
;   115        P 0566  1          ('HELP_FORMAT',,KEY_C_HELPFMT,KEY_ACTION),
;   116        P 0567  1          ('LOCATE_STRING',,KEY_C_FIND_STR,KEY_ACTION),
;   117        P 0568  1          ('LOCATE_NEXT',,KEY_C_FIND_NXT,KEY_ACTION),
;   118        P 0569  1          ('DELETE_ACE',,KEY_C_DEL_ACE,KEY_ACTION),
;   119        P 0570  1          ('UNDELETE_ACE',,KEY_C_UNDEL_ACE,KEY_ACTION),
;   120        P 0571  1          ('SELECT_FIELD',,KEY_C_SEL_FIELD,KEY_ACTION),
;   121        P 0572  1          ('ADVANCE_FIELD',,KEY_C_ADV_FIELD,KEY_ACTION),
;   122        P 0573  1          ('DELETE_WORD',,KEY_C_DEL_WRD,KEY_ACTION),
;   123        P 0574  1          ('UNDELETE_WORD',,KEY_C_UNDEL_WRD,KEY_ACTION),
;   124        P 0575  1          ('ADVANCE_POSITION',,KEY_C_ADVANCE,KEY_ACTION),
;   125        P 0576  1          ('BACKUP_POSITION',,KEY_C_BACKUP,KEY_ACTION),
;   126        P 0577  1          ('DELETE_CHARACTER',,KEY_C_DEL_CHR,KEY_ACTION),
;   127        P 0578  1          ('UNDELETE_CHARACTER',,KEY_C_UNDEL_CHR,KEY_ACTION),
;   128        P 0579  1          ('MOVE_WORD',,KEY_C_MOVE_WRD,KEY_ACTION),
;   129        P 0580  1          ('MOVE_ACE',,KEY_C_MOVE_ACE,KEY_ACTION),
;   130        P 0581  1          ('MOVE_EOL',,KEY_C_MOVE_EOL,KEY_ACTION),
```

```
;  131          P 0582  1          ('DELETE_EOL',,KEY_C_DEL_EOL,KEY_ACTION),
;  132          P 0583  1          ('INSERT_ACE',,KEY_C_INSERT,KEY_ACTION),
;  133          P 0584  1          ('SELECT_ITEM',,KEY_C_SEL_ITEM,KEY_ACTION),
;  134          P 0585  1          ('ENTER_ACE',,KEY_C_ENTER,KEY_ACTION),
;  135          P 0586  1          ('PREVIOUS_SCREEN',,KEY_C_PREV_SCREEN,KEY_ACTION),
;  136          P 0587  1          ('NEXT_SCREEN',,KEY_C_NEXT_SCREEN,KEY_ACTION),
;  137          P 0588  1          ('UP_ARROW',,KEY_C_UP,KEY_ACTION),
;  138          P 0589  1          ('DOWN_ARROW',,KEY_C_DOWN,KEY_ACTION),
;  139          P 0590  1          ('RIGHT_ARROW',,KEY_C_RIGHT,KEY_ACTION),
;  140          P 0591  1          ('LEFT_ARROW',,KEY_C_LEFT,KEY_ACTION),
;  141          P 0592  1          ('INSERT_OVERSTRIKE',,KEY_C_OVERSTRIKE,KEY_ACTION),
;  142          P 0593  1          ('MOVE_BOL',,KEY_C_MOVE_BOL,KEY_ACTION),
;  143          P 0594  1          ('RUBOUT_WORD',,KEY_C_RUB_WRD,KEY_ACTION),
;  144          P 0595  1          ('SCREEN_REFRESH',,KEY_C_REFRESH,KEY_ACTION),
;  145          P 0596  1          ('SESSION_RESET',,KEY_C_RESET,KEY_ACTION),
;  146          P 0597  1          ('RUBOUT_BOL',,KEY_C_RUB_BOL,KEY_ACTION),
;  147          P 0598  1          ('UNDELETE_LINE',,KEY_C_UNDEL_LIN,KEY_ACTION),
;  148          P 0599  1          ('EXIT',,KEY_C_EXIT,KEY_ACTION),
;  149          P 0600  1          ('QUIT_SESSION',,KEY_C_QUIT,KEY_ACTION),
;  150          P 0601  1          ('RUBOUT_CHARACTER',,KEY_C_RUB_CHR,KEY_ACTION)
;  151            0602  1          );
;  152            0603  1
;  153          P 0604  1 $STATE  (SWALLOW_3,
;  154          P 0605  1          (TPA$_BLANK,SWALLOW_3),
;  155          P 0606  1          ('AS')
;  156            0607  1          );
;  157            0608  1
;  158          P 0609  1 $STATE  (KEY_DEFINE,
;  159          P 0610  1          (TPA$_BLANK,KEY_DEFINE),
;  160          P 0611  1          ('GOLD',,KEY_M_GOLDREQ,KEY_FLAGS),
;  161          P 0612  1          ('CONTROL',GET_TEXT,,KEY_M_CTRLCHAR,KEY_FLAGS),
;  162          P 0613  1          ('ESCAPE',GET_TEXT,,KEY_M_ESCSEQ,KEY_FLAGS),
;  163          P 0614  1          ('CSI',GET_TEXT,,KEY_M_CSI,KEY_FLAGS),
;  164          P 0615  1          ('SS3',GET_TEXT,,KEY_M_SS3,KEY_FLAGS),
;  165          P 0616  1          ('RUBOUT',,SET_RUBOUT),
;  166          P 0617  1          (TPA$_EOS,TPA$_FAIL),
;  167            0618  1          );
;  168          P 0619  1 $STATE  (CHECK_END,
;  169          P 0620  1          (TPA$_BLANK,CHECK_END),
;  170          P 0621  1          (',',KEY_DEFINE),
;  171          P 0622  1          ('OR',KEY_DEFINE,SET_DEFINITION),
;  172          P 0623  1          (TPA$_EOS,TPA$_EXIT,SET_DEFINITION)
;  173            0624  1          );
;  174            0625  1
;  175          P 0626  1 $STATE  (GET_TEXT,
;  176          P 0627  1          (TPA$_BLANK,GET_TEXT),
;  177          P 0628  1          (',')
;  178            0629  1          );
;  179          P 0630  1 $STATE  (SWALLOW_4,
;  180          P 0631  1          (TPA$_BLANK,SWALLOW_4),
;  181          P 0632  1          ((GET_STRING),CHECK_END,,KEY_STRING)
;  182            0633  1          );
;  183            0634  1
;  184          P 0635  1 $STATE  (GET_STRING,
;  185          P 0636  1          ((CHECK_DELIM),GET_STRING),
;  186          P 0637  1          (TPA$_LAMBDA,TPA$_EXIT)
;  187            0638  1          );
```

```
;  188          P 0639  1 $STATE  (CHECK_DELIM,
;  189          P 0640  1          (',',TPA$_FAIL),
;  190          P 0641  1          (' ',TPA$_FAIL),
;  191          P 0642  1          (TPA$_EOS,TPA$_FAIL),
;  192          P 0643  1          (TPA$_ANY,TPA$_EXIT)
;  193            0644  1          );
```

```
;  195          0645  1 GLOBAL ROUTINE AED_GETKEYINI =
;  196          0646  1
;  197          0647  1 !++
;  198          0648  1 !
;  199          0649  1 ! FUNCTIONAL DESCRIPTION:
;  200          0650  1 !
;  201          0651  1 !       This routine attempts to open the action definition file pointed
;  202          0652  1 !       to by the logical name ACLEDIT$INIT.  If the logical name does not
;  203          0653  1 !       exist a success return is given.  If the logical name exists, but
;  204          0654  1 !       the file it points to does not, a warning message is given, and a
;  205          0655  1 !       success return is given.  If any errors occur while reading the
;  206          0656  1 !       definition file, the appropriate error message is given.
;  207          0657  1 !
;  208          0658  1 ! CALLING SEQUENCE:
;  209          0659  1 !       AED_GETKEYINI ()
;  210          0660  1 !
;  211          0661  1 ! INPUT PARAMETERS:
;  212          0662  1 !       none
;  213          0663  1 !
;  214          0664  1 ! IMPLICIT INPUTS:
;  215          0665  1 !       none
;  216          0666  1 !
;  217          0667  1 ! OUTPUT PARAMETERS:
;  218          0668  1 !       none
;  219          0669  1 !
;  220          0670  1 ! IMPLICIT OUTPUTS:
;  221          0671  1 !       none
;  222          0672  1 !
;  223          0673  1 ! ROUTINE VALUE:
;  224          0674  1 !       1 if successful, logical name does not exist, or file does not exist
;  225          0675  1 !       error code otherwise
;  226          0676  1 !
;  227          0677  1 ! SIDE EFFECTS:
;  228          0678  1 !       none
;  229          0679  1 !
;  230          0680  1 !--
;  231          0681  1
;  232          0682  2 BEGIN
;  233          0683  2
;  234          0684  2 LOCAL
;  235          0685  2         KEYINI_FAB      : $FAB_DECL,                    ! Key definition file FAB
;  236          0686  2         KEYINI_RAB      : $RAB_DECL,                    ! Key definition file RAB
;  237          0687  2         KEYINI_NAM      : $NAM_DECL,                    ! Key definition file NAM block
;  238          0688  2         KEYINI_EXP_NAM  : $BBLOCK [NAM$C_MAXRSS],       ! Expanded name storage
;  239          0689  2         KEYINI_RES_NAM  : $BBLOCK [NAM$C_MAXRSS],       ! Resultant name storage
;  240          0690  2         DEFINE_LINE     : VECTOR [512,BYTE],            ! Line from definition file
;  241          0691  2         TPARSE_BLOCK    : $BBLOCK [TPA$K_LENGTH0],      ! Parser context block
;  242          0692  2         LINE_INDEX,                                     ! Index into line read in
;  243          0693  2         LOCAL_STATUS;                                   ! Local error status
;  244          0694  2
;  245          0695  2 ! Initialize the necessary RMS data structures.
;  246          0696  2
;  247        P 0697  2 $FAB_INIT (FAB = KEYINI_FAB,
;  248        P 0698  2             FAC = GET,
;  249        P 0699  2             FNA = UPLIT ('ACLEDIT$INIT:'),
;  250        P 0700  2             FNS = %CHARCOUNT ('ACLEDIT$INIT:'),
;  251        P 0701  2             FOP = SQO,
```

```
;  252        P 0702  2              NAM = KEYINI_NAM,
;  253        P 0703  2              ORG = SEQ,
;  254          0704  2              RFM = VAR);
;  255        P 0705  2 $NAM_INIT (NAM = KEYINI_NAM,
;  256        P 0706  2              ESA = KEYINI_EXP_NAM,
;  257        P 0707  2              ESS = NAM$C_MAXRSS,
;  258        P 0708  2              RSA = KEYINI_RES_NAM,
;  259          0709  2              RSS = NAM$C_MAXRSS);
;  260        P 0710  2 $RAB_INIT (RAB = KEYINI_RAB,
;  261        P 0711  2              FAB = KEYINI_FAB,
;  262          0712  2              RAC = SEQ);
;  263          0713  2
;  264          0714  2 ! Open the action definition file.  If the open results in the RMS$_DEV error,
;  265          0715  2 ! it is assumed that the logical name does not exist, and success is returned.
;  266          0716  2 ! If the open results in the RMS$_FNF error, a warning message is issued, and
;  267          0717  2 ! success is returned.  Any other error results in the approprate error message
;  268          0718  2 ! being signaled, and the editing session terminated.
;  269          0719  2
;  270          0720  3 IF NOT $OPEN (FAB = KEYINI_FAB)
;  271          0721  2 THEN
;  272          0722  3     BEGIN
;  273          0723  3     IF .KEYINI_FAB[FAB$L_STS] EQL RMS$_DEV THEN RETURN 1;
;  274          0724  3     AED_FILERROR (AED$_INIOPENIN, KEYINI_FAB, .KEYINI_FAB[FAB$L_STS],
;  275          0725  3                                               .KEYINI_FAB[FAB$L_STV]);
;  276          0726  3     IF .KEYINI_FAB[FAB$L_STS] EQL RMS$_FNF THEN RETURN 1;
;  277          0727  3     RETURN .AED_L_WORSTERR;
;  278          0728  2     END;
;  279          0729  3 IF NOT $CONNECT (RAB = KEYINI_RAB)
;  280          0730  2 THEN
;  281          0731  3     BEGIN
;  282          0732  3     AED_FILERROR (AED$_INIOPENIN, KEYINI_FAB, .KEYINI_RAB[RAB$L_STS],
;  283          0733  3                                               .KEYINI_RAB[RAB$L_STV]);
;  284          0734  3     RETURN .AED_L_WORSTERR;
;  285          0735  2     END;
;  286          0736  2
;  287          0737  2 ! Loop reading the action definition file, replacing any default definition
;  288          0738  2 ! with those from the definition file.
;  289          0739  2
;  290          0740  2 WHILE 1
;  291          0741  2 DO
;  292          0742  3     BEGIN
;  293          0743  3     KEYINI_RAB[RAB$L_UBF] = DEFINE_LINE;
;  294          0744  3     KEYINI_RAB[RAB$W_USZ] = 512;
;  295          0745  4     IF NOT $GET (RAB = KEYINI_RAB)
;  296          0746  3     THEN
;  297          0747  4         BEGIN
;  298          0748  4         IF .KEYINI_RAB[RAB$L_STS] EQL RMS$_EOF THEN EXITLOOP;
;  299          0749  4         AED_FILERROR (AED$_INIREADERR, KEYINI_FAB, .KEYINI_RAB[RAB$L_STS],
;  300          0750  4                                                    .KEYINI_RAB[RAB$L_STV]);
;  301          0751  4         RETURN .AED_L_WORSTERR;
;  302          0752  3         END;
;  303          0753  3     KEY_ACTION = 0;
;  304          0754  3     KEY_FLAGS = 0;
;  305          0755  3     KEY_STRING[DSC$W_LENGTH] = 0;
;  306          0756  3
;  307          0757  3     IF .DEFINE_LINE[0] NEQ '!'
;  308          0758  3     THEN
```

```
;  309          0759  4          BEGIN
;  310          0760  4          LINE_INDEX = 0;
;  311          0761  4          UNTIL .LINE_INDEX GEQ .KEYINI_RAB[RAB$W_RSZ]
;  312          0762  4          DO
;  313          0763  5              BEGIN
;  314          0764  5              IF .DEFINE_LINE[.LINE_INDEX] EQL '<'
;  315          0765  5              THEN
;  316          0766  6                  BEGIN
;  317          0767  6                  DO
;  318          0768  7                      BEGIN
;  319          0769  7                      LINE_INDEX = .LINE_INDEX + 1;
;  320          0770  7                      IF .DEFINE_LINE[.LINE_INDEX] EQL '>' THEN EXITLOOP;
;  321          0771  7                      IF .LINE_INDEX GEQ .KEYINI_RAB[RAB$W_RSZ]
;  322          0772  7                      THEN
;  323          0773  8                          BEGIN
;  324        P 0774  8                          SIGNAL (AED$_DEFSYNTAX, 2, .KEYINI_RAB[RAB$W_RSZ],
;  325          0775  8                                                     DEFINE_LINE);
;  326          0776  8                          RETURN AED$_DEFSYNTAX;
;  327          0777  7                          END;
;  328          0778  7                      END
;  329          0779  6                  UNTIL .LINE_INDEX GEQ .KEYINI_RAB[RAB$W_RSZ];
;  330          0780  5                  END;
;  331          0781  5              IF .DEFINE_LINE[.LINE_INDEX] GEQ 'a'
;  332          0782  5              AND .DEFINE_LINE[.LINE_INDEX] LEQ 'z'
;  333          0783  5              THEN DEFINE_LINE[.LINE_INDEX] = .DEFINE_LINE[.LINE_INDEX] - 32;
;  334          0784  5              LINE_INDEX = .LINE_INDEX + 1;
;  335          0785  4              END;
;  336          0786  4          TPARSE_BLOCK[TPA$L_COUNT] = TPA$K_COUNTO;
;  337          0787  4          TPARSE_BLOCK[TPA$V_ABBREV] = 1;
;  338          0788  4          TPARSE_BLOCK[TPA$V_BLANKS] = 1;
;  339          0789  4          TPARSE_BLOCK[TPA$L_STRINGCNT] = .KEYINI_RAB[RAB$W_RSZ];
;  340          0790  4          TPARSE_BLOCK[TPA$L_STRINGPTR] = DEFINE_LINE;
;  341          0791  4
;  342          0792  4          LOCAL_STATUS = LIB$TPARSE (TPARSE_BLOCK, KEYDEF_STATE, KEYDEF_KEY);
;  343          0793  4          IF NOT .LOCAL_STATUS
;  344          0794  4          THEN
;  345          0795  5              BEGIN
;  346        P 0796  5              SIGNAL (AED$_DEFSYNTAX, 2, .TPARSE_BLOCK[TPA$L_STRINGCNT],
;  347          0797  5                                         .TPARSE_BLOCK[TPA$L_STRINGPTR]);
;  348          0798  5              RETURN AED$_DEFSYNTAX;
;  349          0799  4              END;
;  350          0800  3          END;
;  351          0801  2      END;
;  352          0802  2
;  353          0803  2  RETURN 1;
;  354          0804  2
;  355          0805  1  END;                                   ! End of routine AED_GETKEYINI


