```
   SSSSSSSSSSSS  UUU        UUU  MMM         MMM
   SSSSSSSSSSSS  UUU        UUU  MMM         MMM
   SSSSSSSSSSSS  UUU        UUU  MMM         MMM
SSS              UUU        UUU  MMMMMM   MMMMMM
SSS              UUU        UUU  MMMMMM   MMMMMM
SSS              UUU        UUU  MMMMMM   MMMMMM
SSS              UUU        UUU  MMM   MMM   MMM
SSS              UUU        UUU  MMM   MMM   MMM
SSS              UUU        UUU  MMM   MMM   MMM
   SSSSSSSSS     UUU        UUU  MMM         MMM
   SSSSSSSSS     UUU        UUU  MMM         MMM
   SSSSSSSSS     UUU        UUU  MMM         MMM
            SSS  UUU        UUU  MMM         MMM
            SSS  UUU        UUU  MMM         MMM
            SSS  UUU        UUU  MMM         MMM
            SSS  UUU        UUU  MMM         MMM
            SSS  UUU        UUU  MMM         MMM
            SSS  UUU        UUU  MMM         MMM
SSSSSSSSSSSS     UUUUUUUUUUUUUUU  MMM         MMM
SSSSSSSSSSSS     UUUUUUUUUUUUUUU  MMM         MMM
SSSSSSSSSSSS     UUUUUUUUUUUUUUU  MMM         MMM
```

FILEID**SUMED

```
  SSSSSSSS  UU      UU  MM      MM  EEEEEEEEEE  DDDDDDDD
  SSSSSSSS  UU      UU  MM      MM  EEEEEEEEEE  DDDDDDDD
SS          UU      UU  MMMM  MMMM  EE          DD      DD
SS          UU      UU  MMMM  MMMM  EE          DD      DD
SS          UU      UU  MM  MM  MM  EE          DD      DD
SS          UU      UU  MM  MM  MM  EE          DD      DD
  SSSSSS    UU      UU  MM      MM  EEEEEEEE    DD      DD
  SSSSSS    UU      UU  MM      MM  EEEEEEEE    DD      DD
        SS  UU      UU  MM      MM  EE          DD      DD
        SS  UU      UU  MM      MM  EE          DD      DD
        SS  UU      UU  MM      MM  EE          DD      DD      ....
        SS  UU      UU  MM      MM  EE          DD      DD      ....
SSSSSSSS    UUUUUUUUUU  MM      MM  EEEEEEEEEE  DDDDDDDD        ....
SSSSSSSS    UUUUUUUUUU  MM      MM  EEEEEEEEEE  DDDDDDDD        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

SUM
V04

Table of contents

```
          0000   ;
          0000   ; Version:      'V04-000'
          0000   ;
          0000   ;****************************************************************************
          0000   ;*                                                                          *
          0000   ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
          0000   ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
          0000   ;*  ALL RIGHTS RESERVED.                                                    *
          0000   ;*                                                                          *
          0000   ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
          0000   ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
          0000   ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
          0000   ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
          0000   ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
          0000   ;*  TRANSFERRED.                                                            *
          0000   ;*                                                                          *
          0000   ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
          0000   ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
          0000   ;*  CORPORATION.                                                            *
          0000   ;*                                                                          *
          0000   ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
          0000   ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
          0000   ;*                                                                          *
          0000   ;*                                                                          *
          0000   ;****************************************************************************
          0000   ;
          0000   ;
          0000   ; Assembly parameters
          0000   ;
00000200  0000          BUF_SIZE = 512                      ; Size in bytes of slipr input buffers
00000084  0000          CMD_SIZE = 132                      ; Size of input command line
          0000   ;
          0000          $NAMDEF
          0000          $RABDEF
          0000          $FABDEF
          0000          $CLIDEF
          0000   ;
          0000   ; Edit node offsets
          0000   ;
00000000  0000          ED$L_FWD    = 0                     ; Forward pointer
00000004  0000          ED$L_BWD    = 4                     ; Backword pointer
00000008  0000          ED$W_LOC1   = 8                     ; Locator 1
0000000A  0000          ED$W_LOC2   = 10                    ; Locator 2
0000000C  0000          ED$W_LINES  = 12                    ; Insert lines
0000000E  0000          ED$W_RFA    = 14                    ; Record file address (3 words)
00000014  0000          ED$L_FILE   = 20                    ; File node pointer
00000018  0000          ED$B_FLAGS  = 24                    ; Flags
00000019  0000          ED$B_FILENO= 25                     ; File number
          0000   ;
0000001A  0000          ED$K_BLN    = 26
          0000   ;
          0000   ;
          0000   ; File node offsets
          0000   ;
00000000  0000          SLP$L_FWD   = 0                     ; Forward pointer
00000004  0000          SLP$L_BWD   = 4                     ; Backward pointer
00000008  0000          SLP$W_LOC1  = 8                     ; Locator-1
```

```
0000000A  0000          SLP$W_LOC2 = 10                  ; Locator-2
0000000C  0000          SLP$B_FLAGS= 12                  ; Flags
0000000D  0000          SLP$B_FILENO = 13                ; File priority
0000000E  0000          SLP$W_DOT  = 14                  ; Dot value
00000010  0000          SLP$Q_AUDDS= 16                  ; Audit string desciptor
00000018  0000          SLP$T_AUDST= 24                  ; Audit string
00000028  0000          SLP$Q_AUCDS= 40                  ; Current audit string descriptor
00000030  0000          SLP$T_AUCST= 48                  ; Current audit string
00000040  0000          SLP$Q_CMNT = 64                  ; Comment descriptor
00000048  0000          SLP$T_NAM  = 72                  ; NAM block
          0000    ;
000000A8  0000          SLP$K_BLN  = SLP$T_NAM + NAM$K_BLN
          0000    ;
          0000    ;
          0000    ; Macro to print error message
          0000    ;
          0000          .MACRO  ERRMSG  NAME,LIST
          0000          $$ = 0
          0000          .IRP    L,<LIST>
          0000          PUSHL   L
          0000          $$=$$+1
          0000          .ENDR
          0000          PUSHL   #$$
          0000          MOVL    #MER$_'NAME',R0
          0000          PUSHL   R0
          0000          CALLS   #$$+2,G^LIB$SIGNAL
          0000          .ENDM   ERRMSG
```

```
              0000          .TITLE  SUMED
              0000          .IDENT  /V04-000/
              0000
              0000   ;
              0000   ;****************************************************************************
              0000   ;*                                                                          *
              0000   ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
              0000   ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
              0000   ;*  ALL RIGHTS RESERVED.                                                    *
              0000   ;*                                                                          *
              0000   ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
              0000   ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
              0000   ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
              0000   ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
              0000   ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
              0000   ;*  TRANSFERRED.                                                            *
              0000   ;*                                                                          *
              0000   ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
              0000   ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
              0000   ;*  CORPORATION.                                                            *
              0000   ;*                                                                          *
              0000   ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
              0000   ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
              0000   ;*                                                                          *
              0000   ;*                                                                          *
              0000   ;****************************************************************************
              0000   ;
              0000   ; Procedure to merge slipr files - Main loop
              0000   ;
              0000   ;
              0000   ; Each input file is read and for every editing operation an
              0000   ; Edit node is created.  This node is linked into the list of editing
              0000   ; nodes in locator-1 sequence.  If there is an existing node with
              0000   ; the same locator-1 the new node will be linked in front of it.  This
              0000   ; gives it precedence to an earlier edit.  As each node is linked checks
              0000   ; for clashes are made.  If a clash is detected an error message is
              0000   ; printed.
              0000   ;
              0000   ; When all files have been read the complete Edit nodes list is processed
              0000   ; to form the merged output file
              0000   ;
              0000          $RABDEF
              0000
          00000000          .PSECT  $CODE,EXE,NOWRT
              0000   ;
              0000   ;
              0000   ;
              0000   ;
              0000   MERGE_FILES::
         0000 0000          .WORD   0
        07 10 0002          BSB     PASS_ONE
     03 50 E9 0004          BLBC    R0,10$
      03EF 30 0007          BSBW    PASS_TWO
              000A   10$:
           04 000A          RET
```