                                                                .TITLE  AED$DECODE
                                                                .IDENT  \V04-000\

                                                                .PSECT  _LIB$KEY1$,NOWRT,  SHR,  PIC,1

                                                    00000 ;TPA$KEYSTO
                                                          U.4:      .BLKB   0
                                45  4E  49  46  45  44   00000 ;TPA$KEYST
```

```
                                                        U.6:      .ASCII  \DEFINE\                    ;
                                           FF  00006              .BYTE   -1                          ;
                                           FF  00007  ;TPA$KEYFILL                                    ;
                                                        U.8:      .BYTE   -1                          ;
                                               00008  ;TPA$KEYST0
                                                        U.11:     .BLKB   0
                               44  4C  4F  47  00008  ;TPA$KEYST
                                                        U.13:     .ASCII  \GOLD\                      ;
                                           FF  0000C              .BYTE   -1                          ;
                                               0000D  ;TPA$KEYST0
                                                        U.17:     .BLKB   0
                               50  4C  45  48  0000D  ;TPA$KEYST
                                                        U.19:     .ASCII  \HELP\                      ;
                                           FF  00011              .BYTE   -1                          ;
                                               00012  ;TPA$KEYST0
                                                        U.23:     .BLKB   0
           54  41  4D  52  4F  46  5F  50  4C  45  48  00012  ;TPA$KEYST
                                                        U.25:     .ASCII  \HELP_FORMAT\               ;
                                           FF  0001D              .BYTE   -1                          ;
                                               0001E  ;TPA$KEYST0
                                                        U.29:     .BLKB   0
   47  4E  49  52  54  53  5F  45  54  41  43  4F  4C  0001E  ;TPA$KEYST
                                                        U.31:     .ASCII  \LOCATE_STRING\             ;
                                           FF  0002B              .BYTE   -1                          ;
                                               0002C  ;TPA$KEYST0
                                                        U.35:     .BLKB   0
           54  58  45  4E  5F  45  54  41  43  4F  4C  0002C  ;TPA$KEYST
                                                        U.37:     .ASCII  \LOCATE_NEXT\               ;
                                           FF  00037              .BYTE   -1                          ;
                                               00038  ;TPA$KEYST0
                                                        U.41:     .BLKB   0
               45  43  41  5F  45  54  45  4C  45  44  00038  ;TPA$KEYST
                                                        U.43:     .ASCII  \DELETE_ACE\                ;
                                           FF  00042              .BYTE   -1                          ;
                                               00043  ;TPA$KEYST0
                                                        U.47:     .BLKB   0
       45  43  41  5F  45  54  45  4C  45  44  4E  55  00043  ;TPA$KEYST
                                                        U.49:     .ASCII  \UNDELETE_ACE\              ;
                                           FF  0004F              .BYTE   -1                          ;
                                               00050  ;TPA$KEYST0
                                                        U.53:     .BLKB   0
       44  4C  45  49  46  5F  54  43  45  4C  45  53  00050  ;TPA$KEYST
                                                        U.55:     .ASCII  \SELECT_FIELD\              ;
                                           FF  0005C              .BYTE   -1                          ;
                                               0005D  ;TPA$KEYST0
                                                        U.59:     .BLKB   0
   44  4C  45  49  46  5F  45  43  4E  41  56  44  41  0005D  ;TPA$KEYST
                                                        U.61:     .ASCII  \ADVANCE_FIELD\             ;
                                           FF  0006A              .BYTE   -1                          ;
                                               0006B  ;TPA$KEYST0
                                                        U.65:     .BLKB   0
           44  52  4F  57  5F  45  54  45  4C  45  44  0006B  ;TPA$KEYST
                                                        U.67:     .ASCII  \DELETE_WORD\               ;
                                           FF  00076              .BYTE   -1                          ;
                                               00077  ;TPA$KEYST0
                                                        U.71:     .BLKB   0
   44  52  4F  57  5F  45  54  45  4C  45  44  4E  55  00077  ;TPA$KEYST
```

```
                                                        U.73:    .ASCII  \UNDELETE_WORD\                    ;
                                               FF 00084          .BYTE   -1                                 ;
                                                  00085 ;TPA$KEYST0
                                                        U.77:    .BLKB   0
4F 49 54 49 53 4F 50 5F 45 43 4E 41 56 44 41 00085 ;TPA$KEYST
                                                        U.79:    .ASCII  \ADVANCE_POSITION\                 ;
                                               4E 00094                                                     ;
                                               FF 00095          .BYTE   -1                                 ;
                                                  00096 ;TPA$KEYST0
                                                        U.83:    .BLKB   0
4E 4F 49 54 49 53 4F 50 5F 50 55 4B 43 41 42 00096 ;TPA$KEYST
                                                        U.85:    .ASCII  \BACKUP_POSITION\                  ;
                                               FF 000A5          .BYTE   -1                                 ;
                                                  000A6 ;TPA$KEYST0
                                                        U.89:    .BLKB   0
45 54 43 41 52 41 48 43 5F 45 54 45 4C 45 44 000A6 ;TPA$KEYST
                                                        U.91:    .ASCII  \DELETE_CHARACTER\                 ;
                                               52 000B5                                                     ;
                                               FF 000B6          .BYTE   -1                                 ;
                                                  000B7 ;TPA$KEYST0
                                                        U.95:    .BLKB   0
43 41 52 41 48 43 5F 45 54 45 4C 45 44 4E 55 000B7 ;TPA$KEYST
                                                        U.97:    .ASCII  \UNDELETE_CHARACTER\               ;
                                         52 45 54 000C6                                                     ;
                                               FF 000C9          .BYTE   -1                                 ;
                                                  000CA ;TPA$KEYST0
                                                        U.101:   .BLKB   0
                     44 52 4F 57 5F 45 56 4F 4D 000CA ;TPA$KEYST
                                                        U.103:   .ASCII  \MOVE_WORD\                        ;
                                               FF 000D3          .BYTE   -1                                 ;
                                                  000D4 ;TPA$KEYST0
                                                        U.107:   .BLKB   0
                        45 43 41 5F 45 56 4F 4D 000D4 ;TPA$KEYST
                                                        U.109:   .ASCII  \MOVE_ACE\                         ;
                                               FF 000DC          .BYTE   -1                                 ;
                                                  000DD ;TPA$KEYST0
                                                        U.113:   .BLKB   0
                        4C 4F 45 5F 45 56 4F 4D 000DD ;TPA$KEYST
                                                        U.115:   .ASCII  \MOVE_EOL\                         ;
                                               FF 000E5          .BYTE   -1                                 ;
                                                  000E6 ;TPA$KEYST0
                                                        U.119:   .BLKB   0
                  4C 4F 45 5F 45 54 45 4C 45 44 000E6 ;TPA$KEYST
                                                        U.121:   .ASCII  \DELETE_EOL\                       ;
                                               FF 000F0          .BYTE   -1                                 ;
                                                  000F1 ;TPA$KEYST0
                                                        U.125:   .BLKB   0
                  45 43 41 5F 54 52 45 53 4E 49 000F1 ;TPA$KEYST
                                                        U.127:   .ASCII  \INSERT_ACE\                       ;
                                               FF 000FB          .BYTE   -1                                 ;
                                                  000FC ;TPA$KEYST0
                                                        U.131:   .BLKB   0
               4D 45 54 49 5F 54 43 45 4C 45 53 000FC ;TPA$KEYST
                                                        U.133:   .ASCII  \SELECT_ITEM\                      ;
                                               FF 00107          .BYTE   -1                                 ;
                                                  00108 ;TPA$KEYST0
                                                        U.137:   .BLKB   0
```

```
                                45 43 41 5F 52 45 54 4E 45  00108 ;TPA$KEYST
                                                            U.139:  .ASCII  \ENTER_ACE\
                                                         FF 00111           .BYTE   -1
                                                            00112 ;TPA$KEYSTO
                                                            U.143:  .BLKB   0
4E 45 45 52 43 53 5F 53 55 4F 49 56 45 52 50  00112 ;TPA$KEYST
                                                            U.145:  .ASCII  \PREVIOUS_SCREEN\
                                                         FF 00121           .BYTE   -1
                                                            00122 ;TPA$KEYSTO
                                                            U.149:  .BLKB   0
                             4E 45 45 52 43 53 5F 54 58 45 4E  00122 ;TPA$KEYST
                                                            U.151:  .ASCII  \NEXT_SCREEN\
                                                         FF 0012D           .BYTE   -1
                                                            0012E ;TPA$KEYSTO
                                                            U.155:  .BLKB   0
                                   57 4F 52 52 41 5F 50 55  0012E ;TPA$KEYST
                                                            U.157:  .ASCII  \UP_ARROW\
                                                         FF 00136           .BYTE   -1
                                                            00137 ;TPA$KEYSTO
                                                            U.161:  .BLKB   0
                                57 4F 52 52 41 5F 4E 57 4F 44  00137 ;TPA$KEYST
                                                            U.163:  .ASCII  \DOWN_ARROW\
                                                         FF 00141           .BYTE   -1
                                                            00142 ;TPA$KEYSTO
                                                            U.167:  .BLKB   0
                             57 4F 52 52 41 5F 54 48 47 49 52  00142 ;TPA$KEYST
                                                            U.169:  .ASCII  \RIGHT_ARROW\
                                                         FF 0014D           .BYTE   -1
                                                            0014E ;TPA$KEYSTO
                                                            U.173:  .BLKB   0
                                57 4F 52 52 41 5F 54 46 45 4C  0014E ;TPA$KEYST
                                                            U.175:  .ASCII  \LEFT_ARROW\
                                                         FF 00158           .BYTE   -1
                                                            00159 ;TPA$KEYSTO
                                                            U.179:  .BLKB   0
49 52 54 53 52 45 56 4F 5F 54 52 45 53 4E 49  00159 ;TPA$KEYST
                                                            U.181:  .ASCII  \INSERT_OVERSTRIKE\
                                                      45 4B 00168
                                                         FF 0016A           .BYTE   -1
                                                            0016B ;TPA$KEYSTO
                                                            U.185:  .BLKB   0
                                   4C 4F 42 5F 45 56 4F 4D  0016B ;TPA$KEYST
                                                            U.187:  .ASCII  \MOVE_BOL\
                                                         FF 00173           .BYTE   -1
                                                            00174 ;TPA$KEYSTO
                                                            U.191:  .BLKB   0
                             44 52 4F 57 5F 54 55 4F 42 55 52  00174 ;TPA$KEYST
                                                            U.193:  .ASCII  \RUBOUT_WORD\
                                                         FF 0017F           .BYTE   -1
                                                            00180 ;TPA$KEYSTO
                                                            U.197:  .BLKB   0
                48 53 45 52 46 45 52 5F 4E 45 45 52 43 53  00180 ;TPA$KEYST
                                                            U.199:  .ASCII  \SCREEN_REFRESH\
                                                         FF 0018E           .BYTE   -1
                                                            0018F ;TPA$KEYSTO
                                                            U.203:  .BLKB   0
                   54 45 53 45 52 5F 4E 4F 49 53 53 45 53  0018F ;TPA$KEYST
```

```
                                                             U.205:   .ASCII   \SESSION_RESET\                 ;
                                                 FF 0019C             .BYTE    -1                              ;
                                                    0019D ;TPA$KEYSTO
                                                          U.209:   .BLKB    0
                  4C  4F  42  5F  54  55  4F  42  55  52 0019D ;TPA$KEYST
                                                          U.211:   .ASCII   \RUBOUT_BOL\                    ;
                                                 FF 001A7             .BYTE    -1                              ;
                                                    001A8 ;TPA$KEYSTO
                                                          U.215:   .BLKB    0
   45  4E  49  4C  5F  45  54  45  4C  45  44  4E  55 001A8 ;TPA$KEYST
                                                          U.217:   .ASCII   \UNDELETE_LINE\                 ;
                                                 FF 001B5             .BYTE    -1                              ;
                                                    001B6 ;TPA$KEYSTO
                                                          U.221:   .BLKB    0
                                    54  49  58  45 001B6 ;TPA$KEYST
                                                          U.223:   .ASCII   \EXIT\                          ;
                                                 FF 001BA             .BYTE    -1                              ;
                                                    001BB ;TPA$KEYSTO
                                                          U.227:   .BLKB    0
       4E  4F  49  53  53  45  53  5F  54  49  55  51 001BB ;TPA$KEYST
                                                          U.229:   .ASCII   \QUIT_SESSION\                  ;
                                                 FF 001C7             .BYTE    -1                              ;
                                                    001C8 ;TPA$KEYSTO
                                                          U.233:   .BLKB    0
45  54  43  41  52  41  48  43  5F  54  55  4F  42  55  52 001C8 ;TPA$KEYST
                                                          U.235:   .ASCII   \RUBOUT_CHARACTER\              ;
                                                 52 001D7                                                      ;
                                                 FF 001D8             .BYTE    -1                              ;
                                                 FF 001D9 ;TPA$KEYFILL                                         ;
                                                          U.239:   .BYTE    -1                              ;
                                                    001DA ;TPA$KEYSTO
                                                          U.242:   .BLKB    0
                                            53  41 001DA ;TPA$KEYST
                                                          U.244:   .ASCII   \AS\                            ;
                                                 FF 001DC             .BYTE    -1                              ;
                                                 FF 001DD ;TPA$KEYFILL
                                                          U.246:   .BYTE    -1                              ;
                                                    001DE ;TPA$KEYSTO
                                                          U.249:   .BLKB    0
                                    44  4C  4F  47 001DE ;TPA$KEYST
                                                          U.251:   .ASCII   \GOLD\                          ;
                                                 FF 001E2             .BYTE    -1                              ;
                                                    001E3 ;TPA$KEYSTO
                                                          U.255:   .BLKB    0
                        4C  4F  52  54  4E  4F  43 001E3 ;TPA$KEYST
                                                          U.257:   .ASCII   \CONTROL\                       ;
                                                 FF 001EA             .BYTE    -1                              ;
                                                    001EB ;TPA$KEYSTO
                                                          U.263:   .BLKB    0
                            45  50  41  43  53  45 001EB ;TPA$KEYST
                                                          U.265:   .ASCII   \ESCAPE\                        ;
                                                 FF 001F1             .BYTE    -1                              ;
                                                    001F2 ;TPA$KEYSTO
                                                          U.270:   .BLKB    0
                                        49  53  43 001F2 ;TPA$KEYST
                                                          U.272:   .ASCII   \CSI\                           ;
                                                 FF 001F5             .BYTE    -1                              ;
```