```
                            000B    ;
                            000B    ; Subroutine to read and process all input files
                            000B    ;
                            000B    ; Inputs:
                            000B    ;       Files list
                            000B    ;
                            000B    ; Outputs:
                            000B    ;       Edit nodes
                            000B    ;
                            000B    ;
                            000B    PASS_ONE:
     5A   0000'CF     DE    000B            MOVAL   W^FILE_NODES,R10            ; Get address of file nodes
                            0010    10$:
          5A   6A     D0    0010            MOVL    (R10),R10                   ; Get next node
  00000000'8F  5A     D1    0013            CMPL    R10,#FILE_NODES             ; At end of list?
               03     12    001A            BNEQ    5$                          ; No if NEQ
             00CF     31    001C            BRW     80$
                            001F    5$:
          53   5A     D0    001F            MOVL    R10,R3                      ; Form NAM block pointer
  53  00000048 8F     C0    0022            ADDL2   #SLP$T_NAM,R3
     54   0000'CF     DE    0029            MOVAL   W^INPUT_FAB,R4              ; and FAB pointer
                            002E            $FAB_STORE FAB = R4, NAM = (R3)
       58   04 A3     D0    0032            MOVL    NAM$L_RSA(R3),R8            ; Get file name in case of error
       59   03 A3     9A    0036            MOVZBL  NAM$B_RSL(R3),R9
                            003A            $OPEN   FAB = R4
            1E 50     E8    0043            BLBS    R0,20$                      ; OK if LBS
            0C A4     DD    0046            PUSHL   FAB$L_STV(R4)
               50     DD    0049            PUSHL   R0
                            004B            ERRMSG  REOPEN,<R8,R9>
             0083     31    0061            BRW     66$
                            0064    20$:
     52   0000'CF     DE    0064            MOVAL   W^INPUT_RAB,R2
                            0069            $CONNECT RAB = R2
            1D 50     E8    0072            BLBS    R0,30$                      ; OK if LBS
            0C A2     DD    0075            PUSHL   RAB$L_STV(R2)
               50     DD    0078            PUSHL   R0
                            007A            ERRMSG  CONNEC,<R8,R9>
               55     11    0090            BRB     66$
                            0092    30$:
               5B     10    0092            BSB     SET_UP_NODES
                            0094            $DISCONNECT RAB = R2
            1D 50     E8    009D            BLBS    R0,40$                      ; OK if LBS
            0C A2     DD    00A0            PUSHL   RAB$L_STV(R2)
               50     DD    00A3            PUSHL   R0
                            00A5            ERRMSG  DISCON,<R8,R9>
               2A     11    00BB            BRB     66$
                            00BD    40$:
                            00BD            $CLOSE  FAB = R4
            03 50     E9    00C6            BLBC    R0,50$                      ; Error if LBC
             FF44     31    00C9            BRW     10$
                            00CC    50$:
            0C A2     DD    00CC            PUSHL   FAB$L_STV(R2)
               50     DD    00CF            PUSHL   R0
                            00D1            ERRMSG  CLOSER,<R8,R9>
                            00E7    66$:
  00000000'GF  02     FB    00E7            CALLS   #2,G^LIB$SIGNAL             ; Signal error
                            00EE    80$:
```

```
05  00EE   114        RSB
```

```
                        00EF           .SBTTL  SET_UP_NODES
                        00EF     ;
                        00EF     ; Subroutine to form all edit_nodes
                        00EF     ;
                        00EF     ; Inputs:
                        00EF     ;        R2  = RAB address
                        00EF     ;        R8  = file spec address
                        00EF     ;        R9  = file spec size
                        00EF     ;        R10 = file node address
                        00EF     ;
                        00EF     ; Outputs:
                        00EF     ;        R0  = Success/error status
                        00EF     ;
                        00EF     ;
                        00EF     SET_UP_NODES:
          00DC 8F   BB  00EF           PUSHR   #^M<R2,R3,R4,R6,R7>
         10 A2  01  D0  00F3           MOVL    #1,RAB$W_RFA+0(R2)          ; Initialise record file address (3 words)
            14 A2   B4  00F7           CLRW    RAB$W_RFA+4(R2)
                        00FA     10$:
               7F   10  00FA           BSB     GET_NODE                    ; Get empty node
            77 50   E9  00FC           BLBC    R0,60$                      ; Error if LBC
         14 A5  5A  D0  00FF           MOVL    R10,ED$L_FILE(R5)           ; File address
      19 A5  0D AA  90  0103           MOVB    SLP$B_FILENO(R10), -        ; File number
                        0108                   ED$B_FILENO(R5)
      0E A5  10 A2  D0  0108           MOVL    RAB$W_RFA+0(R2),ED$W_RFA+0(R5) ; Record file adress (3 words)
      12 A5  14 A2  B0  010D           MOVW    RAB$W_RFA+4(R2),ED$W_RFA+4(R5)
            0C A5   B4  0112           CLRW    ED$W_LINES(R5)
                        0115     20$:
      08 A5  08 AA  B0  0115           MOVW    SLP$W_LOC1(R10),ED$W_LOC1(R5) ; Locator numbers
      0A A5  0A AA  B0  011A           MOVW    SLP$W_LOC2(R10),ED$W_LOC2(R5)
      18 A5  0C AA  90  011F           MOVB    SLP$B_FLAGS(R10),ED$B_FLAGS(R5) ; and flags byte
                        0124     30$:
             027D   30  0124           BSBW    READ_LINE                   ; Read line from input file
            4C 50   E9  0127           BLBC    R0,60$                      ; Error if LBC, process as end of edit
             05F4   30  012A           BSBW    COMMAND_CHECK               ; Check for command
            18 50   E8  012D           BLBS    R0,35$                      ; OK if LBS
                        0130           ERRMSG  SYNTAX,<R6,R7>              ; SLP command syntax error
               DC   11  0146           BRB     30$                         ; Ignore line
                        0148     35$:
         54    02   D3  0148           BITL    #2,R4                       ; Data terminating command?
               0A   12  014B           BNEQ    40$                         ; Yes if NEQ
         54    01   93  014D           BITB    #1,R4                       ; Data or normal command?
               D2   12  0150           BNEQ    30$                         ; Normal command if NEQ
            0C A5   B6  0152           INCW    ED$W_LINES(R5)              ; Increment number of insert lines for
               CD   11  0155           BRB     30$                         ; this edit
                        0157     40$:
            08 A5   B5  0157           TSTW    ED$W_LOC1(R5)               ; If Loc-1 and Loc-2 = 0 and Lines <> 0
               13   12  015A           BNEQ    50$                         ; there is an insert in front of
            0A A5   B5  015C           TSTW    ED$W_LOC2(R5)               ; the file, otherwise throw this
               0E   12  015F           BNEQ    50$                         ; Edit node away
            0C A5   B5  0161           TSTW    ED$W_LINES(R5)
               09   12  0164           BNEQ    50$
         54    04   D3  0166           BITL    #4,R4                       ; Was command "/"?
               04   13  0169           BEQL    50$                         ; No if EQL
               26   10  016B           BSB     RETURN_NODE                 ; Return node
               07   11  016D           BRB     60$
                        016F     50$:
```

```
             37   10  016F   173          BSB     INSERT_NODE             ; Insert node
        54   04   D3  0171   174          BITL    #4,R4                   ; Was last command "/"?
             84   13  0174   175          BEQL    10$                     ; No if EQL, back for next edit
                      0176   176 60$:
      00DC 8F     BA  0176   177          POPR    #^M<R2,R3,R4,R6,R7>
                  05  017A   178          RSB
```

```
                              017B          .SBTTL  GET_NODE
                              017B    ;
                              017B    ; Subroutine to get edit node
                              017B    ;
                              017B    ; Inputs:
                              017B    ;         none
                              017B    ;
                              017B    ; Outputs:
                              017B    ;         R0  = Success/error status
                              017B    ;         R5  = Node address
                              017B    ;
                              017B    ;
                              017B    GET_NODE:
           0000'CF      DF    017B          PUSHAL  W^VIRT_ADDR          ; Address to return VA
           0000'CF      DF    017F          PUSHAL  W^ED_SIZE            ; Size of edit node
00000000'GF     02      FB    0183          CALLS   #2,G^LIB$GET_VM      ; Get memory
             05 50      E9    018A          BLBC    R0,10$               ; Error if LBC
     55    0000'CF      D0    018D          MOVL    W^VIRT_ADDR,R5       ; Put node address in R5
                              0192    10$:
                        05    0192          RSB
```