```
                                001F6 ;TPA$KEYSTO
                                      U.277:    .BLKB   0
                          33    53  53  001F6 ;TPA$KEYST
                                      U.279:    .ASCII  \SS3\                   ;
                              FF  001F9         .BYTE   -1                      ;
                                001FA ;TPA$KEYSTO
                                      U.284:    .BLKB   0
                54    55    4F  42  55  52  001FA ;TPA$KEYST
                                      U.286:    .ASCII  \RUBOUT\                ;
                              FF  00200         .BYTE   -1                      ;
                              FF  00201 ;TPA$KEYFILL
                                      U.291:    .BYTE   -1                      ;
                                00202 ;TPA$KEYSTO
                                      U.296:    .BLKB   0
                            52  4F  00202 ;TPA$KEYST
                                      U.298:    .ASCII  \OR\                    ;
                              FF  00204         .BYTE   -1                      ;
                              FF  00205 ;TPA$KEYFILL
                                      U.305:    .BYTE   -1                      ;

                                                .PSECT  _LIB$STATE$,NOWRT,  SHR,  PIC,1

                                00000 KEYDEF_STATE::
                                                .BLKB   0
                                00000 SWALLOW_1:
                                                .BLKB   0
                             11F2  00000 ;TPA$TYPE
                                      U.2:      .WORD   4594                    ;
                           0000*  00002 ;TPA$TARGET
                                      U.3:      .WORD   <<SWALLOW_1-U.3>-2>     ;
                             0500  00004 ;TPA$TYPE
                                      U.7:      .WORD   1280                    ;
                                00006 SWALLOW_2:
                                                .BLKB   0
                             11F2  00006 ;TPA$TYPE
                                      U.9:      .WORD   4594                    ;
                           0000*  00008 ;TPA$TARGET
                                      U.10:     .WORD   <<SWALLOW_2-U.10>-2>    ;
                             6101  0000A ;TPA$TYPE
                                      U.14:     .WORD   24833                   ;
                        00000000*  0000C ;TPA$ADDR
                                      U.15:     .LONG   <<KEY_ACTION-U.15>-4>   ;
                        00000001   00010 ;TPA$MASK
                                      U.16:     .LONG   1                       ;
                             6102  00014 ;TPA$TYPE
                                      U.20:     .WORD   24834                   ;
                        00000000*  00016 ;TPA$ADDR
                                      U.21:     .LONG   <<KEY_ACTION-U.21>-4>   ;
                        00000002   0001A ;TPA$MASK
                                      U.22:     .LONG   2                       ;
                             6103  0001E ;TPA$TYPE
                                      U.26:     .WORD   24835                   ;
                        00000000*  00020 ;TPA$ADDR
                                      U.27:     .LONG   <<KEY_ACTION-U.27>-4>   ;
                        00000003   00024 ;TPA$MASK
                                      U.28:     .LONG   3                       ;
                             6104  00028 ;TPA$TYPE
```

```
                        U.32:   .WORD   24836                           ;
00000000* 0002A         ;TPA$ADDR
                        U.33:   .LONG   <<KEY_ACTION-U.33>-4>           ;
00000004  0002E         ;TPA$MASK
                        U.34:   .LONG   4                               ;
    6105  00032         ;TPA$TYPE
                        U.38:   .WORD   24837                           ;
00000000* 00034         ;TPA$ADDR
                        U.39:   .LONG   <<KEY_ACTION-U.39>-4>           ;
00000005  00038         ;TPA$MASK
                        U.40:   .LONG   5                               ;
    6106  0003C         ;TPA$TYPE
                        U.44:   .WORD   24838                           ;
00000000* 0003E         ;TPA$ADDR
                        U.45:   .LONG   <<KEY_ACTION-U.45>-4>           ;
00000006  00042         ;TPA$MASK
                        U.46:   .LONG   6                               ;
    6107  00046         ;TPA$TYPE
                        U.50:   .WORD   24839                           ;
00000000* 00048         ;TPA$ADDR
                        U.51:   .LONG   <<KEY_ACTION-U.51>-4>           ;
00000007  0004C         ;TPA$MASK
                        U.52:   .LONG   7                               ;
    6108  00050         ;TPA$TYPE
                        U.56:   .WORD   24840                           ;
00000000* 00052         ;TPA$ADDR
                        U.57:   .LONG   <<KEY_ACTION-U.57>-4>           ;
00000008  00056         ;TPA$MASK
                        U.58:   .LONG   8                               ;
    6109  0005A         ;TPA$TYPE
                        U.62:   .WORD   24841                           ;
00000000* 0005C         ;TPA$ADDR
                        U.63:   .LONG   <<KEY_ACTION-U.63>-4>           ;
00000009  00060         ;TPA$MASK
                        U.64:   .LONG   9                               ;
    610A  00064         ;TPA$TYPE
                        U.68:   .WORD   24842                           ;
00000000* 00066         ;TPA$ADDR
                        U.69:   .LONG   <<KEY_ACTION-U.69>-4>           ;
0000000A  0006A         ;TPA$MASK
                        U.70:   .LONG   10                              ;
    610B  0006E         ;TPA$TYPE
                        U.74:   .WORD   24843                           ;
00000000* 00070         ;TPA$ADDR
                        U.75:   .LONG   <<KEY_ACTION-U.75>-4>           ;
0000000B  00074         ;TPA$MASK
                        U.76:   .LONG   11                              ;
    610C  00078         ;TPA$TYPE
                        U.80:   .WORD   24844                           ;
00000000* 0007A         ;TPA$ADDR
                        U.81:   .LONG   <<KEY_ACTION-U.81>-4>           ;
0000000C  0007E         ;TPA$MASK
                        U.82:   .LONG   12                              ;
    610D  00082         ;TPA$TYPE
                        U.86:   .WORD   24845                           ;
00000000* 00084         ;TPA$ADDR
                        U.87:   .LONG   <<KEY_ACTION-U.87>-4>           ;
```

```
0000000E  00088 ;TPA$MASK
                U.88:   .LONG   14                                      ;
    610E  0008C ;TPA$TYPE
                U.92:   .WORD   24846                                   ;
00000000* 0008E ;TPA$ADDR
                U.93:   .LONG   <<KEY_ACTION-U.93>-4>                   ;
00000010  00092 ;TPA$MASK
                U.94:   .LONG   16                                      ;
    610F  00096 ;TPA$TYPE
                U.98:   .WORD   24847                                   ;
00000000* 00098 ;TPA$ADDR
                U.99:   .LONG   <<KEY_ACTION-U.99>-4>                   ;
00000011  0009C ;TPA$MASK
                U.100:  .LONG   17                                      ;
    6110  000A0 ;TPA$TYPE
                U.104:  .WORD   24848                                   ;
00000000* 000A2 ;TPA$ADDR
                U.105:  .LONG   <<KEY_ACTION-U.105>-4>                  ;
00000012  000A6 ;TPA$MASK
                U.106:  .LONG   18                                      ;
    6111  000AA ;TPA$TYPE
                U.110:  .WORD   24849                                   ;
00000000* 000AC ;TPA$ADDR
                U.111:  .LONG   <<KEY_ACTION-U.111>-4>                  ;
00000013  000B0 ;TPA$MASK
                U.112:  .LONG   19                                      ;
    6112  000B4 ;TPA$TYPE
                U.116:  .WORD   24850                                   ;
00000000* 000B6 ;TPA$ADDR
                U.117:  .LONG   <<KEY_ACTION-U.117>-4>                  ;
00000014  000BA ;TPA$MASK
                U.118:  .LONG   20                                      ;
    6113  000BE ;TPA$TYPE
                U.122:  .WORD   24851                                   ;
00000000* 000C0 ;TPA$ADDR
                U.123:  .LONG   <<KEY_ACTION-U.123>-4>                  ;
00000015  000C4 ;TPA$MASK
                U.124:  .LONG   21                                      ;
    6114  000C8 ;TPA$TYPE
                U.128:  .WORD   24852                                   ;
00000000* 000CA ;TPA$ADDR
                U.129:  .LONG   <<KEY_ACTION-U.129>-4>                  ;
00000016  000CE ;TPA$MASK
                U.130:  .LONG   22                                      ;
    6115  000D2 ;TPA$TYPE
                U.134:  .WORD   24853                                   ;
00000000* 000D4 ;TPA$ADDR
                U.135:  .LONG   <<KEY_ACTION-U.135>-4>                  ;
00000017  000D8 ;TPA$MASK
                U.136:  .LONG   23                                      ;
    6116  000DC ;TPA$TYPE
                U.140:  .WORD   24854                                   ;
00000000* 000DE ;TPA$ADDR
                U.141:  .LONG   <<KEY_ACTION-U.141>-4>                  ;
00000018  000E2 ;TPA$MASK
                U.142:  .LONG   24                                      ;
    6117  000E6 ;TPA$TYPE
```

```
                        U.146:  .WORD   24855                           ;
00000000* 000E8 ;TPA$ADDR
                        U.147:  .LONG   <<KEY_ACTION-U.147>-4>          ;
00000019  000EC ;TPA$MASK
                        U.148:  .LONG   25                              ;
    6118  000F0 ;TPA$TYPE
                        U.152:  .WORD   24856                           ;
00000000* 000F2 ;TPA$ADDR
                        U.153:  .LONG   <<KEY_ACTION-U.153>-4>          ;
0000001A  000F6 ;TPA$MASK
                        U.154:  .LONG   26                              ;
    6119  000FA ;TPA$TYPE
                        U.158:  .WORD   24857                           ;
00000000* 000FC ;TPA$ADDR
                        U.159:  .LONG   <<KEY_ACTION-U.159>-4>          ;
0000001B  00100 ;TPA$MASK
                        U.160:  .LONG   27                              ;
    611A  00104 ;TPA$TYPE
                        U.164:  .WORD   24858                           ;
00000000* 00106 ;TPA$ADDR
                        U.165:  .LONG   <<KEY_ACTION-U.165>-4>          ;
0000001C  0010A ;TPA$MASK
                        U.166:  .LONG   28                              ;
    611B  0010E ;TPA$TYPE
                        U.170:  .WORD   24859                           ;
00000000* 00110 ;TPA$ADDR
                        U.171:  .LONG   <<KEY_ACTION-U.171>-4>          ;
0000001D  00114 ;TPA$MASK
                        U.172:  .LONG   29                              ;
    611C  00118 ;TPA$TYPE
                        U.176:  .WORD   24860                           ;
00000000* 0011A ;TPA$ADDR
                        U.177:  .LONG   <<KEY_ACTION-U.177>-4>          ;
0000001E  0011E ;TPA$MASK
                        U.178:  .LONG   30                              ;
    611D  00122 ;TPA$TYPE
                        U.182:  .WORD   24861                           ;
00000000* 00124 ;TPA$ADDR
                        U.183:  .LONG   <<KEY_ACTION-U.183>-4>          ;
0000001F  00128 ;TPA$MASK
                        U.184:  .LONG   31                              ;
    611E  0012C ;TPA$TYPE
                        U.188:  .WORD   24862                           ;
00000000* 0012E ;TPA$ADDR
                        U.189:  .LONG   <<KEY_ACTION-U.189>-4>          ;
00000021  00132 ;TPA$MASK
                        U.190:  .LONG   33                              ;
    611F  00136 ;TPA$TYPE
                        U.194:  .WORD   24863                           ;
00000000* 00138 ;TPA$ADDR
                        U.195:  .LONG   <<KEY_ACTION-U.195>-4>          ;
00000022  0013C ;TPA$MASK
                        U.196:  .LONG   34                              ;
    6120  00140 ;TPA$TYPE
                        U.200:  .WORD   24864                           ;
00000000* 00142 ;TPA$ADDR
                        U.201:  .LONG   <<KEY_ACTION-U.201>-4>          ;
```

```
00000025  00146 ;TPA$MASK
                U.202:  .LONG   37                                      ;
    6121  0014A ;TPA$TYPE
                U.206:  .WORD   24865                                   ;
00000000* 0014C ;TPA$ADDR
                U.207:  .LONG   <<KEY_ACTION-U.207>-4>                  ;
00000026  00150 ;TPA$MASK
                U.208:  .LONG   38                                      ;
    6122  00154 ;TPA$TYPE
                U.212:  .WORD   24866                                   ;
00000000* 00156 ;TPA$ADDR
                U.213:  .LONG   <<KEY_ACTION-U.213>-4>                  ;
00000023  0015A ;TPA$MASK
                U.214:  .LONG   35                                      ;
    6123  0015E ;TPA$TYPE
                U.218:  .WORD   24867                                   ;
00000000* 00160 ;TPA$ADDR
                U.219:  .LONG   <<KEY_ACTION-U.219>-4>                  ;
00000024  00164 ;TPA$MASK
                U.220:  .LONG   36                                      ;
    6124  00168 ;TPA$TYPE
                U.224:  .WORD   24868                                   ;
00000000* 0016A ;TPA$ADDR
                U.225:  .LONG   <<KEY_ACTION-U.225>-4>                  ;
00000027  0016E ;TPA$MASK
                U.226:  .LONG   39                                      ;
    6125  00172 ;TPA$TYPE
                U.230:  .WORD   24869                                   ;
00000000* 00174 ;TPA$ADDR
                U.231:  .LONG   <<KEY_ACTION-U.231>-4>                  ;
00000028  00178 ;TPA$MASK
                U.232:  .LONG   40                                      ;
    6526  0017C ;TPA$TYPE
                U.236:  .WORD   25894                                   ;
00000000* 0017E ;TPA$ADDR
                U.237:  .LONG   <<KEY_ACTION-U.237>-4>                  ;
00000029  00182 ;TPA$MASK
                U.238:  .LONG   41                                      ;
          00186 SWALLOW_3:
                .BLKB   0
    11F2  00186 ;TPA$TYPE
                U.240:  .WORD   4594                                    ;
    0000* 00188 ;TPA$TARGET
                U.241:  .WORD   <<SWALLOW_3-U.241>-2>                   ;
    0527  0018A ;TPA$TYPE
                U.245:  .WORD   1319                                    ;
          0018C KEY_DEFINE:
                .BLKB   0
    11F2  0018C ;TPA$TYPE
                U.247:  .WORD   4594                                    ;
    0000* 0018E ;TPA$TARGET
                U.248:  .WORD   <<KEY_DEFINE-U.248>-2>                  ;
    6128  00190 ;TPA$TYPE
                U.252:  .WORD   24872                                   ;
00000000* 00192 ;TPA$ADDR
                U.253:  .LONG   <<KEY_FLAGS-U.253>-4>                   ;
00000004  00196 ;TPA$MASK
```

```
                        U.254:  .LONG   4                                       ;
    7129  0019A ;TPA$TYPE
                        U.258:  .WORD   28969                                   ;
00000000* 0019C ;TPA$ADDR
                        U.259:  .LONG   <<KEY_FLAGS-U.259>-4>                   ;
00000008  001A0 ;TPA$MASK
                        U.260:  .LONG   8                                       ;
    0000* 001A4 ;TPA$TARGET
                        U.262:  .WORD   <<U.261-U.262>-2>                       ;
    712A  001A6 ;TPA$TYPE
                        U.266:  .WORD   28970                                   ;
00000000* 001A8 ;TPA$ADDR
                        U.267:  .LONG   <<KEY_FLAGS-U.267>-4>                   ;
00000010  001AC ;TPA$MASK
                        U.268:  .LONG   16                                      ;
    0000* 001B0 ;TPA$TARGET
                        U.269:  .WORD   <<U.261-U.269>-2>                       ;
    712B  001B2 ;TPA$TYPE
                        U.273:  .WORD   28971                                   ;
00000000* 001B4 ;TPA$ADDR
                        U.274:  .LONG   <<KEY_FLAGS-U.274>-4>                   ;
00000001  001B8 ;TPA$MASK
                        U.275:  .LONG   1                                       ;
    0000* 001BC ;TPA$TARGET
                        U.276:  .WORD   <<U.261-U.276>-2>                       ;
    712C  001BE ;TPA$TYPE
                        U.280:  .WORD   28972                                   ;
00000000* 001C0 ;TPA$ADDR
                        U.281:  .LONG   <<KEY_FLAGS-U.281>-4>                   ;
00000002  001C4 ;TPA$MASK
                        U.282:  .LONG   2                                       ;
    0000* 001C8 ;TPA$TARGET
                        U.283:  .WORD   <<U.261-U.283>-2>                       ;
    812D  001CA ;TPA$TYPE
                        U.287:  .WORD   -32467                                  ;
00000000V 001CC ;TPA$ACTION
                        U.288:  .LONG   <<SET_RUBOUT-U.288>-4>                  ;
    15F7  001D0 ;TPA$TYPE
                        U.289:  .WORD   5623                                    ;
    FFFE  001D2 ;TPA$TARGET
                        U.290:  .WORD   -2                                      ;
          001D4 CHECK_END:
                        .BLKB   0
    11F2  001D4 ;TPA$TYPE
                        U.292:  .WORD   4594                                    ;
    0000* 001D6 ;TPA$TARGET
                        U.293:  .WORD   <<CHECK_END-U.293>-2>                   ;
    102C  001D8 ;TPA$TYPE
                        U.294:  .WORD   4140                                    ;
    0000* 001DA ;TPA$TARGET
                        U.295:  .WORD   <<KEY_DEFINE-U.295>-2>                  ;
    912E  001DC ;TPA$TYPE
                        U.299:  .WORD   -28370                                  ;
00000000V 001DE ;TPA$ACTION
                        U.300:  .LONG   <<SET_DEFINITION-U.300>-4>              ;
    0000* 001E2 ;TPA$TARGET
                        U.301:  .WORD   <<KEY_DEFINE-U.301>-2>                  ;
```

```
    95F7  001E4 ;TPA$TYPE
                U.302:  .WORD   -27145                                  ;
00000000V 001E6 ;TPA$ACTION
                U.303:  .LONG   <<SET_DEFINITION-U.303>-4>              ;
    FFFF  001EA ;TPA$TARGET
                U.304:  .WORD   -1                                      ;
          001EC ;GET_TEXT
                U.26T:  .BLKB   0
    11F2  001EC ;TPA$TYPE
                U.306:  .WORD   4594                                    ;
    0000* 001EE ;TPA$TARGET
                U.307:  .WORD   <<U.261-U.307>-2>                       ;
    042C  001F0 ;TPA$TYPE
                U.308:  .WORD   1068                                    ;
          001F2 SWALLOW_4:
                        .BLKB   0
    11F2  001F2 ;TPA$TYPE
                U.309:  .WORD   4594                                    ;
    0000* 001F4 ;TPA$TARGET
                U.310:  .WORD   <<SWALLOW_4-U.310>-2>                   ;
    5DF8  001F6 ;TPA$TYPE
                U.311:  .WORD   24056                                   ;
    0000* 001F8 ;TPA$SUBEXP
                U.313:  .WORD   <<U.312-U.313>-2>                       ;
00000000* 001FA ;TPA$ADDR
                U.314:  .LONG   <<KEY_STRING-U.314>-4>                  ;
    0000* 001FE ;TPA$TARGET
                U.315:  .WORD   <<CHECK_END-U.315>-2>                   ;
          00200 ;GET_STRING
                U.312:  .BLKB   0
    19F8  00200 ;TPA$TYPE
                U.316:  .WORD   6648                                    ;
    0000* 00202 ;TPA$SUBEXP
                U.318:  .WORD   <<U.317-U.318>-2>                       ;
    0000* 00204 ;TPA$TARGET
                U.319:  .WORD   <<U.312-U.319>-2>                       ;
    15F6  00206 ;TPA$TYPE
                U.320:  .WORD   5622                                    ;
    FFFF  00208 ;TPA$TARGET
                U.321:  .WORD   -1                                      ;
          0020A ;CHECK_DELIM
                U.317:  .BLKB   0
    102C  0020A ;TPA$TYPE
                U.322:  .WORD   4140                                    ;
    FFFE  0020C ;TPA$TARGET
                U.323:  .WORD   -2                                      ;
    1020  0020E ;TPA$TYPE
                U.324:  .WORD   4128                                    ;
    FFFE  00210 ;TPA$TARGET
                U.325:  .WORD   -2                                      ;
    11F7  00212 ;TPA$TYPE
                U.326:  .WORD   4599                                    ;
    FFFE  00214 ;TPA$TARGET
                U.327:  .WORD   -2                                      ;
    15ED  00216 ;TPA$TYPE
                U.328:  .WORD   5613                                    ;
    FFFF  00218 ;TPA$TARGET
```

```
                U.329:  .WORD   -1                                      ;

                        .PSECT  _LIB$KEYO$,NOWRT,  SHR,  PIC,1

      00000 KEYDEF_KEY::
                        .BLKB   0
      00000 ;TPA$KEYO
                U.1:    .BLKB   0
0000* 00000 ;TPA$KEY
                U.5:    .WORD   <U.4-U.1>                               ;
0000* 00002 ;TPA$KEY
                U.12:   .WORD   <U.11-U.1>                              ;
0000* 00004 ;TPA$KEY
                U.18:   .WORD   <U.17-U.1>                              ;
0000* 00006 ;TPA$KEY
                U.24:   .WORD   <U.23-U.1>                              ;
0000* 00008 ;TPA$KEY
                U.30:   .WORD   <U.29-U.1>                              ;
0000* 0000A ;TPA$KEY
                U.36:   .WORD   <U.35-U.1>                              ;
0000* 0000C ;TPA$KEY
                U.42:   .WORD   <U.41-U.1>                              ;
0000* 0000E ;TPA$KEY
                U.48:   .WORD   <U.47-U.1>                              ;
0000* 00010 ;TPA$KEY
                U.54:   .WORD   <U.53-U.1>                              ;
0000* 00012 ;TPA$KEY
                U.60:   .WORD   <U.59-U.1>                              ;
0000* 00014 ;TPA$KEY
                U.66:   .WORD   <U.65-U.1>                              ;
0000* 00016 ;TPA$KEY
                U.72:   .WORD   <U.71-U.1>                              ;
0000* 00018 ;TPA$KEY
                U.78:   .WORD   <U.77-U.1>                              ;
0000* 0001A ;TPA$KEY
                U.84:   .WORD   <U.83-U.1>                              ;
0000* 0001C ;TPA$KEY
                U.90:   .WORD   <U.89-U.1>                              ;
0000* 0001E ;TPA$KEY
                U.96:   .WORD   <U.95-U.1>                              ;
0000* 00020 ;TPA$KEY
                U.102:  .WORD   <U.101-U.1>                             ;
0000* 00022 ;TPA$KEY
                U.108:  .WORD   <U.107-U.1>                             ;
0000* 00024 ;TPA$KEY
                U.114:  .WORD   <U.113-U.1>                             ;
0000* 00026 ;TPA$KEY
                U.120:  .WORD   <U.119-U.1>                             ;
0000* 00028 ;TPA$KEY
                U.126:  .WORD   <U.125-U.1>                             ;
0000* 0002A ;TPA$KEY
                U.132:  .WORD   <U.131-U.1>                             ;
0000* 0002C ;TPA$KEY
                U.138:  .WORD   <U.137-U.1>                             ;
0000* 0002E ;TPA$KEY
                U.144:  .WORD   <U.143-U.1>                             ;
0000* 00030 ;TPA$KEY
```

```
                U.150:  .WORD   <U.149-U.1>                     ;
0000* 00032 ;TPA$KEY
                U.156:  .WORD   <U.155-U.1>                     ;
0000* 00034 ;TPA$KEY
                U.162:  .WORD   <U.161-U.1>                     ;
0000* 00036 ;TPA$KEY
                U.168:  .WORD   <U.167-U.1>                     ;
0000* 00038 ;TPA$KEY
                U.174:  .WORD   <U.173-U.1>                     ;
0000* 0003A ;TPA$KEY
                U.180:  .WORD   <U.179-U.1>                     ;
0000* 0003C ;TPA$KEY
                U.186:  .WORD   <U.185-U.1>                     ;
0000* 0003E ;TPA$KEY
                U.192:  .WORD   <U.191-U.1>                     ;
0000* 00040 ;TPA$KEY
                U.198:  .WORD   <U.197-U.1>                     ;
0000* 00042 ;TPA$KEY
                U.204:  .WORD   <U.203-U.1>                     ;
0000* 00044 ;TPA$KEY
                U.210:  .WORD   <U.209-U.1>                     ;
0000* 00046 ;TPA$KEY
                U.216:  .WORD   <U.215-U.1>                     ;
0000* 00048 ;TPA$KEY
                U.222:  .WORD   <U.221-U.1>                     ;
0000* 0004A ;TPA$KEY
                U.228:  .WORD   <U.227-U.1>                     ;
0000* 0004C ;TPA$KEY
                U.234:  .WORD   <U.233-U.1>                     ;
0000* 0004E ;TPA$KEY
                U.243:  .WORD   <U.242-U.1>                     ;
0000* 00050 ;TPA$KEY
                U.250:  .WORD   <U.249-U.1>                     ;
0000* 00052 ;TPA$KEY
                U.256:  .WORD   <U.255-U.1>                     ;
0000* 00054 ;TPA$KEY
                U.264:  .WORD   <U.263-U.1>                     ;
0000* 00056 ;TPA$KEY
                U.271:  .WORD   <U.270-U.1>                     ;
0000* 00058 ;TPA$KEY
                U.278:  .WORD   <U.277-U.1>                     ;
0000* 0005A ;TPA$KEY
                U.285:  .WORD   <U.284-U.1>                     ;
0000* 0005C ;TPA$KEY
                U.297:  .WORD   <U.296-U.1>                     ;

                        .PSECT  AED_COMMON,NOEXE,  OVR,0

        00000 AED_L_FLAGS:
                        .BLKB   4
        00004 AED_B_OPTIONS:
                        .BLKB   1
        00005           .BLKB   3
        00008 AED_L_OBJTYP:
                        .BLKB   4
        0000C AED_Q_OBJNAM:
                        .BLKB   8
```

```
00014 AED_L_WORSTERR:
                .BLKB   4
00018 AED_L_PAGEWIDTH:
                .BLKB   4
0001C AED_L_PAGESIZE:
                .BLKB   4
00020 AED_B_COLUMN:
                .BLKB   1
00021           .BLKB   3
00024 AED_B_LINE:
                .BLKB   1
00025           .BLKB   3
00028 AED_B_SAVE_COL:
                .BLKB   1
00029           .BLKB   3
0002C AED_B_SAVE_LIN:
                .BLKB   1
0002D           .BLKB   3
00030 AED_Q_LINETABLE:
                .BLKB   12
0003C AED_L_CURACE:
                .BLKB   4
00040 AED_L_FIRSTLINE:
                .BLKB   4
00044 AED_L_LASTLINE:
                .BLKB   4
00048 AED_L_BEGINLINE:
                .BLKB   4
0004C AED_W_INPUTLEN:
                .BLKB   2
0004E           .BLKB   2
00050 AED_Q_DEL_ACE:
                .BLKB   8
00058 AED_Q_DEL_LINE:
                .BLKB   8
00060 AED_Q_DEL_WORD:
                .BLKB   8
00068 AED_B_DEL_CHAR:
                .BLKB   1
00069           .BLKB   3
0006C AED_A_ACLBUFFER:
                .BLKB   4
00070 AED_Q_OUTLINE:
                .BLKB   8
00078 AED_W_OBJCHAN:
                .BLKB   2
0007A           .BLKB   2
0007C AED_W_TERMIN:
                .BLKB   2
0007E           .BLKB   2
00080 AED_W_TERMOUT:
                .BLKB   2
00082           .BLKB   2
00084 AED_W_IOSB:
                .BLKB   8
0008C AED_L_STATUS:
                .BLKB   4
```

```
00090 AED_B_FIELD:
                .BLKB   1
00091           .BLKB   3
00094 AED_W_FIELDBEG:
                .BLKB   2
00096           .BLKB   2
00098 AED_W_FIELDEND:
                .BLKB   2
0009A           .BLKB   2
0009C AED_B_ITEM:
                .BLKB   1
0009D           .BLKB   3
000A0 AED_W_ITEMBEG:
                .BLKB   2
000A2           .BLKB   2
000A4 AED_W_ITEMEND:
                .BLKB   2
000A6           .BLKB   2
000A8 AED_B_ACETYPE:
                .BLKB   1
000A9           .BLKB   3
000AC AED_W_JOURNAL:
                .BLKB   2
000AE           .BLKB   2
000B0 AED_T_CURLINE:
                .BLKB   532
002C4 AED_W_TOTALSIZE:
                .BLKB   2
002C6           .BLKB   2
002C8 JOURNAL_FAB:
                .BLKB   80
00318 JOURNAL_NAM:
                .BLKB   96
00378 JOURNAL_RAB:
                .BLKB   68
003BC JOURNAL_XABPRO:
                .BLKB   88
00414 JOURNAL_BUFFER:
                .BLKB   10
0041E           .BLKB   2
00420 JOURNAL_INDEX:
                .BLKB   4
00424 RECOVER_FAB:
                .BLKB   80
00474 RECOVER_NAM:
                .BLKB   96
004D4 RECOVER_RAB:
                .BLKB   68
00518 RECOVER_BUFFER:
                .BLKB   10
00522           .BLKB   2
00524 RECOVER_INDEX:
                .BLKB   4

                .PSECT  $PLIT$,NOWRT,NOEXE,2

00 00 3A 54 49 4E 49 24 54 49 44 45 4C 43 41  00000 P.AAA:  .ASCII  \ACLEDIT$INIT:\<0><0><0>                    ;
```

```
00  0000F                                                                   ;

                        .PSECT  $OWN$,NOEXE,2

    00000 KEY_BLOCK:
                        .BLKB   11
    0000B               .BLKB   1
    0000C KEY_STRING:
                        .BLKB   8

                KEY_ACTION=         KEY_BLOCK+8
                KEY_FLAGS=          KEY_BLOCK+10
                        .EXTRN  CLI$GET_VALUE, CLI$PRESENT
                        .EXTRN  LIB$FREE_VM, LIB$GET_VM
                        .EXTRN  LIB$TPARSE, SCR$DOWN_SCROLL
                        .EXTRN  SCR$ERASE_LINE, SCR$ERASE_PAGE
                        .EXTRN  SCR$SET_CURSOR, SCR$SET_SCROLL
                        .EXTRN  SCR$UP_SCROLL, AED$_OBJLOCKED
                        .EXTRN  AED$_BADKEEP, AED$_LOCATERR
                        .EXTRN  AED$_INIREADERR
                        .EXTRN  AED$_JOUWRITERR
                        .EXTRN  AED$_JOUOPENOUT
                        .EXTRN  AED$_JOUCLOSOUT
                        .EXTRN  AED$_RECREADERR
                        .EXTRN  AED$_RECOPENIN, AED$_RECLOSEIN
                        .EXTRN  AED$_BADUIC, AED$_BADGRPMEM
                        .EXTRN  AED$_SYNTAX, AED$_BADTYPE
                        .EXTRN  AED$_NOITEMSEL, AED$_MUSTENTER
                        .EXTRN  AED$_INIOPENIN, AED$_INICLOSIN
                        .EXTRN  AED$_DEFSYNTAX, AED$_NODELETE
                        .EXTRN  AED$_NOMODIFY, AED$_NOHIDDEN
                        .EXTRN  AED$_DUPLICATE, AED$_NOCOMBINE
                        .EXTRN  AED$_NODEFAULT, AED$_NOCTRLCHAR
                        .EXTRN  AED$_NOTFOUND, AED$_CONTROL_C
                        .EXTRN  AED$_ACLUPDATED
                        .EXTRN  AED$_NOCHANGE, AED_FILERROR
                        .EXTRN  AED_PUTOUTPUT, AED_SET_CURSOR
                        .EXTRN  KEY_TABLE, SYS$OPEN
                        .EXTRN  SYS$CONNECT, SYS$GET
                        .EXTRN  LIB$SIGNAL