```
                               0193          .SBTTL  RETURN_NODE
                               0193  ;
                               0193  ; Subroutine to return edit node
                               0193  ;
                               0193  ; Inputs:
                               0193          R5 = Node address
                               0193  ;
                               0193  ; Outputs:
                               0193          R0 = Success/error status
                               0193  ;
                               0193  ;
                               0193  RETURN_NODE:
     0000'CF      55     D0    0193          MOVL    R5,W^VIRT_ADDR          ; Store address of node
         0000'CF         DF    0198          PUSHAL  W^VIRT_ADDR             ; Push parameters
         0000'CF         DF    019C          PUSHAL  W^ED_SIZE
 00000000'GF      02     FB    01A0          CALLS   #2,G^LIB$FREE_VM
                  05           01A7          RSB
```

```
                          01A8          .SBTTL  INSERT_NODE
                          01A8   ;
                          01A8   ; Subroutine to insert node into edit list and report any clashes
                          01A8   ; with previous edits
                          01A8   ;
                          01A8   ; Inputs:
                          01A8          R5  = address of node to insert
                          01A8   ;
                          01A8   ; Outputs:
                          01A8          None
                          01A8   ;
                          01A8   ;
                          01A8  INSERT_NODE:
                1C   BB   01A8          PUSHR   #^M<R2,R3,R4>
    53   0000'CF     DE   01AA          MOVAL   W^EDIT_NODES,R3              ; Get address of list head
                52   D4   01AF          CLRL    R2                           ; Initialise clash count
                          01B1  10$:
           53   63   D0   01B1          MOVL    (R3),R3                      ; Get next node
 00000000'8F    53   D1   01B4          CMPL    R3,#EDIT_NODES               ; At end of list?
           12   13        01BB          BEQL    20$                          ; Yes if EQL
  08 A3   08 A5      B1   01BD          CMPW    ED$W_LOC1(R5),ED$W_LOC1(R3) ; Is new LOC-1 <= current LOC-1
                0B   15   01C2          BLEQ    20$                          ; Yes if LEQ, found place for node
  0A A3   08 A5      B1   01C4          CMPW    ED$W_LOC1(R5),ED$W_LOC2(R3) ; Does current LOC-2 overlap with new LO
                E6   14   01C9          BGTR    10$                          ; No if GTR
                29   10   01CB          BSB     REPORT_CLASH
                E2   11   01CD          BRB     10$
                          01CF  20$:
      04 B3     65   0E   01CF          INSQUE  (R5),@ED$L_BWD(R3)           ; Insert new node into list
      54   0A A5     3C   01D3          MOVZWL  ED$W_LOC2(R5),R4             ; Get LOC-2 value
                04   12   01D7          BNEQ    30$
      54   08 A5     3C   01D9          MOVZWL  ED$W_LOC1(R5),R4
                          01DD  30$:
 00000000'8F    53   D1   01DD          CMPL    R3,#EDIT_NODES               ; End of list?
                0D   13   01E4          BEQL    40$                          ; Yes if EQL
      08 A3     54   B1   01E6          CMPW    R4,ED$W_LOC1(R3)             ; Does edit clash?
                07   19   01EA          BLSS    40$                          ; No if LSS
                08   10   01EC          BSB     REPORT_CLASH
           53   63   D0   01EE          MOVL    (R3),R3                      ; Advance to next node
                EA   11   01F1          BRB     30$
                          01F3  40$:
                1C   BA   01F3          POPR    #^M<R2,R3,R4>
                     05   01F5          RSB
```

```
                                  01F6                  .SBTTL  REPORT_CLASH
                                  01F6          ;
                                  01F6          ; Subroutine to report clashes between edits
                                  01F6          ;
                                  01F6          ; Inputs:
                                  01F6          ;       R2  = Number of clashes
                                  01F6          ;       R3  = Node address of clashing edit
                                  01F6          ;       R5  = Node address of new edit
                                  01F6          ;
                                  01F6          ; Outputs:
                                  01F6          ;       R2  = Number of clashes
                                  01F6          ;
                                  01F6          ;
                                  01F6          REPORT_CLASH:
                      08 A3   B5  01F6                  TSTW    ED$W_LOC1(R3)           ; If LOC-1, LOC-2 and LINES all
                         0A   12  01F9                  BNEQ    5$                      ; equal zero node is not full edit
                      0A A3   B5  01FB                  TSTW    ED$W_LOC2(R3)
                         05   12  01FE                  BNEQ    5$
                      0C A3   B5  0200                  TSTW    ED$W_LINES(R3)
                         36   13  0203                  BEQL    20$
                                  0205          5$:
   09 18 A3   00000000'8F     E1  0205                  BBC     #SLPV_SUPPRESS, -       ; Branch if clash report has
                                  020E                          ED$B_FLAGS(R3),7$       ; not been suppressed
   24 18 A5   00000000'8F     E0  020E                  BBS     #SLPV_SUPPRESS, -       ; Branch if clash report has
                                  0217                          ED$B_FLAGS(R5),20$      ; been suppressed
                                  0217          7$:
                         52   D5  0217                  TSTL    R2                      ; Is this first clash?
                         1C   12  0219                  BNEQ    10$                     ; No if NEQ
                         53   DD  021B                  PUSHL   R3                      ; Save R3
                    53   55   D0  021D                  MOVL    R5,R3
                                  0220                  ERRMSG  CLASH                   ; Report new clash
                         08   10  0232                  BSB     CLASH_LINE              ; Print edit line
                         53 8ED0  0234                  POPL    R3                      ; Restore R3
                                  0237          10$:
                         52   D6  0237                  INCL    R2                      ; Increment number of clashes
                         01   10  0239                  BSB     CLASH_LINE              ; Print edit line
                                  023B          20$:
                              05  023B                  RSB
```

```
                              023C          .SBTTL  CLASH_LINE
                              023C  ;
                              023C  ; Subroutine to print random line from slp input file
                              023C  ;
                              023C  ;
                              023C  ; Inputs:
                              023C  ;       R3 = Edit node address
                              023C  ;
                              023C  ; Outputs:
                              023C  ;       None
                              023C  ;
                              023C  ;
                              023C  CLASH_LINE:
          07FC 8F      BB     023C          PUSHR   #^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
               6B      10     0240          BSB     RANDOM_OPEN                  ; Open file for random access
            5C 50      E9     0242          BLBC    R0,70$                       ; Error if LBC
     52    0000'CF     DE     0245          MOVAL   W^RANDOM_RAB,R2         ; Set pointer to RAB
  10 A2      0E A3     D0     024A          MOVL    ED$W_RFA+0(R3),RAB$W_RFA+0(R2) ; Copy record file address
  14 A2      12 A3     B0     024F          MOVW    ED$W_RFA+4(R3),RAB$W_RFA+4(R2) ; (3 words)
                              0254          $GET    RAB = R2
            1D 50      E8     025D          BLBS    R0,60$
            0C A2      DD     0260          PUSHL   RAB$L_STV(R2)
               50      DD     0263          PUSHL   R0
                              0265          ERRMSG  READER,<R8,R9>
               24      11     027B          BRB     70$
                              027D  60$:
     56      28 A2     D0     027D          MOVL    RAB$L_RBF(R2),R6
     57      22 A2     3C     0281          MOVZWL  RAB$W_RSZ(R2),R7
                              0285          ERRMSG  CLSHLN,<R6,R7,R8,R9>
               07      11     029F          BRB     80$
                              02A1  70$:
 00000000'GF   02      FB     02A1          CALLS   #2,G^LIB$SIGNAL
                              02A8  80$:
          07FC 8F      BA     02A8          POPR    #^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
                       05     02AC          RSB
```