                        .PSECT  $CODE$,NOWRT,2

                                    0FFC 00000      .ENTRY  AED_GETKEYINI, Save R2,R3,R4,R5,R6,R7,R8,-  ; 0645
                                                            R9,R10,R11
                            5B 00000000G   00 9E 00002  MOVAB   LIB$SIGNAL, R11                         ;
                            5A 00000000G   00 9E 00009  MOVAB   SCR$ERASE_PAGE, R10                     ;
                            59 00000000G   8F D0 00010  MOVL    #AED$_INIOPENIN, R9                     ;
                            58 00000000G   8F D0 00017  MOVL    #AED$_DEFSYNTAX, R8                     ;
                            57 00000000G   00 9E 0001E  MOVAB   SCR$SET_CURSOR, R7                      ;
                            56      0000'  CF 9E 00025  MOVAB   AED_L_WORSTERR, R6                      ;
                            5E       FAE8  CE 9E 0002A  MOVAB   -1304(SP), SP                           ;
    0050    8F      00          6E         00 2C 0002F  MOVC5   #0, (SP), #0, #80, $RMS_PTR             ; 0704
                                    B0     AD    00036                                                  ;
                        B0  AD      5003   8F B0 00038  MOVW    #20483, $RMS_PTR                        ;
                        B4  AD        40   8F 9A 0003E  MOVZBL  #64, $RMS_PTR+4                         ;
                        C6  AD             02 90 00043  MOVB    #2, $RMS_PTR+22                         ;
```

```
                                CD   AD  94 00047          CLRB    $RMS_PTR+29                          ;
                CF   AD              02  90 0004A          MOVB    #2, $RMS_PTR+31                      ;
                D8   AD       FF0C   CD  9E 0004E          MOVAB   KEYINI_NAM, $RMS_PTR+40              ;
                DC   AD      0000'   CF  9E 00054          MOVAB   P.AAA, $RMS_PTR+44                   ;
                E4   AD              0D  90 0005A          MOVB    #13, $RMS_PTR+52                     ;
0060   8F          00   6E           00  2C 0005E          MOVC5   #0, (SP), #0, #96, $RMS_PTR          ; 0709
                              FF0C   CD     00065                                                       ;
              FF0C   CD       6002   8F  B0 00068          MOVW    #24578, $RMS_PTR                     ;
              FF0E   CD              01  8E 0006F          MNEGB   #1, $RMS_PTR+2                       ;
              FF10   CD       0224   CE  9E 00074          MOVAB   KEYINI_RES_NAM, $RMS_PTR+4           ;
              FF16   CD              01  8E 0007B          MNEGB   #1, $RMS_PTR+10                      ;
              FF18   CD       FE0C   CD  9E 00080          MOVAB   KEYINI_EXP_NAM, $RMS_PTR+12          ;
0044   8F          00   6E           00  2C 00087          MOVC5   #0, (SP), #0, #68, $RMS_PTR          ; 0712
                              FF6C   CD     0008E                                                       ;
              FF6C   CD       4401   8F  B0 00091          MOVW    #17409, $RMS_PTR                     ;
                                8A   AD  94 00098          CLRB    $RMS_PTR+30                          ;
                A8   AD         B0   AD  9E 0009B          MOVAB   KEYINI_FAB, $RMS_PTR+60              ;
                                B0   AD  9F 000A0          PUSHAB  KEYINI_FAB                           ; 0720
         00000000G   00              01  FB 000A3          CALLS   #1, SYS$OPEN                         ;
                     25              50  E8 000AA          BLBS    R0, 2$                               ;
          000184C4   8F         B8   AD  D1 000AD          CMPL    KEYINI_FAB+8, #99524                 ; 0723
                     18                  13 000B5          BEQL    1$                                   ;
                     7E         B8   AD  7D 000B7          MOVQ    KEYINI_FAB+8, -(SP)                  ; 0724
                                B0   AD  9F 000BB          PUSHAB  KEYINI_FAB                           ;
                                     59  DD 000BE          PUSHL   R9                                   ;
             0000G   CF              04  FB 000C0          CALLS   #4, AED_FILERROR                     ;
          00018292   8F         B8   AD  D1 000C5          CMPL    KEYINI_FAB+8, #98962                 ; 0726
                                     54  12 000CD          BNEQ    5$                                   ;
                                   016D  31 000CF  1$:     BRW     23$                                  ;
                              FF6C   CD  9F 000D2  2$:     PUSHAB  KEYINI_RAB                           ; 0729
         00000000G   00              01  FB 000D6          CALLS   #1, SYS$CONNECT                      ;
                     0C              50  E8 000DD          BLBS    R0, 3$                               ;
                     7E       FF74   CD  7D 000E0          MOVQ    KEYINI_RAB+8, -(SP)                  ; 0732
                                B0   AD  9F 000E5          PUSHAB  KEYINI_FAB                           ;
                                     59  DD 000E8          PUSHL   R9                                   ;
                                     32  11 000EA          BRB     4$                                   ;
                90   AD         24   AE  9E 000EC  3$:     MOVAB   DEFINE_LINE, KEYINI_RAB+36           ; 0743
                8C   AD       0200   8F  B0 000F1          MOVW    #512, KEYINI_RAB+32                  ; 0744
                              FF6C   CD  9F 000F7          PUSHAB  KEYINI_RAB                           ; 0745
         00000000G   00              01  FB 000FB          CALLS   #1, SYS$GET                          ;
                     22              50  E8 00102          BLBS    R0, 6$                               ;
          0001627A   8F       FF74   CD  D1 00105          CMPL    KEYINI_RAB+8, #98938                 ; 0748
                     BF                  13 0010E          BEQL    1$                                   ;
                     7E       FF74   CD  7D 00110          MOVQ    KEYINI_RAB+8, -(SP)                  ; 0749
                                B0   AD  9F 00115          PUSHAB  KEYINI_FAB                           ;
                         00000000G   8F  DD 00118          PUSHL   #AED$_INIREADERR                     ;
             0000G   CF              04  FB 0011E  4$:     CALLS   #4, AED_FILERROR                     ;
                     50              66  D0 00123  5$:     MOVL    AED_L_WORSTERR, R0                   ; 0751
                                         04 00126          RET                                          ;
                             0000'   CF  94 00127  6$:     CLRB    KEY_ACTION                           ; 0753
                             0000'   CF  94 0012B          CLRB    KEY_FLAGS                            ; 0754
                             0000'   CF  B4 0012F          CLRW    KEY_STRING                           ; 0755
                     21         24   AE  91 00133          CMPB    DEFINE_LINE, #33                     ; 0757
                                     B3  13 00137          BEQL    3$                                   ;
                                     52  D4 00139          CLRL    LINE_INDEX                           ; 0760
 52        8E   AD   10              00  ED 0013B  7$:     CMPZV   #0, #16, KEYINI_RAB+34, LINE_INDEX   ; 0761
                                     03  14 00141          BGTR    8$                                   ;
```

```
                                                0083  31 00143          BRW     17$                                          ;
                            3C            24 AE42  91 00146  8$:      CMPB    DEFINE_LINE[LINE_INDEX], #60                 ; 0764
                                                  62  12 0014B          BNEQ    15$                                          ;
                                                  52  D6 0014D  9$:      INCL    LINE_INDEX                                   ; 0769
                            3E            24 AE42  91 0014F          CMPB    DEFINE_LINE[LINE_INDEX], #62                 ; 0770
                                                  59  13 00154          BEQL    15$                                          ;
                                                  53  D4 00156          CLRL    R3                                           ; 0771
       52        8E    AD               10        00  ED 00158          CMPZV   #0, #16, KEYINI_RAB+34, LINE_INDEX          ;
                                                  4C  14 0015E          BGTR    14$                                          ;
                                                  53  D6 00160          INCL    R3                                           ;
                       0E        EC     A6        03  E1 00162          BBC     #3, AED_L_FLAGS, 10$                         ; 0775
                                                  01  DD 00167          PUSHL   #1                                           ;
                                                  15  DD 00169          PUSHL   #21                                          ;
                                        6A        02  FB 0016B          CALLS   #2, SCR$ERASE_PAGE                           ;
                                                  01  DD 0016E          PUSHL   #1                                           ;
                                                  15  DD 00170          PUSHL   #21                                          ;
                                        67        02  FB 00172          CALLS   #2, SCR$SET_CURSOR                           ;
                                           24     AE  9F 00175  10$:     PUSHAB  DEFINE_LINE                                  ;
                                        7E    8E  AD  3C 00178          MOVZWL  KEYINI_RAB+34, -(SP)                         ;
                                                  02  DD 0017C          PUSHL   #2                                           ;
                                                  58  DD 0017E          PUSHL   R8                                           ;
                                        6B        04  FB 00180          CALLS   #4, LIB$SIGNAL                               ;
                       0B        EC     A6        03  E1 00183          BBC     #3, AED_L_FLAGS, 11$                         ;
                                        7E    0C  A6  9A 00188          MOVZBL  AED_B_COLUMN, -(SP)                          ;
                                        7E    10  A6  9A 0018C          MOVZBL  AED_B_LINE, -(SP)                            ;
                                        67        02  FB 00190          CALLS   #2, SCR$SET_CURSOR                           ;
                                        00000000*  8F  D5 00193  11$:     TSTL    #<AED$_DEFSYNTAX&7>                          ;
                                                  03  12 00199          BNEQ    13$                                          ;
                                                009D  31 0019B  12$:     BRW     22$                                          ;
00000000*  8F         66                03        00  ED 0019E  13$:     CMPZV   #0, #3, AED_L_WORSTERR, #<AED$_DEFSYNTAX&7> ;
                                                  F2  18 001A7          BGEQ    12$                                          ;
                                                008C  31 001A9          BRW     21$                                          ;
                                        9E        53  E9 001AC  14$:     BLBC    R3, 9$                                       ; 0779
                              61        8F  24 AE42  91 001AF  15$:     CMPB    DEFINE_LINE[LINE_INDEX], #97                 ; 0781
                                                  0D  1F 001B5          BLSSU   16$                                          ;
                              7A        8F  24 AE42  91 001B7          CMPB    DEFINE_LINE[LINE_INDEX], #122                ; 0782
                                                  05  1A 001BD          BGTRU   16$                                          ;
                              24 AE42             20  82 001BF          SUBB2   #32, DEFINE_LINE[LINE_INDEX]                 ; 0783
                                                  52  D6 001C4  16$:     INCL    LINE_INDEX                                   ; 0784
                                                FF72  31 001C6          BRW     7$                                           ; 0761
                                        6E        08  D0 001C9  17$:     MOVL    #8, TPARSE_BLOCK                             ; 0786
                              04        AE        03  88 001CC          BISB2   #3, TPARSE_BLOCK+4                           ; 0788
                              08        AE    8E  AD  3C 001D0          MOVZWL  KEYINI_RAB+34, TPARSE_BLOCK+8                ; 0789
                              0C        AE    24  AE  9E 001D5          MOVAB   DEFINE_LINE, TPARSE_BLOCK+12                 ; 0790
                                           0000'  CF  9F 001DA          PUSHAB  KEYDEF_KEY                                   ; 0792
                                           0000'  CF  9F 001DE          PUSHAB  KEYDEF_STATE                                 ;
                                              08  AE  9F 001E2          PUSHAB  TPARSE_BLOCK                                 ;
                  00000000G     00                03  FB 001E5          CALLS   #3, LIB$TPARSE                               ;
                                        54        50  D0 001EC          MOVL    R0, LOCAL_STATUS                             ;
                                        03        54  E9 001EF          BLBC    LOCAL_STATUS, 18$                            ; 0793
                                                FEF7  31 001F2          BRW     3$                                           ;
                       0E        EC     A6        03  E1 001F5  18$:     BBC     #3, AED_L_FLAGS, 19$                         ; 0797
                                                  01  DD 001FA          PUSHL   #1                                           ;
                                                  15  DD 001FC          PUSHL   #21                                          ;
                                        6A        02  FB 001FE          CALLS   #2, SCR$ERASE_PAGE                           ;
                                                  01  DD 00201          PUSHL   #1                                           ;
                                                  15  DD 00203          PUSHL   #21                                          ;
```

```
                         67               02  FB 00205       CALLS   #2, SCR$SET_CURSOR                              ;
                                  0C      AE  DD 00208 19$:  PUSHL   TPARSE_BLOCK+12                                 ;
                                  0C      AE  DD 0020B       PUSHL   TPARSE_BLOCK+8                                  ;
                                          02  DD 0020E       PUSHL   #2                                              ;
                                          58  DD 00210       PUSHL   R8                                              ;
                         6B               04  FB 00212       CALLS   #4, LIB$SIGNAL                                  ;
             0B      EC  A6               03  E1 00215       BBC     #3, AED_L_FLAGS, 20$                            ;
                         7E       0C      A6  9A 0021A       MOVZBL  AED_B_COLUMN, -(SP)                             ;
                         7E       10      A6  9A 0021E       MOVZBL  AED_B_LINE, -(SP)                               ;
                         67               02  FB 00222       CALLS   #2, SCR$SET_CURSOR                              ;
                            00000000*     8F  D5 00225 20$:  TSTL    #<AED$_DEFSYNTAX&7>                             ;
                                          0E  13 0022B       BEQL    22$                                             ;
00000000*  8F        66  03               00  ED 0022D       CMPZV   #0, #3, AED_L_WORSTERR, #<AED$_DEFSYNTAX&7>     ;
                                          03  18 00236       BGEQ    22$                                             ;
                         66               58  D0 00238 21$:  MOVL    R8, AED_L_WORSTERR                              ;
                         50               58  D0 0023B 22$:  MOVL    R8, R0                                          ; 0798
                                              04 0023E       RET                                                     ;
                         50               01  D0 0023F 23$:  MOVL    #1, R0                                          ; 0803
                                              04 00242       RET                                                     ; 0805

; Routine Size:  579 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 ROUTINE SET_RUBOUT =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine sets up the string descriptor to point to a single
1 !       rubout character.
1 !
1 ! CALLING SEQUENCE:
1 !       SET_RUBOUT ()
1 !
1 ! INPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT INPUTS:
1 !       none
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       KEY_STRING: descriptor to action defining string
1 !
1 ! ROUTINE VALUE:
1 !       1
1 !
1 ! SIDE EFFECTS:
1 !       none
1 !
1 !--
1
2 BEGIN
2
2 KEY_STRING[DSC$W_LENGTH] = 1;
2 KEY_STRING[DSC$A_POINTER] = UPLIT BYTE (%CHAR (%X'7F'));
2
2 RETURN 1;
2
1 END;                                          ! End of routine SET_RUBOUT


                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                    7F  00010 P.AAB:    .ASCII  <127>                              ;


                                        .PSECT  $CODE$,NOWRT,2

                  0000 00000 SET_RUBOUT:
                                        .WORD   Save nothing                       ; 0806
0000'  CF                 01  B0 00002  MOVW    #1, KEY_STRING                     ; 0840
0000'  CF     0000'  CF   9E 00007      MOVAB   P.AAB, KEY_STRING+4                ; 0841
       50                 01  D0 0000E  MOVL    #1, R0                             ; 0843
                              04 00011  RET                                        ; 0845
```

; Routine Size: 18 bytes,   Routine Base: $CODE$ + 0243

```
;  398          0846  1 ROUTINE SET_DEFINITION =
;  399          0847  1
;  400          0848  1 !++
;  401          0849  1 !
;  402          0850  1 ! FUNCTIONAL DESCRIPTION:
;  403          0851  1 !
;  404          0852  1 !       This routine replaces a default definition with one from the
;  405          0853  1 !       action definition file.
;  406          0854  1 !
;  407          0855  1 ! CALLING SEQUENCE:
;  408          0856  1 !       SET_DEFINITION ()
;  409          0857  1 !
;  410          0858  1 ! INPUT PARAMETERS:
;  411          0859  1 !       none
;  412          0860  1 !
;  413          0861  1 ! IMPLICIT INPUTS:
;  414          0862  1 !       KEY_ACTION: ACL editor action code
;  415          0863  1 !       KEY_FLAGS: flags associated with the key definition
;  416          0864  1 !       KEY_STRING: descriptor of the string that defines a key
;  417          0865  1 !
;  418          0866  1 ! OUTPUT PARAMETERS:
;  419          0867  1 !       none
;  420          0868  1 !
;  421          0869  1 ! IMPLICIT OUTPUTS:
;  422          0870  1 !       none
;  423          0871  1 !
;  424          0872  1 ! ROUTINE VALUE:
;  425          0873  1 !       1
;  426          0874  1 !
;  427          0875  1 ! SIDE EFFECTS:
;  428          0876  1 !       The definition table is updated to reflect the new key definition.
;  429          0877  1 !
;  430          0878  1 !--
;  431          0879  1
;  432          0880  2 BEGIN
;  433          0881  2
;  434          0882  2 LITERAL
;  435          0883  2         CHAR_CSI        = %X'9B',               ! C1 CSI character
;  436          0884  2         CHAR_CSI_1      = %X'1B',               ! C0 CSI
;  437          0885  2         CHAR_CSI_2      = %X'5B',               !    equivalent
;  438          0886  2         CHAR_SS3        = %X'8F',               ! C1 SS3 character
;  439          0887  2         CHAR_SS3_1      = %X'1B',               ! C0 SS3
;  440          0888  2         CHAR_SS3_2      = %X'4F';               !    equivalent
;  441          0889  2
;  442          0890  2 LOCAL
;  443          0891  2         LOCAL_STATUS,                           ! Local error status
;  444          0892  2         NEW_KEY         : REF $BBLOCK,          ! Address of new definition storage
;  445          0893  2         NEXT_DEF        : REF $BBLOCK,          ! Address of next key definition
;  446          0894  2         KEY_INSERTED,                           ! Flag to indicate key inserted
;  447          0895  2         TERM_OFFSET;                            ! Size of overhead sequence
;  448          0896  2
;  449          0897  2 ! Check for angle bracket delimiters.  If present, there must be a matched pair.
;  450          0898  2
;  451          0899  2 IF .KEY_STRING[DSC$W_LENGTH] GTR 1
;  452          0900  2 THEN
;  453          0901  3     BEGIN
;  454          0902  3     IF .VECTOR[.KEY_STRING[DSC$A_POINTER], 0; ,BYTE] EQL '<'
```

```
;  455          0903  3      THEN
;  456          0904  4          BEGIN
;  457          0905  4          KEY_STRING[DSC$A_POINTER] = .KEY_STRING[DSC$A_POINTER] + 1;
;  458          0906  4          KEY_STRING[DSC$W_LENGTH] = .KEY_STRING[DSC$W_LENGTH] - 2;
;  459          0907  4          IF .VECTOR[.KEY_STRING[DSC$A_POINTER], .KEY_STRING[DSC$W_LENGTH]; ,BYTE] NEQ '>'
;  460          0908  4          THEN RETURN 0;
;  461          0909  3          END;
;  462          0910  2      END;
;  463          0911  2
;  464          0912  2  ! Check for conflicting type definitions.
;  465          0913  2
;  466          0914  3  IF (.KEY_BLOCK[KEY_V_CTRLCHAR] AND .KEY_BLOCK[KEY_V_ESCSEQ])
;  467          0915  3  OR (.KEY_BLOCK[KEY_V_CTRLCHAR] AND .KEY_STRING[DSC$W_LENGTH] NEQ 1)
;  468          0916  2  THEN RETURN 0;
;  469          0917  2
;  470          0918  2  ! If this is a C1 type definition, loop twice (once for the C1 definition
;  471          0919  2  ! and once for the C0 equivalent definition).  Otherwise, only go through
;  472          0920  2  ! once.
;  473          0921  2
;  474          0922  3  INCR J FROM 1 TO (IF .KEY_BLOCK[KEY_V_CSI] OR .KEY_BLOCK[KEY_V_SS3]
;  475          0923  3                  THEN 2 ELSE 1)
;  476          0924  2  DO
;  477          0925  3      BEGIN
;  478          0926  3
;  479          0927  3  ! Determine the size of the overhead area.
;  480          0928  3
;  481          0929  4      TERM_OFFSET = (IF .KEY_BLOCK[KEY_V_CSI] OR .KEY_BLOCK[KEY_V_SS3]
;  482          0930  4                     THEN .J
;  483          0931  4                     ELSE IF .KEY_BLOCK[KEY_V_ESCSEQ]
;  484          0932  4                          THEN 1
;  485          0933  3                          ELSE 0);
;  486          0934  3
;  487          0935  3  ! Allocate storage for the key definition block.
;  488          0936  3
;  489        P 0937  3      AED_L_WORSTERR = ALLOCATE (.KEY_STRING[DSC$W_LENGTH] + KEY_C_LENGTH +
;  490          0938  3                                                          + .TERM_OFFSET, NEW_KEY);
;  491          0939  3      IF NOT .AED_L_WORSTERR THEN RETURN 0;
;  492          0940  3
;  493          0941  3  ! Save the needed information in the key definition block.
;  494          0942  3
;  495          0943  3      NEW_KEY[KEY_B_ACTION] = .KEY_ACTION;
;  496          0944  3      NEW_KEY[KEY_B_SIZE] = .KEY_STRING[DSC$W_LENGTH] + .TERM_OFFSET;
;  497          0945  3      NEW_KEY[KEY_B_FLAGS] = .KEY_FLAGS OR KEY_M_USERDEF;
;  498          0946  3
;  499          0947  3  ! Set up the overhead area for the key text definition.
;  500          0948  3
;  501          0949  3      IF .KEY_BLOCK[KEY_V_CSI] OR .KEY_BLOCK[KEY_V_SS3]
;  502          0950  3      THEN
;  503          0951  4          BEGIN
;  504          0952  4          IF .J EQL 1
;  505          0953  5          THEN NEW_KEY[KEY_T_TEXT] = (IF .KEY_BLOCK[KEY_V_CSI]
;  506          0954  5                                      THEN CHAR_CSI ELSE CHAR_SS3)
;  507          0955  4          ELSE
;  508          0956  5              BEGIN
;  509          0957  6              NEW_KEY[KEY_T_TEXT] = (IF .KEY_BLOCK[KEY_V_CSI]
;  510          0958  5                                     THEN CHAR_CSI_1 ELSE CHAR_SS3_1);
;  511          0959  6              (NEW_KEY[KEY_T_TEXT]) + 1 = (IF .KEY_BLOCK[KEY_V_CSI]
```

```
5                                               THEN CHAR_CSI_2 ELSE CHAR_SS3_2);
4               END;
4           END
3       ELSE IF .KEY_BLOCK[KEY_V_ESCSEQ]
3           THEN NEW_KEY[KEY_T_TEXT] = %X'1B'
3           ELSE IF .KEY_BLOCK[KEY_V_CTRLCHAR]
3               THEN .KEY_STRING[DSC$A_POINTER] = ..KEY_STRING[DSC$A_POINTER] - %X'40';
3
3 ! Move over the key definition text.
3
3       CH$MOVE (.KEY_STRING[DSC$W_LENGTH], .KEY_STRING[DSC$A_POINTER],
3               NEW_KEY[KEY_T_TEXT] + .TERM_OFFSET);
3
3 ! Check for and remove any default definitions that this new definition
3 ! replaces.
3
3       NEXT_DEF = .KEY_TABLE[KEY_L_FLINK];
3       KEY_INSERTED = 0;
3       UNTIL .NEXT_DEF EQLA KEY_TABLE[KEY_L_FLINK]
3       DO
4           BEGIN
4           IF .NEXT_DEF[KEY_B_ACTION] EQL .KEY_ACTION
4           THEN
5               BEGIN
5               IF .KEY_INSERTED EQL 0
5               THEN
6                   BEGIN
6                   INSQUE (NEW_KEY[KEY_L_FLINK], NEXT_DEF[KEY_L_FLINK]);
6                   KEY_INSERTED = 1;
5                   END;
5               IF NOT .NEXT_DEF[KEY_V_USERDEF]
5               THEN
6                   BEGIN
6                   NEW_KEY = .NEXT_DEF[KEY_L_BLINK];
6                   REMQUE (NEXT_DEF[KEY_L_FLINK], KEY_INSERTED);
6                   NEXT_DEF = .NEW_KEY;
5                   END;
4               END;
4           NEXT_DEF = .NEXT_DEF[KEY_L_FLINK];
3           END;                                        ! End of C1 loop
2       END;
2 KEY_FLAGS = 0;
2 RETURN 1;
2
1 END;                                                  ! End of routine SET_DEFINITION
```

```
                              0FFC 00000 SET_DEFINITION:
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11     ; 0846
   5E               08   C2 00002               SUBL2   #8, SP                                    ;
   01       0000'   CF   B1 00005               CMPW    KEY_STRING, #1                            ; 0899
                    1F   1B 0000A               BLEQU   1$                                        ;
   3C       0000'   DF   91 0000C               CMPB    @KEY_STRING+4, #60                        ; 0902
                    18   12 00011               BNEQ    1$                                        ;
```

```
                                         0000'   CF   D6 00013        INCL    KEY_STRING+4                       ; 0905
                     0000'  CF                   02   A2 00017        SUBW2   #2, KEY_STRING                     ; 0906
                            50   0000'   CF   3C 0001C        MOVZWL  KEY_STRING, R0                     ; 0907
                            50   0000'   CF   C0 00021        ADDL2   KEY_STRING+4, R0                   ;
                            3E           60   91 00026        CMPB    (R0), #62                          ;
                                         0C   12 00029        BNEQ    2$                                 ;
              16     0000'  CF           03   E1 0002B  1$:   BBC     #3, KEY_BLOCK+10, 4$               ; 0914
              03     0000'  CF           04   E1 00031        BBC     #4, KEY_BLOCK+10, 3$               ;
                                       0142   31 00037  2$:   BRW     27$                                ;
              07     0000'  CF           03   E1 0003A  3$:   BBC     #3, KEY_BLOCK+10, 4$               ; 0915
                            01   0000'   CF   B1 00040        CMPW    KEY_STRING, #1                     ;
                                         F0   12 00045        BNEQ    2$                                 ;
                            06   0000'   CF   E8 00047  4$:   BLBS    KEY_BLOCK+10, 5$                   ; 0922
              05     0000'  CF           01   E1 0004C        BBC     #1, KEY_BLOCK+10, 6$               ;
                            5B           02   D0 00052  5$:   MOVL    #2, R11                            ;
                                         03   11 00055        BRB     7$                                 ;
                            5B           01   D0 00057  6$:   MOVL    #1, R11                            ;
                                         59   D4 0005A  7$:   CLRL    J                                  ;
                                       010F   31 0005C        BRW     26$                                ;
                            06   0000'   CF   E8 0005F  8$:   BLBS    KEY_BLOCK+10, 9$                   ; 0929
              05     0000'  CF           01   E1 00064        BBC     #1, KEY_BLOCK+10, 10$              ;
                            57           59   D0 00069  9$:   MOVL    J, TERM_OFFSET                     ; 0930
                                         0D   11 0006D        BRB     12$                                ;
              05     0000'  CF           04   E1 0006F  10$:  BBC     #4, KEY_BLOCK+10, 11$              ; 0931
                            57           01   D0 00075        MOVL    #1, TERM_OFFSET                    ;
                                         02   11 00078        BRB     12$                                ;
                                         57   D4 0007A  11$:  CLRL    TERM_OFFSET                        ;
                                   04    AE   9F 0007C  12$:  PUSHAB  NEW_KEY                            ; 0938
                            50   0000'   CF   3C 0007F        MOVZWL  KEY_STRING, R0                     ;
                        04  AE   0B A740      9E 00084        MOVAB   11(TERM_OFFSET)[R0], 4(SP)         ;
                                   04    AE   9F 0008A        PUSHAB  4(SP)                              ;
                 00000000G  00           02   FB 0008D        CALLS   #2, LIB$GET_VM                     ;
                            58           50   D0 00094        MOVL    R0, VM_STATUS                      ;
                            11           58   E9 00097        BLBC    VM_STATUS, 13$                     ;
                            50   0000'   CF   3C 0009A        MOVZWL  KEY_STRING, R0                     ;
                            50   0B A740      9E 0009F        MOVAB   11(TERM_OFFSET)[R0], R0            ;
50            00            6E           00   2C 000A4        MOVC5   #0, (SP), #0, R0, @NEW_KEY         ;
                                   04    BE      000A9                                                   ;
                     0000'  CF           58   D0 000AB  13$:  MOVL    VM_STATUS, AED_L_WORSTERR          ;
                            82   0000'   CF   E9 000B0        BLBC    AED_L_WORSTERR, 2$                 ; 0939
                            50     04    AE   D0 000B5        MOVL    NEW_KEY, R0                        ; 0943
                        08  A0   0000'   CF   90 000B9        MOVB    KEY_ACTION, 8(R0)                  ;
          09  A0     0000'  CF           57   81 000BF        ADDB3   TERM_OFFSET, KEY_STRING, 9(R0)     ; 0944
          0A  A0     0000'  CF           20   89 000C6        BISB3   #32, KEY_FLAGS, 10(R0)             ; 0945
52     0000'  CF            01           00   EF 000CD        EXTZV   #0, #1, KEY_BLOCK+10, R2           ; 0949
                            06           52   E8 000D4        BLBS    R2, 14$                            ;
              32     0000'  CF           01   E1 000D7        BBC     #1, KEY_BLOCK+10, 20$              ;
                            01           59   D1 000DD  14$:  CMPL    J, #1                              ; 0952
                                         13   12 000E0        BNEQ    17$                                ;
                            06           52   E9 000E2        BLBC    R2, 15$                            ; 0953
                            51     9B    8F   9A 000E5        MOVZBL  #155, R1                           ;
                                         04   11 000E9        BRB     16$                                ;
                            51     8F    8F   9A 000EB  15$:  MOVZBL  #143, R1                           ;
                        0B  A0           51   90 000EF  16$:  MOVB    R1, 11(R0)                         ;
                                         35   11 000F3        BRB     22$                                ;
                            51           1B   D0 000F5  17$:  MOVL    #27, R1                            ; 0957
                        0B  A0           51   90 000F8        MOVB    R1, 11(R0)                         ;
```