```
                        02AD          .SBTTL  RANDOM_OPEN
                        02AD  ;
                        02AD  ; Subroutine to open a slp input file for random access.  If the
                        02AD  ; file is the currently open file then no file access are performed.
                        02AD  ; Otherwise, the current file is close and disonnected from the RAB
                        02AD  ; and the rqueisted file opened and connected to the RAB
                        02AD  ;
                        02AD  ;
                        02AD  ; Inputs:
                        02AD  ;       R3 = Edit node address
                        02AD  ;
                        02AD  ; Outputs:
                        02AD  ;       R8 = Address of current file specification
                        02AD  ;       R9 = Size of current file specification
                        02AD  ;
                        02AD  ;
                        02AD RANDOM_OPEN:
         04FC 8F    BB  02AD          PUSHR   #^M<R2,R3,R4,R5,R6,R7,R10>
           50 01    D0  02B1          MOVL    #1,R0                          ; Assume success
        5A  14 A3   D0  02B4          MOVL    ED$L_FILE(R3),R10              ; Get file node address
       52  0000'CF  DE  02B8          MOVAL   W^RANDOM_RAB,R2                ; Set RAB pointer
       54  0000'CF  DE  02BD          MOVAL   W^RANDOM_FAB,R4                ; FAB pointer
           55  5A   D0  02C2          MOVL    R10,R5                         ; NAM pointer
   55  00000048 8F  C0  02C5          ADDL2   #SLP$T_NAM,R5
        58  04 A5   D0  02CC          MOVL    NAM$L_RSA(R5),R8               ; Pointers to file spec
        59  03 A5   9A  02D0          MOVZBL  NAM$B_RSL(R5),R9               ; string and size
          0000'CF   D5  02D4          TSTL    W^RANDOM_FILE                  ; Is a file open?
               61   13  02D8          BEQL    20$                            ; No if EQL
     0000'CF   5A   D1  02DA          CMPL    R10,W^RANDOM_FILE              ; Is it file required?
               03   12  02DF          BNEQ    5$                             ; No if NEQ
             00BB   31  02E1          BRW     80$
                        02E4 5$:
          0000'CF   D4  02E4          CLRL    W^RANDOM_FILE                  ; Close current file
                        02E8          $DISCONNECT RAB = R2
            1E 50   E8  02F1          BLBS    R0,10$
            0C A2   DD  02F4          PUSHL   RAB$L_STV(R2)
               50   DD  02F7          PUSHL   R0
                        02F9          ERRMSG  DISCON,<R8,R9>
             0086   31  030F          BRW     70$
                        0312 10$:
                        0312          $CLOSE  FAB = R4
            1D 50   E8  031B          BLBS    R0,20$
            0C A4   DD  031E          PUSHL   FAB$L_STV(R4)
               50   DD  0321          PUSHL   R0
                        0323          ERRMSG  CLOSER,<R8,R9>
               5D   11  0339          BRB     70$
                        033B 20$:
                        033B          $FAB_STORE FAB = R4, NAM = (R5)
                        033F          $OPEN   FAB = R4
            1D 50   E8  0348          BLBS    R0,30$
            0C A4   DD  034B          PUSHL   FAB$L_STV(R4)
               50   DD  034E          PUSHL   R0
                        0350          ERRMSG  REOPEN,<R8,R9>
               30   11  0366          BRB     70$
                        0368 30$:
                        0368          $CONNECT RAB = R2
            1D 50   E8  0371          BLBS    R0,40$
```

```
        0C A2   DD  0374         PUSHL   RAB$L_STV(R2)
           50   DD  0377         PUSHL   R0
                    0379         ERRMSG  CONNEC,<R8,R9>
           07   11  038F         BRB     70$
                    0391  40$:
   0000'CF 5A   D0  0391         MOVL    R10,W^RANDOM_FILE          ; Set new file open
           07   11  0396         BRB     80$
                    0398  70$:
00000000'GF 02  FB  0398         CALLS   #2,G^LIB$SIGNAL
                    039F  80$:
      04FC 8F   BA  039F         POPR    #^M<R2,R3,R4,R5,R6,R7,R10>
                05  03A3         RSB
```

```
                          03A4          .SBTTL  READ_LINE
                          03A4   ;
                          03A4   ; Subroutine to read line sequentially from current slipr input file
                          03A4   ;
                          03A4   ; Inputs:
                          03A4   ;      R8 = Address of file spec
                          03A4   ;      R9 = Size of file spec
                          03A4   ;
                          03A4   ; Outputs:
                          03A4   ;      R0 = success/error status
                          03A4   ;      R6 = Line buffer address
                          03A4   ;      R7 = Line size
                          03A4   ;
                          03A4   ;
                          03A4  READ_LINE:
                   52  DD 03A4          PUSHL   R2
         52  0000'CF   DE 03A6          MOVAL   W^INPUT_RAB,R2          ; Get RAB address
                          03AB          $GET    RAB = R2                ; Read line
                36 50  E8 03B4          BLBS    R0,20$                  ; OK if LBS
   00000000'8F     50  D1 03B7          CMPL    R0,#RMS$_EOF            ; Unexpected EOF?
                   1F  12 03BE          BNEQ    10$                     ; No if NEQ
                          03C0          ERRMSG  INPEOF,<R8,R9>          ; Report error
    50  00000000'8F    D0 03D6          MOVL    #RMS$_EOF,R0            ; Reset R0 to indicate error
                   16  11 03DD          BRB     30$
                          03DF  10$:
                0C A2  DD 03DF          PUSHL   RAB$L_STV(R2)           ; Signal error
                   50  DD 03E2          PUSHL   R0
   00000000'GF     02  FB 03E4          CALLS   #2,G^LIB$SIGNAL
                   08  11 03EB          BRB     30$
                          03ED  20$:
         56     28 A2  D0 03ED          MOVL    RAB$L_RBF(R2),R6        ; Set buffer address
         57     22 A2  3C 03F1          MOVZWL  RAB$W_RSZ(R2),R7        ; and line size
                          03F5  30$:
                52 8ED0   03F5          POPL    R2
                   05     03F8          RSB
```

```
                              03F9        .SBTTL  PASS_TWO
                              03F9  ;
                              03F9  ; Subroutine to perform Pass 2
                              03F9  ;
                              03F9  ; This routine processes each edit node, by forming and outputing the
                              03F9  ; editing command and then copying any data lines.  For each edit it
                              03F9  ; checks that the current audit trail switch (on or off) and audit
                              03F9  ; trail string are what are required for that file.  If not commands
                              03F9  ; are generated and output to set the audit trail switch and string
                              03F9  ; to the correct values
                              03F9  ;
                              03F9  ;
                              03F9  ; Inputs:
                              03F9  ;       Edit nodes
                              03F9  ;
                              03F9  ; Outputs:
                              03F9  ;       None
                              03F9  ;
                              03F9  ; Define R2 flag bits
                              03F9  ;
                              03F9  _VIELD  SUM,0,< -
                              03F9          <EXPED,,M> -                    ; Expected edit command has been seen
                              03F9          <1STEDIT,,M> -                  ; 1st edit has been processed
                              03F9          <SUPPRESS,,M> -                 ; Edit is '--' form
                              03F9          <SPRCDE,,M> -                   ; Insert/replace lines being superceded
                              03F9          >
                              03F9  ;
                              03F9  ;
                              03F9  PASS_TWO:
       53    0000'CF    DE    03F9          MOVAL   W^EDIT_NODES,R3         ; Set up edit nodes pointer
                              03FE  10$:
          53    63      D0    03FE          MOVL    (R3),R3                 ; Get next node
  00000000'8F   53      D1    0401          CMPL    R3,#EDIT_NODES          ; End of list?
                0B      13    0408          BEQL    20$                     ; Yes if EQL
                52      D4    040A          CLRL    R2                      ; Clear all flags
              00AD      30    040C          BSBW    RANGE_EDIT              ; Find range of next edit
              000F      30    040F          BSBW    PROCESS_EDIT            ; Process next edit
             E9 50      E8    0412          BLBS    R0,10$                  ; All ok if LBS
                              0415  20$:
       56    0000'CF    DE    0415          MOVAL   W^END_EDIT,R6           ; Get address of end edit line
          57    01      D0    041A          MOVL    #1,R7                   ; and set its size
              02BB      30    041D          BSBW    WRITE_LINE              ; Write line to output file
                05            0420          RSB
```

```
                        0421          .SBTTL  PROCESS_EDIT
                        0421    ;
                        0421    ; Subroutine to process next edit operation
                        0421    ;
                        0421    ; Inputs:
                        0421    ;
                        0421    ;         R2  = Flags
                        0421    ;         R3  = First edit node address
                        0421    ;         R5  = Last edit node address
                        0421    ;         R11 = Highest loc-2 of edit
                        0421    ;
                        0421    ; Outputs:
                        0421    ;
                        0421    ;         R3  = Last edit node address
                        0421
                        0421    ;
                        0421    ;
                        0421    PROCESS_EDIT:
             53    DD   0421            PUSHL   R3                          ; Save first node address
                        0423    5$:
      5A  14 A3    D0   0423            MOVL    ED$L_FILE(R3),R10           ; Get file node address
           00DE    30   0427            BSBW    ACCESS_EDIT                 ; Set up RMS to read edit
          03 50    E8   042A            BLBS    R0,7$                       ; OK if LBS
           0088    31   042D            BRW     120$                        ; Error
                        0430    7$:
          08 A3    B5   0430            TSTW    ED$W_LOC1(R3)               ; Is loc-1 zero?
             03    12   0433            BNEQ    10$                         ; No if NEQ
      52     01    C8   0435            BISL2   #SUM_M_EXPED,R2             ; Set expected edit seen flag
                        0438    10$:
           00EC    30   0438            BSBW    READ_RECORD                 ; Get next record
          7A 50    E9   043B            BLBC    R0,120$                     ; Error if LBC
           02E0    30   043E            BSBW    COMMAND_CHECK               ; Check for command
          F4 50    E9   0441            BLBC    R0,10$                      ; Error if LBC
       12 54 00    E1   0444            BBC     #0,R4,30$                   ; Branch if data line
       17 54 01    E0   0448            BBS     #1,R4,40$                   ; Branch if edit terminating command
                        044C    15$:
          10 AA    B5   044C            TSTW    SLP$Q_AUDDS(R10)            ; Was new audit trail specified?
             03    13   044F            BEQL    20$                         ; No if EQL
           0114    30   0451            BSBW    RESET_AUDIT                 ; Set new current audit string
                        0454    20$:
 0000'CF  0C AA    9B   0454            MOVZBW  SLP$B_FLAGS(R10),W^SLP_FLAGS ; Set new current flags
                        045A    30$:
    DA 52    03    E0   045A            BBS     #SUM_V_SPRCDE,R2,10$        ; Branch if edit being superceded
           027A    30   045E            BSBW    WRITE_LINE                  ; Write line to output file
             D5    11   0461            BRB     10$                         ; Back for next node
                        0463    40$:
    1F 52    00    E2   0463            BBSS    #SUM_V_EXPED,R2,70$         ; Is this expected edit command?
    E1 52    03    E0   0467            BBS     #SUM_V_SPRCDE,R2,15$        ; Branch if edit is being superceded
    06 52    01    E2   046B            BBSS    #SUM_V_1STEDIT,R2,50$       ; Is this first edit of range?
    0A AA    5B    B0   046F            MOVW    R11,SLP$W_LOC2(R10)         ; Yes: set Loc-2 to highest in range
             09    11   0473            BRB     60$
                        0475    50$:
       08 AA       B4   0475            CLRW    SLP$W_LOC1(R10)             ; Zero both Loc-1
       0A AA       B4   0478            CLRW    SLP$W_LOC2(R10)             ; and Loc-2
       52    04    CA   047B            BICL2   #SUM_M_SUPPRESS,R2          ; Only first edit may be '--' form
                        047E    60$:
           010C    30   047E            BSBW    WRITE_EDIT                  ; Write edit line
```

```
       B4 50    E8  0481         BLBS    R0,10$                        ; OK if LBS
          32    11  0484         BRB     120$
                    0486  70$:
       52 09    CA  0486         BICL2   #SUM_M_EXPED!SUM_M_SPRCDE,R2 ; Clear flags
       55 53    D1  0489         CMPL    R3,R5                         ; Last edit in range?
          2A    13  048C         BEQL    120$                          ; Yes if EQL
       54 53    D0  048E         MOVL    R3,R4                         ; Set temp node pointer
       53 63    D0  0491         MOVL    (R3),R3                       ; Move to next node
                    0494  80$:
 19 A3  19 A4   91  0494         CMPB    ED$B_FILENO(R4), -            ; Is file number of this edit =>
                    0499                 ED$B_FILENO(R3)               ; file number of next edit?
          0C    19  0499         BLSS    90$                           ; No if LSS, inserts are not superceded
       08 A4    B5  049B         TSTW    ED$W_LOC1(R4)                 ; Insert to front of file?
          07    13  049E         BEQL    90$                           ; Yes if EQL, insert is not superceded
 08 A3  0A A4   B1  04A0         CMPW    ED$W_LOC2(R4), -              ; Does edit's loc-2 reach next's loc-1
                    04A5                 ED$W_LOC1(R3)
          0B    18  04A5         BGEQ    100$                          ; Yes if GEQ, inserts are superceded
                    04A7  90$:
       6E 54    D1  04A7         CMPL    R4,(SP)                       ; Back to first node?
          09    13  04AA         BEQL    110$                          ; Yes if EQL
    54  04 A4   D0  04AC         MOVL    ED$L_BWD(R4),R4               ; Go back to previous node
          E2    11  04B0         BRB     80$                           ; and test again
                    04B2  100$:
       52 08    88  04B2         BISB    #SUM_M_SPRCDE,R2              ; Set inserts to be superceded flags
                    04B5  110$:
        FF6B    31  04B5         BRW     5$                            ; Process next edit
                    04B8  120$:
       5E 04    C0  04B8         ADDL2   #4,SP                         ; Clear stack
                05  04BB         RSB
```

```
                              04BC          .SBTTL  RANGE_EDIT
                              04BC    ;
                              04BC    ; Subroutine to find range of next edit operation
                              04BC    ;
                              04BC    ; Inputs:
                              04BC    ;
                              04BC    ;       R2 = Flags
                              04BC    ;       R3 = Current (first) node
                              04BC    ;
                              04BC    ; Outputs:
                              04BC    ;
                              04BC    ;       R2 = Flags
                              04BC    ;       R5 = Last node of edit
                              04BC    ;       R11= Highest locator-2 number of edit range
                              04BC    ;
                              04BC    ;
                              04BC    RANGE_EDIT:
                  53     DD   04BC          PUSHL   R3                        ; Save R3
               52 04     C8   04BE          BISL2   #SUM_M_SUPPRESS,R2        ; Assume all clash messages suppressed
            5B  0A A3    3C   04C1          MOVZWL  ED$W_LOC2(R3),R11         ; Initialise highest locator-2
               55 53     D0   04C5          MOVL    R3,R5                     ; and current node pointer
                              04C8    10$:
03 18 A3   00000000'8F   E0   04C8          BBS     #SLPV_SUPPRESS, -         ; Branch if '--' form of edit
                              04D1                  ED$B_FLAGS(R3),15$
               52 04     CA   04D1          BICL2   #SUM_M_SUPPRESS,R2        ; Edit will be '-' form
                              04D4    15$:
               53 65     D0   04D4          MOVL    (R5),R3                   ; Point to next edit node
      00000000'8F 53     D1   04D7          CMPL    R3,#EDIT_NODES            ; Is it end of list?
                  24     13   04DE          BEQL    30$                       ; Yes if EQL
               51 5B     D0   04E0          MOVL    R11,R1                    ; Get highest locator-2
                  0A     12   04E3          BNEQ    20$                       ; If non-zero compare with this
            51  0A A5    3C   04E5          MOVZWL  ED$W_LOC2(R5),R1          ; Get current locator-2
                  04     12   04E9          BNEQ    20$                       ; If zero use
            51  08 A5    3C   04EB          MOVZWL  ED$W_LOC1(R5),R1          ; locator-1
                              04EF    20$:
            08 A3  51    B1   04EF          CMPW    R1,ED$W_LOC1(R3)          ; Does this edit overlap with next?
                  0F     19   04F3          BLSS    30$                       ; No if LSS
               55 53     D0   04F5          MOVL    R3,R5                     ; Set next node to be current node
            0A A5  5B    B1   04F8          CMPW    R11,ED$W_LOC2(R5)         ; Is locator-2 higher then current
                  CA     18   04FC          BGEQ    10$                       ; No if GEQ
            5B  0A A5    3C   04FE          MOVZWL  ED$W_LOC2(R5),R11         ; Reset with higher number
                  C4     11   0502          BRB     10$
                              0504    30$:
                53 8ED0       0504          POPL    R3                        ; Restore R3
                     05       0507          RSB
```

```
                      0508          .SBTTL  ACCESS_EDIT
                      0508  ;
                      0508  ; Subroutine to set up RMS 32 to read an edit from an input file
                      0508  ;
                      0508  ; Inputs:
                      0508  ;       R3 = Edit node address
                      0508  ;
                      0508  ; Outputs:
                      0508  ;       RAB block set up for RFA access to edit lines
                      0508  ;
                      0508  ;
                      0508  ACCESS_EDIT:
              52   DD 0508          PUSHL   R2
   52    0000'CF   DE 050A          MOVAL   W^RANDOM_RAB,R2            ; Set pointer to RAB
           FD9B    30 050F          BSBW    RANDOM_OPEN                ; Get correct file open
           0E 50   E9 0512          BLBC    R0,10$                     ; Error if LBC
 10 A2     0E A3   D0 0515          MOVL    ED$W_RFA+0(R3),RAB$W_RFA+0(R2) ; Copy RFA (3 words)
 14 A2     12 A3   B0 051A          MOVW    ED$W_RFA+4(R3),RAB$W_RFA+4(R2)
   1E A2     02    90 051F          MOVB    #RAB$C_RFA,RAB$B_RAC(R2)       ; Set into RFA access mode
                      0523  10$:
            52 8ED0   0523          POPL    R2
                 05   0526          RSB
```