```
                         06          52  E9 000FC         BLBC    R2, 18$                              ; 0959
                         51     5B   8F  9A 000FF         MOVZBL  #91, R1                              ;
                                     04  11 00103         BRB     19$                                  ;
                         51     4F   8F  9A 00105 18$:    MOVZBL  #79, R1                              ;
                   0C    A0          51  D0 00109 19$:    MOVL    R1, 12(R0)                           ;
                                     1B  11 0010D         BRB     22$                                  ; 0949
        06    0000'      CF          04  E1 0010F 20$:    BBC     #4, KEY_BLOCK+10, 21$                ; 0963
                 0B      A0          1B  90 00115         MOVB    #27, 11(R0)                          ; 0964
                                     0F  11 00119         BRB     22$                                  ;
        09    0000'      CF          03  E1 0011B 21$:    BBC     #3, KEY_BLOCK+10, 22$                ; 0965
              0000'      DF 00000040 8F  C2 00121         SUBL2   #64, @KEY_STRING+4                   ; 0966
   0B A740    0000'      DF    0000' CF  28 0012A 22$:    MOVC3   KEY_STRING, @KEY_STRING+4, 11(TERM_OFFSET)- ; 0971
                                                                  [R0]                                 ;
                         56    0000G CF  D0 00134         MOVL    KEY_TABLE, NEXT_DEF                  ; 0976
                                     5A  D4 00139         CLRL    KEY_INSERTED                         ; 0977
                         50    0000G CF  9E 0013B 23$:    MOVAB   KEY_TABLE, R0                        ; 0978
                         50          56  D1 00140         CMPL    NEXT_DEF, R0                         ;
                                     29  13 00143         BEQL    26$                                  ;
              0000'      CF     08   A6  91 00145         CMPB    8(NEXT_DEF), KEY_ACTION              ; 0981
                                     1C  12 0014B         BNEQ    25$                                  ;
                                     5A  D5 0014D         TSTL    KEY_INSERTED                         ; 0984
                                     07  12 0014F         BNEQ    24$                                  ;
                         66     04   BE  0E 00151         INSQUE  @NEW_KEY, (NEXT_DEF)                 ; 0987
                         5A          01  D0 00155         MOVL    #1, KEY_INSERTED                     ; 0988
        0C       0A      A6          05  E0 00158 24$:    BBS     #5, 10(NEXT_DEF), 25$                ; 0990
                 04      AE     04   A6  D0 0015D         MOVL    4(NEXT_DEF), NEW_KEY                 ; 0993
                         5A          66  0F 00162         REMQUE  (NEXT_DEF), KEY_INSERTED             ; 0994
                         56     04   AE  D0 00165         MOVL    NEW_KEY, NEXT_DEF                    ; 0995
                         56          66  D0 00169 25$:    MOVL    (NEXT_DEF), NEXT_DEF                 ; 0998
                                     CD  11 0016C         BRB     23$                                  ; 0978
  FEEB           59      01          5B  F1 0016E 26$:    ACBL    R11, #1, J, 8$                       ; 0922
                               0000' CF  94 00174         CLRB    KEY_FLAGS                            ; 1001
                         50          01  D0 00178         MOVL    #1, R0                               ; 1002
                                         04 0017B         RET                                          ;
                                     50  D4 0017C 27$:    CLRL    R0                                   ; 1004
                                         04 0017E         RET                                          ;

; Routine Size:  383 bytes,    Routine Base:  $CODE$ + 0255
```

```
;  558          1005  1 GLOBAL ROUTINE AED_DECODEKEY =
;  559          1006  1
;  560          1007  1 !++
;  561          1008  1 !
;  562          1009  1 ! FUNCTIONAL DESCRIPTION:
;  563          1010  1 !
;  564          1011  1 !       This routine accepts input from the input channel and decodes it
;  565          1012  1 !       according to the definitions from the action definition file (or
;  566          1013  1 !       the default definitions).
;  567          1014  1 !
;  568          1015  1 ! CALLING SEQUENCE:
;  569          1016  1 !       AED_DECODEKEY ()
;  570          1017  1 !
;  571          1018  1 ! INPUT PARAMETERS:
;  572          1019  1 !       none
;  573          1020  1 !
;  574          1021  1 ! IMPLICIT INPUTS:
;  575          1022  1 !       none
;  576          1023  1 !
;  577          1024  1 ! OUTPUT PARAMETERS:
;  578          1025  1 !       none
;  579          1026  1 !
;  580          1027  1 ! IMPLICIT OUTPUTS:
;  581          1028  1 !       none
;  582          1029  1 !
;  583          1030  1 ! ROUTINE VALUE:
;  584          1031  1 !       0 if a fatal error occurs,
;  585          1032  1 !       Action code value if special (AED_V_ACTION also set)
;  586          1033  1 !       ASCII character value
;  587          1034  1 !
;  588          1035  1 ! SIDE EFFECTS:
;  589          1036  1 !       none
;  590          1037  1 !
;  591          1038  1 !--
;  592          1039  1
;  593          1040  2 BEGIN
;  594          1041  2
;  595          1042  2 MACRO
;  596          1043  2         TERM_CHAR       = AED_W_IOSB[2] %,
;  597          1044  2         TERM_SIZE       = AED_W_IOSB[3] %,
;  598          1045  2         TERM_STRING     = INPUT_BUFFER[.AED_W_IOSB[1]] %;
;  599          1046  2
;  600          1047  2 LABEL
;  601          1048  2         DECODE_KEY;
;  602          1049  2
;  603          1050  2 LOCAL
;  604          1051  2         LOCAL_STATUS,                          ! Local routine return status
;  605          1052  2         INPUT_BUFFER    : VECTOR [10,BYTE],    ! Storage for input characters
;  606          1053  2         TERM_DESC       : $BBLOCK [DSC$C_S_BLN],       ! Term table descr
;  607          1054  2         TERM_TABLE      : VECTOR [8]           ! Terminator table
;  608          1055  2                           INITIAL (REP 8 OF (-1)),     ! All are terminators
;  609          1056  2         NEXT_DEF        : REF $BBLOCK,         ! Address of next key definition
;  610          1057  2         KEY_WITHOUT_GLD : REF $BBLOCK,         ! Address of key definition without gold required
;  611          1058  2         RETURN_CHAR;                           ! Character/code to return
;  612          1059  2
;  613          1060  2 ! If this is a recovery, get a word (16 bits) from the recovery record.  This
;  614          1061  2 ! contains a character if the high byte is zero, or an editor action if not.
```

```
;  615          1062  2
;  616          1063  2 IF .AED_B_OPTIONS[AED_V_RECOVER]
;  617          1064  2 THEN
;  618          1065  3     BEGIN
;  619          1066  3     IF .RECOVER_RAB[RAB$W_RSZ] LEQ 0
;  620          1067  3     THEN
;  621          1068  4         BEGIN
;  622          1069  5         IF NOT (LOCAL_STATUS = $GET (RAB = RECOVER_RAB))
;  623          1070  4         THEN
;  624          1071  5             BEGIN
;  625          1072  5             IF .LOCAL_STATUS NEQ RMS$_EOF
;  626          1073  5             THEN
;  627          1074  6                 BEGIN
;  628          1075  6                 AED_FILERROR (AED$_RECREADERR, RECOVER_FAB,
;  629          1076  6                               .RECOVER_RAB[RAB$L_STS], .RECOVER_RAB[RAB$L_STV]);
;  630          1077  6                 AED_B_OPTIONS[AED_V_RECOVER] = 0;
;  631          1078  5                 END;
;  632          1079  5             $CLOSE (FAB = RECOVER_FAB);
;  633          1080  5             AED_B_OPTIONS[AED_V_RECOVER] = 0;
;  634          1081  5             RETURN 1;
;  635          1082  4             END;
;  636          1083  4         RECOVER_INDEX = 0;
;  637          1084  3         END;
;  638          1085  3     RETURN_CHAR = .RECOVER_BUFFER[.RECOVER_INDEX];
;  639          1086  3     RECOVER_INDEX = .RECOVER_INDEX + 1;
;  640          1087  3     AED_L_FLAGS[AED_V_ACTIONKEY] = .RECOVER_BUFFER[.RECOVER_INDEX];
;  641          1088  3     RECOVER_INDEX = .RECOVER_INDEX + 1;
;  642          1089  3     RECOVER_RAB[RAB$W_RSZ] = .RECOVER_RAB[RAB$W_RSZ] - 2;
;  643          1090  3     END
;  644          1091  2 ELSE
;  645          1092  2
;  646          1093  2 ! Get a character typed (or escape sequence) by the user.
;  647          1094  2
;  648          1095  3 DECODE_KEY: BEGIN
;  649          1096  3     TERM_DESC[DSC$W_LENGTH] = 8*4;
;  650          1097  3     TERM_DESC[DSC$A_POINTER] = TERM_TABLE;
;  651        P 1098  3     AED_L_STATUS = $QIOW (CHAN = .AED_W_TERMIN, ! Get character
;  652        P 1099  3                           FUNC = IO$_READVBLK OR IO$M_ESCAPE
;  653        P 1100  3                                                OR IO$M_NOFILTR
;  654        P 1101  3                                                OR IO$M_TRMNOECHO,
;  655        P 1102  3                           IOSB = AED_W_IOSB,
;  656        P 1103  3                           P1 = INPUT_BUFFER,
;  657        P 1104  3                           P2 = 10,
;  658          1105  3                           P4 = TERM_DESC);
;  659          1106  3     IF .AED_L_STATUS THEN AED_L_STATUS = .AED_W_IOSB[0];
;  660          1107  3     IF NOT .AED_L_STATUS
;  661          1108  3     THEN
;  662          1109  4         BEGIN
;  663          1110  4         IF .AED_L_STATUS EQL SS$_BADESCAPE
;  664          1111  4         THEN
;  665          1112  5             BEGIN
;   66          1113  5             AED_L_STATUS = 1;
;  667          1114  5             RETURN_CHAR = AED_C_CHAR_ESC;
;  668          1115  5             LEAVE DECODE_KEY;
;  669          1116  4             END;
;  670          1117  4         SIGNAL (.AED_L_STATUS);
;  671          1118  4         RETURN 0;
```

```
;  672          1119  3          END;
;  673          1120  3
;  674          1121  3 ! If the character is nothing special, simply return with the character.
;  675          1122  3
;  676          1123  3      AED_L_FLAGS[AED_V_ACTIONKEY] = 0;
;  677          1124  3      IF .TERM_CHAR GEQ ' ' AND .TERM_CHAR NEQ %X'7F'
;  678          1125  3      THEN
;  679          1126  4          BEGIN
;  680          1127  4          RETURN_CHAR = .TERM_CHAR;
;  681          1128  4          LEAVE DECODE_KEY;
;  682          1129  3          END;
;  683          1130  3
;  684          1131  3 ! Otherwise, it will be necessary to search the action definition table to
;  685          1132  3 ! determine whether or not the character (or characters) defines an ACL
;  686          1133  3 ! editor action.
;  687          1134  3
;  688          1135  3      KEY_WITHOUT_GLD = 0;
;  689          1136  3      NEXT_DEF = .KEY_TABLE[KEY_L_FLINK];
;  690          1137  3      UNTIL .NEXT_DEF EQLA KEY_TABLE[KEY_L_FLINK]
;  691          1138  3      DO
;  692          1139  4          BEGIN
;  693          1140  4          IF CH$EQL (.NEXT_DEF[KEY_B_SIZE], NEXT_DEF[KEY_T_TEXT],
;  694          1141  4                     .TERM_SIZE, TERM_STRING, 0)
;  695          1142  4          THEN
;  696          1143  5              BEGIN
;  697          1144  5              IF .NEXT_DEF[KEY_V_GOLDREQ] EQL .AED_L_FLAGS[AED_V_GOLDKEY]
;  698          1145  5              THEN
;  699          1146  6                  BEGIN
;  700          1147  6                  AED_L_FLAGS[AED_V_ACTIONKEY] = 1;
;  701          1148  6                  RETURN_CHAR = .NEXT_DEF[KEY_B_ACTION];
;  702          1149  6                  LEAVE DECODE_KEY;
;  703          1150  5                  END;
;  704          1151  5              IF NOT .NEXT_DEF[KEY_V_GOLDREQ] THEN KEY_WITHOUT_GLD = .NEXT_DEF;
;  705          1152  4              END;
;  706          1153  4          NEXT_DEF = .NEXT_DEF[KEY_L_FLINK];
;  707          1154  3          END;
;  708          1155  3
;  709          1156  3 ! Nothing has been found in the definition table.  Check to see if there
;  710          1157  3 ! was a key defined except that the gold key was hit but not required.
;  711          1158  3 ! If this is the case, clear the GOLDKEY flag and return the appropriate
;  712          1159  3 ! action code.  Otherwise simply return the terminating character.
;  713          1160  3
;  714          1161  3      IF .KEY_WITHOUT_GLD NEQ 0
;  715          1162  3      THEN
;  716          1163  4          BEGIN
;  717          1164  4          AED_L_FLAGS[AED_V_GOLDKEY] = 0;
;  718          1165  4          AED_L_FLAGS[AED_V_ACTIONKEY] = 1;
;  719          1166  4          RETURN_CHAR = .KEY_WITHOUT_GLD[KEY_B_ACTION];
;  720          1167  4          LEAVE DECODE_KEY;
;  721          1168  3          END;
;  722          1169  3      RETURN_CHAR = .TERM_CHAR;
;  723          1170  2      END;                                        ! End of DECODE_KEY block
;  724          1171  2
;  725          1172  2 ! If the action cannot be logged (EXIT or QUIT), simply return now.
;  726          1173  2
;  727          1174  2 IF .AED_L_FLAGS[AED_V_ACTIONKEY]
;  728          1175  3 AND (.RETURN_CHAR EQL KEY_C_EXIT OR .RETURN_CHAR EQL KEY_C_QUIT)
```

```
;   729          1176  2 THEN RETURN .RETURN_CHAR;
;   730          1177  2
;   731          1178  2 ! If necessary, put the character or code into the journal buffer.  If
;   732          1179  2 ! the buffer fills up, write it out.
;   733          1180  2
;   734          1181  2 IF .AED_B_OPTIONS[AED_V_JOURNAL]
;   735          1182  2 THEN
;   736          1183  3     BEGIN
;   737          1184  3     IF .JOURNAL_INDEX GEQ 10
;   738          1185  3     THEN
;   739          1186  4         BEGIN
;   740          1187  4         IF NOT $PUT (RAB = JOURNAL_RAB) THEN AED_B_OPTIONS[AED_V_JOURNAL] = 0;
;   741          1188  4         CH$FILL (0, 10, JOURNAL_BUFFER);
;   742          1189  4         JOURNAL_INDEX = 0;
;   743          1190  3         END;
;   744          1191  3     JOURNAL_BUFFER[.JOURNAL_INDEX] = .RETURN_CHAR;
;   745          1192  3     JOURNAL_INDEX = .JOURNAL_INDEX + 1;
;   746          1193  3     IF .AED_L_FLAGS[AED_V_ACTIONKEY]
;   747          1194  3     THEN JOURNAL_BUFFER[.JOURNAL_INDEX] = 1
;   748          1195  3     ELSE JOURNAL_BUFFER[.JOURNAL_INDEX] = 0;
;   749          1196  3     JOURNAL_INDEX = .JOURNAL_INDEX + 1;
;   750          1197  2     END;
;   751          1198  2
;   752          1199  2 RETURN .RETURN_CHAR;
;   753          1200  2
;   754          1201  1 END;                                    ! End of routine AED_DECODEKEY


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    00011       .BLKB   3
                          FFFFFFFF# 00014 P.AAC: .LONG   -1[8]                                   ;

                                                .EXTRN  SYS$CLOSE, SYS$QIOW
                                                .EXTRN  SYS$PUT

                                                .PSECT  $CODE$,NOWRT,2

                                    0FFC 00000  .ENTRY  AED_DECODEKEY, Save R2,R3,R4,R5,R6,R7,R8,-  ; 1005
                                                        R9,R10,R11
                          5B 00000000G  00  9E 00002    MOVAB   SCR$SET_CURSOR, R11                ;
                          5A      0000' CF  9E 00009    MOVAB   AED_L_FLAGS, R10                   ;
                          5E            34  C2 0000E    SUBL2   #52, SP                            ;
6E          0000'     CF              20  28 00011    MOVC3   #32, P.AAC, TERM_TABLE             ; 1055
76            04      AA              01  E1 00017    BBC     #1, AED_B_OPTIONS, 4$              ; 1063
                              04F6    CA  B5 0001C    TSTW    RECOVER_RAB+34                     ; 1066
                                      46  12 00020    BNEQ    3$                                 ;
                              04D4    CA  9F 00022    PUSHAB  RECOVER_RAB                        ; 1069
    00000000G     00                  01  FB 00026    CALLS   #1, SYS$GET                        ;
                  34                  50  E8 0002D    BLBS    LOCAL_STATUS, 2$                   ;
    0001827A      8F                  50  D1 00030    CMPL    LOCAL_STATUS, #98938               ; 1072
                                      18  13 00037    BEQL    1$                                 ;
                  7E          04DC    CA  7D 00039    MOVQ    RECOVER_RAB+8, -(SP)               ; 1076
                              0424    CA  9F 0003E    PUSHAB  RECOVER_FAB                        ; 1075
                        00000000G     8F  DD 00042    PUSHL   #AED$_RECREADERR                   ;
          0000G     CF                  04  FB 00048    CALLS   #4, AED_FILERROR                   ;
```