```
                         0527          .SBTTL  READ_RECORD
                         0527  ;
                         0527  ; Subroutine to read edit record.  The first record is read from
                         0527  ; a random position, subsequent records are read sequentially from
                         0527  ; the first.  To achieve this this routine changes the record access
                         0527  ; mode from RFA to SEQ after reading the first record
                         0527  ;
                         0527  ;
                         0527  ; Inputs:
                         0527  ;       R8 = address of file spec string
                         0527  ;       R9 = size of file spec string
                         0527  ;
                         0527  ; Outputs:
                         0527  ;       R6 = Record address
                         0527  ;       R7 = Record size
                         0527  ;
                         0527  ;
                         0527  READ_RECORD:
                  52 DD  0527          PUSHL   R2
          52  0000'CF DE 0529          MOVAL   W^RANDOM_RAB,R2         ; Set pointer to RAB
                         052E          $GET    RAB = R2                ; Get line
               18 50 E8  0537          BLBS    R0,10$                  ; OK if LBS
                         053A          ERRMSG  READER,<R8,R9>
                  12 11  0550          BRB     20$
                         0552  10$:
            56 28 A2 D0  0552          MOVL    RAB$L_RBF(R2),R6        ; Set buffer address
            57 22 A2 3C  0556          MOVZWL  RAB$W_RSZ(R2),R7        ; and line size
            02 1E A2 91  055A          CMPB    RAB$B_RAC(R2),#RAB$C_RFA ; Is it in RFA mode?
                  04 12  055E          BNEQ    20$                     ; No if NEQ
            1E A2 00 90  0560          MOVB    #RAB$C_SEQ,RAB$B_RAC(R2) ; Set into SEQ access mode
                         0564  20$:
                52 8ED0  0564          POPL    R2
                     05  0567          RSB
```

```
                                    0568          .SBTTL  RESET_AUDIT
                                    0568  ;
                                    0568  ; Subroutine to set new current audit trail string
                                    0568  ;
                                    0568  ; Inputs:
                                    0568          R10 = File node address
                                    0568  ;
                                    0568  ; Outputs:
                                    0568          None
                                    0568  ;
                                    0568          .ENABL  LSB
                                    0568  ;
                                    0568  ;
                                    0568  RESET_AUDIT:
                           3C   BB  0568          PUSHR   #^M<R2,R3,R4,R5>
      30 AA    18 AA    10 AA   28  056A          MOVC3   SLP$Q_AUDDS(R10), -     ; Copy audit string
                                    0571                  SLP$T_AUDST(R10), -
                                    0571                  SLP$T_AUCST(R10)
               28 AA    10 AA   B0  0571          MOVW    SLP$Q_AUDDS(R10),SLP$Q_AUCDS(R10) ; and string length
                           02   11  0576          BRB     10$
                                    0578  ;
                                    0578  RESET_CURRENT:
                           3C   BB  0578          PUSHR   #^M<R2,R3,R4,R5>
                                    057A  10$:
00000000'EF    30 AA    28 AA   28  057A          MOVC3   SLP$Q_AUCDS(R10), -     ; Copy audit string
                                    0584                  SLP$T_AUCST(R10), -
                                    0584                  SLP_AUDST
               0000'CF  28 AA   B0  0584          MOVW    SLP$Q_AUCDS(R10),W^SLP_AUDDS ; and string length
                           3C   BA  058A          POPR    #^M<R2,R3,R4,R5>
                           05       058C          RSB
                                    058D  ;
                                    058D          .DSABL  LSB
```

```
                                 058D          .SBTTL  WRITE_EDIT
                                 058D  ;
                                 058D  ; Subroutine to form and write edit command to output file
                                 058D  ;
                                 058D  ; Inputs:
                                 058D  ;       R2  = Flags
                                 058D  ;       R3  = Edit node address
                                 058D  ;       R10 = File node address
                                 058D  ;
                                 058D  ; Outputs:
                                 058D  ;       None
                                 058D  ;
                                 058D  ;
                                 058D  WRITE_EDIT:
   51   0C AA   0000'CF    8D    058D          XORB3   W^SLP_FLAGS, -                ; Is audit trail switch for this file
                                 0594                  SLP$B_FLAGS(R10),R1           ; different to current setting?
           51   0000'8F    B3    0594          BITW    #SLPM_AUDIT,R1                ;
                     2C    13    0599          BEQL    30$                           ; No if EQL
                57   01    D0    059B          MOVL    #1,R7                         ; Set output line size
     0000'CF    0000'8F    B3    059E          BITW    #SLPM_AUDIT,W^SLP_FLAGS       ; Was audit ON?
                     0E    12    05A5          BNEQ    10$                           ; Yes if NEQ
           56   0000'CF    DE    05A7          MOVAL   W^AUDIT_ON,R6                 ; It was OFF so turn ON
     0000'CF    0000'8F    A8    05AC          BISW    #SLPM_AUDIT,W^SLP_FLAGS
                     12    11    05B3          BRB     30$
                                 05B5  10$:
           56   0000'CF    DE    05B5          MOVAL   W^AUDIT_OFF,R6                ; It was ON so turn OFF
     0000'CF    0000'8F    AA    05BA          BICW    #SLPM_AUDIT,W^SLP_FLAGS
                                 05C1  20$:
                   0117    30    05C1          BSBW    WRITE_LINE                    ; Write command to file
                  EE 50    E9    05C4          BLBC    R0,10$                        ; Error of LBC
                                 05C7  30$:
                     57    D4    05C7          CLRL    R7                            ; Initialise line size
                     54    D4    05C9          CLRL    R4                            ; Zero comment count
     0000'CF    0200 8F    3C    05CB          MOVZWL  #BUF_SIZE,W^OUTDES            ; Initialise output buffer
     0004'CF    0000'CF    DE    05D2          MOVAL   W^OUTPUT_BUF,W^OUTDES+4       ; descriptor
             0D 52   02    E1    05D9          BBC     #SUM_V_SUPPRESS,R2,32$        ; '--' form of edit?
        0004'DF      2D    90    05DD          MOVB    #^A/-/,@W^OUTDES+4            ; Insert first '-'
        0000'CF      01    D0    05E2          MOVL    #1,W^OUTLEN
                   00DD    30    05E7          BSBW    ADJUST_COUNTS
                                 05EA  32$:
                  08 AA    B5    05EA          TSTW    SLP$W_LOC1(R10)               ; Is there a locator-1
                     18    13    05ED          BEQL    33$                           ; No if EQL
                                 05EF          $FAO_S  W^LOC_ONE,W^OUTLEN, -         ; Convert and store locator-1
                                 05EF                  W^OUTDES,SLP$W_LOC1(R10)
                     0A    11    0605          BRB     37$
                                 0607  33$:
        0004'DF      2D    90    0607          MOVB    #^A/-/,@W^OUTDES+4            ; Insert a '-'
        0000'CF      01    D0    060C          MOVL    #1,W^OUTLEN
                                 0611  37$:
                   00B3    30    0611          BSBW    ADJUST_COUNTS
                                 0614  ;
                  0A AA    B5    0614          TSTW    SLP$W_LOC2(R10)               ; Is there a locator-2?
                     1B    13    0617          BEQL    40$                           ; No if EQL
                                 0619          $FAO_S  W^LOC_TWO,W^OUTLEN, -         ; Convert and store locator-2
                                 0619                  W^OUTDES,SLP$W_LOC2(R10)
                   0095    30    062F          BSBW    ADJUST_COUNTS
                     54    D6    0632          INCL    R4                            ; One comma now in line
```

```
                                 0634    40$:
                    10 AA   B5   0634            TSTW    SLP$Q_AUDDS(R10)          ; Audit trail in this command?
                       05   13   0637            BEQL    50$                       ; No if EQL
                     FF2C   30   0639            BSBW    RESET_AUDIT               ; Reset audit trail
                       15   11   063C            BRB     60$
                                 063E    50$:
                       2D   BB   063E            PUSHR   #^M<R0,R2,R3,R5>
28 AA    00    0000'CF    0000'CF   2D   0640    CMPC5   W^SLP_AUDDS,W^SLP_AUDST,#0, - ; Is audit trail required for this
                    30 AA        064A
                                 064C            SLP$Q_AUCDS(R10),SLP$T_AUCST(R10) ; file equal to current?
                       2D   BA   064C            POPR    #^M<R0,R2,R3,R5>
                       35   13   064E            BEQL    80$                       ; Yes if EQL
                     FF25   30   0650            BSBW    RESET_CURRENT             ; Reset current audit string
                                 0653    60$:
                  01   54   D1   0653            CMPL    R4,#1                     ; One comma on line?
                       10   18   0656            BGEQ    70$                       ; Yes if GEQ
              0004'DF  2C   90   0658            MOVB    #^A/,/,@W^OUTDES+4        ; Insert comma
              0000'CF  01   D0   065D            MOVL    #1,W^OUTLEN
                       63   10   0662            BSB     ADJUST_COUNTS
                       54   D6   0664            INCL    R4                        ; Increment number of commas
                       EB   11   0666            BRB     60$
                                 0668    70$:
                                 0668            $FAO_S  W^AUDIT_TRAIL,W^OUTLEN, - ; Insert audit trail
                                 0668                    W^OUTDES,#SLP_AUDDS
                       44   10   0681            BSB     ADJUST_COUNTS
                       54   D6   0683            INCL    R4                        ; Increment number of commas on line
                                 0685    80$:
                    40 AA   B5   0685            TSTW    SLP$Q_CMNT(R10)           ; Comment in command?
                       30   13   0688            BEQL    110$                      ; No if EQL
                                 068A    90$:
                  02   54   D1   068A            CMPL    R4,#2                     ; Two commas on line
                       10   18   068D            BGEQ    100$                      ; Yes if GEQ
              0004'DF  2C   90   068F            MOVB    #^A/,/,@W^OUTDES+4        ; Insert comma
              0000'CF  01   D0   0694            MOVL    #1,W^OUTLEN
                       2C   10   0699            BSB     ADJUST_COUNTS
                       54   D6   069B            INCL    R4                        ; Increment number of commas
                       EB   11   069D            BRB     90$
                                 069F    100$:
               51   40 AA   DE   069F            MOVAL   SLP$Q_CMNT(R10),R1
                                 06A3            $FAO_S  W^COMMENTS,W^OUTLEN, -    ; Insert comment into line
                                 06A3                    W^OUTDES,R1
                       0D   10   06B8            BSB     ADJUST_COUNTS
                                 06BA    110$:
                  01   57   D1   06BA            CMPL    R7,#1                     ; Is line only '-'?
                       07   13   06BD            BEQL    120$                      ; Yes if EQL so dont output it
              56  0000'CF   DE   06BF            MOVAL   W^OUTPUT_BUF,R6           ; Set buffer pointer
                       15   10   06C4            BSB     WRITE_LINE
                                 06C6    120$:
                       05        06C6            RSB
                                 06C7    ;
                                 06C7    ;
                                 06C7    ;
                                 06C7    ; Subroutine to adjust output buffer descriptor as each
                                 06C7    ; element of command line is inserted
                                 06C7    ;
                                 06C7    ;
                                 06C7    ; Inputs:
```

```
                          06C7  ;        R7 = Buffer size
                          06C7  ;        OUTLEN = Length of last entry
                          06C7  ;
                          06C7  ; Outputs:
                          06C7  ;        R7 = New buffer length
                          06C7  ;        OUTDES = Describes remainder of buffer
                          06C7  ;
                          06C7  ;
                          06C7 ADJUST_COUNTS:
      57   0000'CF    A0  06C7         ADDW2   W^OUTLEN,R7              ; Form new line length
 0000'CF   0000'CF    A2  06CC         SUBW2   W^OUTLEN,W^OUTDES        ; Reduce size of remaining buffer
 0004'CF   0000'CF    C0  06D3         ADDL2   W^OUTLEN,W^OUTDES+4      ; Set pointer to first unused byte
                      05  06DA         RSB
```

```
                                06DB          .SBTTL  WRITE_LINE
                                06DB    ;
                                06DB    ; Subroutine to write line to SLPPR output file
                                06DB    ;
                                06DB    ; Inputs:
                                06DB    ;         R6 = Line buffer address
                                06DB    ;         R7 = Line size
                                06DB    ;
                                06DB    ;
                                06DB    ; Outputs:
                                06DB    ;         R0 = success/error status code
                                06DB    ;
                                06DB    ;
                                06DB    WRITE_LINE:
                          52  DD  06DB          PUSHL   R2
  0000'CF      0000'8F        B3  06DD          BITW    #MERM_OUTPUT,W^MERGE_FLAGS ; Was output file specified?
                          37  13  06E4          BEQL    10$                        ; No if EQL
          52   0000'CF        DE  06E6          MOVAL   W^OUTPUT_RAB,R2            ; Get address of RAB
             28 A2        56  D0  06EB          MOVL    R6,RAB$L_RBF(R2)           ; Insert line address
             22 A2        57  B0  06EF          MOVW    R7,RAB$W_RSZ(R2)           ; and size
                                06F3          $PUT    RAB = R2
                    1E 50     E8  06FC          BLBS    R0,10$                     ; OK if LBS
                    0C A2     DD  06FF          PUSHL   RAB$L_STV(R2)              ; Signal error
                          50  DD  0702          PUSHL   R0
                                0704          ERRMSG  OUTERR
   00000000'GF     02         FB  0716          CALLS   #2,G^LIB$SIGNAL
                                071D    10$:
                     52 8ED0      071D          POPL    R2
                          05      0720          RSB
```

```
                                   0721          .SBTTL  COMMAND_CHECK
                                   0721  ;
                                   0721  ; Subroutine to check if line is a command
                                   0721  ;
                                   0721  ; Inputs:
                                   0721  ;       R6 = Address of line
                                   0721  ;       R7 = Size of line
                                   0721  ;       R10= File address
                                   0721  ;
                                   0721  ; Outputs:
                                   0721  ;       R4[00] = 0:Data  1:Command
                                   0721  ;       R4[01] = 0:Normal command  1:Data terminator command
                                   0721  ;       R4[02] = 0:Data terminator  1:End of edit
                                   0721  ;       R6 = Address of line
                                   0721  ;       R7 = Size of line
                                   0721  ;
                                   0721          $TPADEF
                                   0721  ;
                                   0721  COMMAND_CHECK:
    0000'CF    0E AA         B0    0721          MOVW    SLP$W_DOT(R10),W^DOT     ; Get dot value for this file
    0008'CF    0C AA         9B    0727          MOVZBW  SLP$B_FLAGS(R10),W^FLAGS ; Get current flags word
    0008'CF    00'8F         8A    072D          BICB2   #SLPM_SUPPRESS,W^FLAGS   ; but clear suppress flag
               0002'CF       B4    0733          CLRW    W^LOC
               0004'CF       B4    0737          CLRW    W^LOC1
               0006'CF       B4    073B          CLRW    W^LOC2
               0016'CF       7C    073F          CLRQ    W^AUDIT
               001E'CF       7C    0743          CLRQ    W^COMMENT
       000E'CF    56         D0    0747          MOVL    R6,W^ADDR                ; Save line address and size
       0012'CF    57         D0    074C          MOVL    R7,W^SIZE
                                   0751  ;
       0032'CF    56         D0    0751          MOVL    R6,W^TPARSE_BLOCK+TPA$L_STRINGPTR ; Set TPARSE input
       002E'CF    57         D0    0756          MOVL    R7,W^TPARSE_BLOCK+TPA$L_STRINGCNT ; string descriptor
               0000'CF       DF    075B          PUSHAL  W^SUM_MER_KEY
               0000'CF       DF    075F          PUSHAL  W^SUM_MER_STATE
               0026'CF       DF    0763          PUSHAL  W^TPARSE_BLOCK
   00000000'GF    03         FB    0767          CALLS   #3,G^LIB$TPARSE
                 47 50       E9    076E          BLBC    R0,20$                   ; Error if LBC
         56    000E'CF       D0    0771          MOVL    W^ADDR,R6                ; Reset line address and size
         57    0012'CF       D0    0776          MOVL    W^SIZE,R7
      0E AA    0000'CF       B0    077B          MOVW    W^DOT,SLP$W_DOT(R10)
      0C AA    0008'CF       90    0781          MOVB    W^FLAGS,SLP$B_FLAGS(R10)
      08 AA    0004'CF       B0    0787          MOVW    W^LOC1,SLP$W_LOC1(R10)
      0A AA    0006'CF       B0    078D          MOVW    W^LOC2,SLP$W_LOC2(R10)
    0010 CA    0016'CF       B0    0793          MOVW    W^AUDIT,W^SLP$Q_AUDDS(R10)
               001A'CF       D5    079A          TSTL    W^AUDIT+4                ; Was audit trail specified?
                    0D       13    079E          BEQL    10$                      ; No if EQL
                    3F       BB    07A0          PUSHR   #^M<R0,R1,R2,R3,R4,R5>
14 BA  001A'DF  0016'CF      28    07A2          MOVC3   W^AUDIT,@W^AUDIT+4,@SLP$Q_AUDDS+4(R10)
                    3F       BA    07AB          POPR    #^M<R0,R1,R2,R3,R4,R5>
                                   07AD  10$:
      40 AA    001E'CF       7D    07AD          MOVQ    W^COMMENT,SLP$Q_CMNT(R10)
         54    000A'CF       D0    07B3          MOVL    W^TYPE,R4
                                   07B8  20$:
                             05    07B8          RSB
                                   07B9
                         00000000                .PSECT  $DATA,NOEXE,WRT
                                   0000  ;
```