```
                04  AA              02  8A 0004D      BICB2   #2, AED_B_OPTIONS                  ; 1077
                          0424      CA  9F 00051 1$:  PUSHAB  RECOVER_FAB                        ; 1079
         00000000G  00              01  FB 00055      CALLS   #1, SYS$CLOSE
                04  AA              02  8A 0005C      BICB2   #2, AED_B_OPTIONS                  ; 1080
                    50              01  D0 00060      MOVL    #1, R0                             ; 1081
                                    04  00063         RET
                          0524      CA  D4 00064 2$:  CLRL    RECOVER_INDEX                      ; 1083
                    50    0518      CA  9E 00068 3$:  MOVAB   RECOVER_BUFFER, R0                 ; 1085
                    57    0524    DA40  9A 0006D      MOVZBL  @RECOVER_INDEX[R0], RETURN_CHAR
                          0524      CA  D6 00073      INCL    RECOVER_INDEX                      ; 1086
                    50    0518      CA  9E 00077      MOVAB   RECOVER_BUFFER, R0                 ; 1087
                          0524    DA40  9F 0007C      PUSHAB  @RECOVER_INDEX[R0]
 02  AA     01      05              9E  F0 00081      INSV    @(SP)+, #5, #1, AED_L_FLAGS+2
                          0524      CA  D6 00087      INCL    RECOVER_INDEX                      ; 1088
              04F6  CA              02  A2 0008B      SUBW2   #2, RECOVER_RAB+34                 ; 1089
                                    50  11 00090      BRB     6$                                 ; 1063
                20  AE              20  B0 00092 4$:  MOVW    #32, TERM_DESC                     ; 1096
                24  AE              6E  9E 00096      MOVAB   TERM_TABLE, TERM_DESC+4            ; 1097
                                    7E  7C 0009A      CLRQ    -(SP)                              ; 1105
                            28      AE  9F 0009C      PUSHAB  TERM_DESC
                    7E              0A  7D 0009F      MOVQ    #10, -(SP)
                            3C      AE  9F 000A2      PUSHAB  INPUT_BUFFER
                                    7E  7C 000A5      CLRQ    -(SP)
                          0084      CA  9F 000A7      PUSHAB  AED_W_IOSB
                    7E    5231      8F  3C 000AB      MOVZWL  #21041, -(SP)
                    7E      7C      AA  3C 000B0      MOVZWL  AED_W_TERMIN, -(SP)
                                    7E  D4 000B4      CLRL    -(SP)
         00000000G  00              0C  FB 000B6      CALLS   #12, SYS$QIOW
              008C  CA              50  D0 000BD      MOVL    R0, AED_L_STATUS
                    0C    008C      CA  E9 000C2      BLBC    AED_L_STATUS, 5$                   ; 1106
              008C  CA    0084      CA  3C 000C7      MOVZWL  AED_W_IOSB, AED_L_STATUS
                    60    008C      CA  E8 000CE      BLBS    AED_L_STATUS, 11$                  ; 1107
                    3C    008C      CA  D1 000D3 5$:  CMPL    AED_L_STATUS, #60                  ; 1110
                                    0B  12 000D8      BNEQ    7$
              008C  CA              01  D0 000DA      MOVL    #1, AED_L_STATUS                   ; 1113
                    57              1B  D0 000DF      MOVL    #27, RETURN_CHAR                   ; 1114
                                  00C7  31 000E2 6$:  BRW     18$                                ; 1115
            12      6A              03  E1 000E5 7$:  BBC     #3, AED_L_FLAGS, 8$                ; 1117
                                    01  DD 000E9      PUSHL   #1
                                    15  DD 000EB      PUSHL   #21
         00000000G  00              02  FB 000ED      CALLS   #2, SCR$ERASE_PAGE
                                    01  DD 000F4      PUSHL   #1
                                    15  DD 000F6      PUSHL   #21
                    6B              02  FB 000F8      CALLS   #2, SCR$SET_CURSOR
                          008C      CA  DD 000FB 8$:  PUSHL   AED_L_STATUS
         00000000G  00              01  FB 000FF      CALLS   #1, LIB$SIGNAL
            0B      6A              03  E1 00106      BBC     #3, AED_L_FLAGS, 9$
                    7E      20      AA  9A 0010A      MOVZBL  AED_B_COLUMN, -(SP)
                    7E      24      AA  9A 0010E      MOVZBL  AED_B_LINE, -(SP)
                    6B              02  FB 00112      CALLS   #2, SCR$SET_CURSOR
                    50    008C      CA  D0 00115 9$:  MOVL    AED_L_STATUS, R0
                    07              50  93 0011A      BITB    R0, #7
                                    11  13 0011D      BEQL    10$
 51         50      03              00  EF 0011F      EXTZV   #0, #3, R0, R1
 51    14   AA      03              00  ED 00124      CMPZV   #0, #3, AED_L_WORSTERR, R1
                                    04  18 0012A      BGEQ    10$
                14  AA              50  D0 0012C      MOVL    R0, AED_L_WORSTERR
```

```
                                   00DE   31 00130 10$:   BRW     25$                                              ; 1118
                        02    AA          20   8A 00133 11$:   BICB2   #32, AED_L_FLAGS+2                          ; 1123
                              58   0088   CA   3C 00137        MOVZWL  AED_W_IOSB+4, R8                            ; 1124
                              20          58   B1 0013C        CMPW    R8, #32                                     ;
                                          07   1F 0013F        BLSSU   12$                                         ;
                      007F    8F          58   B1 00141        CMPW    R8, #127                                    ;
                                          61   12 00146        BNEQ    17$                                         ;
                                          56   D4 00148 12$:   CLRL    KEY_WITHOUT_GLD                             ; 1135
                              54   0000G  CF   D0 0014A        MOVL    KEY_TABLE, NEXT_DEF                         ; 1136
                              55   0086   CA   3C 0014F        MOVZWL  AED_W_IOSB+2, R5                            ; 1141
                              59     28   AE   9E 00154        MOVAB   INPUT_BUFFER, R9                            ;
                              50   0000G  CF   9E 00158 13$:   MOVAB   KEY_TABLE, R0                               ; 1137
                              50          54   D1 0015D        CMPL    NEXT_DEF, R0                                ;
                                          35   13 00160        BEQL    16$                                         ;
                              50     09   A4   9A 00162        MOVZBL  9(NEXT_DEF), R0                             ; 1140
 008A   CA        00    0B    A4          50   2D 00166        CMPC5   R0, 11(NEXT_DEF), #0, AED_W_IOSB+6, (R9)-   ; 1141
                                        6945      0016E                [R5]                                        ;
                                          20   12 00170        BNEQ    15$                                         ;
        50   01   AA          01          03   EF 00172        EXTZV   #3, #1, AED_L_FLAGS+1, R0                   ; 1144
        50   0A   A4          01          02   ED 00178        CMPZV   #2, #1, 10(NEXT_DEF), R0                    ;
                                          0A   12 0017E        BNEQ    14$                                         ;
                        02    AA          20   88 00180        BISB2   #32, AED_L_FLAGS+2                          ; 1147
                              57     08   A4   9A 00184        MOVZBL  8(NEXT_DEF), RETURN_CHAR                    ; 1148
                                          22   11 00188        BRB     18$                                         ; 1149
                  03    0A    A4          02   E0 0018A 14$:   BBS     #2, 10(NEXT_DEF), 15$                       ; 1151
                              56          54   D0 0018F        MOVL    NEXT_DEF, KEY_WITHOUT_GLD                   ;
                              54          64   D0 00192 15$:   MOVL    (NEXT_DEF), NEXT_DEF                        ; 1153
                                          C1   11 00195        BRB     13$                                         ; 1137
                                          56   D5 00197 16$:   TSTL    KEY_WITHOUT_GLD                             ; 1161
                                          0E   13 00199        BEQL    17$                                         ;
                        01    AA          08   8A 0019B        BICB2   #8, AED_L_FLAGS+1                           ; 1164
                        02    AA          20   88 0019F        BISB2   #32, AED_L_FLAGS+2                          ; 1165
                              57     08   A6   9A 001A3        MOVZBL  8(KEY_WITHOUT_GLD), RETURN_CHAR             ; 1166
                                          03   11 001A7        BRB     18$                                         ; 1167
                              57          58   D0 001A9 17$:   MOVL    R8, RETURN_CHAR                             ; 1169
                  0A    02    AA          05   E1 001AC 18$:   BBC     #5, AED_L_FLAGS+2, 19$                      ; 1174
                              27          57   D1 001B1        CMPL    RETURN_CHAR, #39                            ; 1175
                                          57   13 001B4        BEQL    24$                                         ;
                              28          57   D1 001B6        CMPL    RETURN_CHAR, #40                            ;
                                          52   13 001B9        BEQL    24$                                         ;
                              4E     04   AA   E9 001BB 19$:   BLBC    AED_B_OPTIONS, 24$                          ; 1181
                              0A   0420   CA   D1 001BF        CMPL    JOURNAL_INDEX, #10                          ; 1184
                                          1E   19 001C4        BLSS    21$                                         ;
                                   0378   CA   9F 001C6        PUSHAB  JOURNAL_RAB                                 ; 1187
                 00000000G    00          01   FB 001CA        CALLS   #1, SYS$PUT                                 ;
                              04          50   E8 001D1        BLBS    R0, 20$                                     ;
                        04    AA          01   8A 001D4        BICB2   #1, AED_B_OPTIONS                           ;
        0A        00          6E          00   2C 001D8 20$:   MOVC5   #0, (SP), #0, #10, JOURNAL_BUFFER           ; 1188
                                   0414   CA      001DD                                                        ;
                                   0420   CA   D4 001E0        CLRL    JOURNAL_INDEX                               ; 1189
                              50   0414   CA   9E 001E4 21$:   MOVAB   JOURNAL_BUFFER, R0                          ; 1191
                      0420  DA40          57   90 001E9        MOVB    RETURN_CHAR, @JOURNAL_INDEX[R0]             ;
                                   0420   CA   D6 001EF        INCL    JOURNAL_INDEX                               ; 1192
                              50   0414   CA   9E 001F3        MOVAB   JOURNAL_BUFFER, R0                          ; 1194
                              50   0420   CA   C0 001F8        ADDL2   JOURNAL_INDEX, R0                           ;
                  05    02    AA          05   E1 001FD        BBC     #5, AED_L_FLAGS+2, 22$                      ; 1193
                              60          01   90 00202        MOVB    #1, (R0)                                    ; 1194
```

```
                    02  11 00205         BRB     23$                   ;
                    60  94 00207 22$:    CLRB    (R0)                  ; 1195
              0420  CA  D6 00209 23$:    INCL    JOURNAL_INDEX         ; 1196
        50          57  D0 0020D 24$:    MOVL    RETURN_CHAR, R0       ; 1199
                        04 00210         RET                           ;
                    50  D4 00211 25$:    CLRL    R0                    ; 1201
                        04 00213         RET                           ;

; Routine Size:  532 bytes,    Routine Base:  $CODE$ + 03D4
```

```
1 GLOBAL ROUTINE AED_FLUSHKEY =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine flushes the journal buffer and closes the journal file.
1 !
1 ! CALLING SEQUENCE:
1 !       AED_FLUSHKEY ()
1 !
1 ! INPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT INPUTS:
1 !       OWN storage
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       none
1 !
1 ! ROUTINE VALUE:
1 !       1
1 !
1 ! SIDE EFFECTS:
1 !       none
1 !
1 !--
1
2 BEGIN
2
2 ! If not writing a journal file, simply return now.
2
2 IF NOT .AED_B_OPTIONS[AED_V_JOURNAL] THEN RETURN 1;
2
2 IF .JOURNAL_INDEX GTR 0
2 THEN
3     BEGIN
3     JOURNAL_RAB[RAB$W_RSZ] = .JOURNAL_INDEX * 2;
3     $PUT (RAB = JOURNAL_RAB);
2     END;
2
2 JOURNAL_FAB[FAB$V_DLT] = NOT .AED_B_OPTIONS[AED_V_KEEPJNL];
2 $CLOSE (FAB = JOURNAL_FAB);
2
2 RETURN 1;
2
1 END;                                          ! End of routine AED_FLUSHKEY


                    0000 00000        .ENTRY  AED_FLUSHKEY, Save nothing           ; 1202
 34     0000'  CF   E9 00002          BLBC    AED_B_OPTIONS, 2$                    ; 1237
```

```
                                50      0000'  CF  D0 00007          MOVL    JOURNAL_INDEX, R0                       ; 1239
                                                   11 15 0000C          BLEQ    1$                                      ;
                    0000'  CF                   50      02 A5 0000E          MULW3   #2, R0, JOURNAL_RAB+34          ; 1242
                                                   0000'  CF 9F 00014          PUSHAB  JOURNAL_RAB                     ; 1243
                           00000000G            00      01 FB 00018          CALLS   #1, SYS$PUT                     ;
            50      0000'  CF                   01      03 EF 0001F 1$:      EXTZV   #3, #1, AED_B_OPTIONS, R0       ; 1246
                                                   50      50 D2 00026          MCOML   R0, R0                          ;
    0000'   CF              01                  07      50 F0 00029          INSV    R0, #7, #1, JOURNAL_FAB+5       ;
                                                   0000'  CF 9F 00030          PUSHAB  JOURNAL_FAB                     ; 1247
                           00000000G            00      01 FB 00034          CALLS   #1, SYS$CLOSE                   ;
                                                   50      01 D0 0003B 2$:      MOVL    #1, R0                          ; 1249
                                                           04 0003E             RET                                     ; 1251

; Routine Size:  63 bytes,    Routine Base:  $CODE$ + 05E8

;  806          1252  1
;  807          1253  1 END
;  808          1254  0 ELUDOM
```

```
;                                   PSECT SUMMARY
;
;       Name                            Bytes                           Attributes
;
;  AED_COMMON                          1320  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(0)
;  $OWN$                                 20  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  _LIB$KEY0$                            94  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;  _LIB$STATE$                          538  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;  _LIB$KEY1$                           518  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;  $PLIT$                                52  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                              1575  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                               Library Statistics
;
;                                       -------- Symbols --------       Pages       Processing
;       File                            Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1         18619       122         0        1000       00:01.8
;  _$255$DUA28:[SYSLIB]TPAMAC.L32;1         42        29        69          14       00:00.2


;                                   COMMAND QUALIFIERS

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

```
; Size:          1575 code + 2542 data bytes
; Run Time:        01:10.7
; Elapsed Time:    03:37.1
; Lines/CPU Min:    1064
; Lexemes/CPU-Min: 71863
; Memory Used:  431 pages
; Compilation Complete
```

AEDCLENUP
LIS

SYMBOLS
LIS

UTILITY
LIS

ACLEDTDEF
REQ

ACLEDT

SETSHOACL
MAP

ACLEDT
MAP

AEDLCLDEE
SDL

SUMMARY
LIS

SYMBOLMSG
LIS

AEDDECODE
LIS

AEDHELP
LIS

AEDINIT
LIS

AEDMAIN
LIS

AEDKEYTAB
LIS