```
                          0000     924 ; Local data
                          0000     925 ;
                    0000  0000     926 DOT:      .WORD   0                    ; '.' value
                    0000  0002     927 LOC:      .WORD   0
                    0000  0004     928 LOC1:     .WORD   0
                    0000  0006     929 LOC2:     .WORD   0
                    0000  0008     930 FLAGS:    .WORD   0
                00000000  000A     931 TYPE:     .LONG   0
                00000000  000E     932 ADDR:     .LONG   0
                00000000  0012     933 SIZE:     .LONG   0
       00000000 00000000  0016     934 AUDIT:    .QUAD   0
       00000000 00000000  001E     935 COMMENT:  .QUAD   0
```

```
                    0026         .SBTTL  TPARSE
                    0026 ;
                    0026 ;
                    0026 TPARSE_BLOCK:
          00000008  0026         .LONG   TPA$K_COUNT0
          000000AA  002A         .BLKL   TPA$K_LENGTH0-4
                    00AA ;
          0000002C  00AA         COMMA = ^X2C
          0000003B  00AA         SEMICOLON = ^X3B
                    00AA ;
                    00AA         .SAVE
                    00AA ;
                    00AA         $INIT_STATE     SUM_MER_STATE,SUM_MER_KEY
                    00AA ;
                    00AA ; Get 1st character of line
                    00AA ;
                    00AA         $STATE
                    00AA         $TRAN   TPA$_LAMBDA,,ACT_BLANKS_SIG
                    00AA         $STATE
                    00AA         $TRAN   '%',CMND,ACT_PERCENT
                    00AA         $TRAN   '-',EDIT
                    00AA         $TRAN   '/',TERM
                    00AA         $TRAN   '@',TPA$_FAIL,ACT_ATSIGN,,,0
                    00AA         $TRAN   '\',CMND,ACT_BACKSLASH
                    00AA         $TRAN   TPA$_EOS,DATA
                    00AA         $TRAN   TPA$_ANY,DATA
                    00AA ;
                    00AA ; End data line
                    00AA ;
                    00AA         $STATE  DATA
                    00AA         $TRAN   TPA$_LAMBDA,TPA$_EXIT,ACT_EXIT,,,0
                    00AA ;
                    00AA ; End normal command line
                    00AA ;
                    00AA         $STATE  CMND
                    00AA         $TRAN   TPA$_LAMBDA,TPA$_EXIT,ACT_EXIT,,,1
                    00AA ;
                    00AA ; End data terminating command
                    00AA ;
                    00AA         $STATE  TERM
                    00AA         $TRAN   TPA$_LAMBDA,TPA$_EXIT,ACT_EXIT,,,7
                    00AA ;
                    00AA ;
                    00AA ; Edit command
                    00AA ;
                    00AA ; Read locator-1
                    00AA ;
                    00AA         $STATE  EDIT
                    00AA         $TRAN   '-',,ACT_SUPPRESS
                    00AA         $TRAN   TPA$_LAMBDA
                    00AA         $STATE
                    00AA         $TRAN   TPA$_LAMBDA,,ACT_BLANKS_NSIG
                    00AA         $STATE
                    00AA         $TRAN   !LOCATOR,,ACT_LOC1
                    00AA ;
                    00AA ; Read Locator-2
                    00AA ;
```

```
    00AA            $STATE
    00AA            $TRAN   TPA$_EOS,TPA$_EXIT
    00AA            $TRAN   SEMICOLON,CMNT,ACT_CMNT
    00AA            $TRAN   COMMA
    00AA            $STATE
    00AA            $TRAN   !LOCATOR,,ACT_LOC2
    00AA            $TRAN   TPA$_EOS,TPA$_EXIT
    00AA       ;
    00AA       ; Read audit string
    00AA       ;
    00AA            $STATE
    00AA            $TRAN   TPA$_EOS,TPA$_EXIT
    00AA            $TRAN   SEMICOLON,CMNT,ACT_CMNT
    00AA            $TRAN   COMMA
    00AA            $STATE
    00AA            $TRAN   '/',,ACT_AUDIT
    00AA            $TRAN   TPA$_EOS,TPA$_EXIT
    00AA            $TRAN   SEMICOLON,CMNT,ACT_CMNT
    00AA            $STATE  AUDCH
    00AA            $TRAN   '/',,ACT_AUDEND
    00AA            $TRAN   TPA$_ANY,AUDCH,ACT_AUDCH
    00AA       ;
    00AA       ; Read comment line
    00AA       ;
    00AA            $STATE
    00AA            $TRAN   TPA$_EOS,TPA$_EXIT
    00AA            $TRAN   SEMICOLON,CMNT,ACT_CMNT
    00AA            $STATE  CMNT
    00AA            $TRAN   TPA$_LAMBDA,TPA$_EXIT
    00AA       ;
    00AA       ;
    00AA       ; Subexpression to parse locator
    00AA       ;
    00AA            $STATE  LOCATOR
    00AA            $TRAN   '.',,ACT_DOT
    00AA            $TRAN   TPA$_DECIMAL,,ACT_LOCNUM
    00AA            $TRAN   TPA$_LAMBDA,TPA$_EXIT
    00AA            $STATE
    00AA            $TRAN   '+'
    00AA            $TRAN   TPA$_LAMBDA,TPA$_EXIT
    00AA            $STATE
    00AA            $TRAN   TPA$_DECIMAL,,ACT_PLUS
    00AA            $STATE
    00AA            $TRAN   TPA$_LAMBDA,TPA$_EXIT
    00AA       ;
    00AA            $END_STATE
    00AA       ;
    00AA       ;
000000AA            .RESTORE
```

```
                            00AA  ;
                            00AA  ; Tparse action routines
                            00AA  ;
                            00AA  ;
                        000007B9          .PSECT  $CODE,NOWRT,EXE
                            07B9
                            07B9  ACT_BLANKS_SIG:
                     0000   07B9          .WORD   0
      00 04 AC    00   E2   07BB          BBSS    #TPA$V_BLANKS,TPA$L_OPTIONS(AP),10$
                            07C0  10$:
                       04   07C0          RET
                            07C1  ;
                            07C1  ;
                            07C1  ACT_BLANKS_NSIG:
                     0000   07C1          .WORD   0
      00 04 AC    00   E5   07C3          BBCC    #TPA$V_BLANKS,TPA$L_OPTIONS(AP),10$
                            07C8  10$:
                       04   07C8          RET
                            07C9  ;
                            07C9  ;
                            07C9  ACT_PERCENT:
                     0000   07C9          .WORD   0
  0008'CF    0000'8F   A8   07CB          BISW    #SLPM_AUDIT,W^FLAGS       ; Switch on audit trail
                       04   07D2          RET
                            07D3  ;
                            07D3  ;
                            07D3  ACT_ATSIGN:
                     0000   07D3          .WORD   0
                            07D5          ERRMSG  ATSIGN,<ADDR,SIZE>        ; Print error line
    000A'CF    20 AC   DE   07F3          MOVAL   TPA$L_PARAM(AP),W^TYPE
              50   01  D0   07F9          MOVL    #1,R0
                       04   07FC          RET
                            07FD  ;
                            07FD  ;
                            07FD  ACT_BACKSLASH:
                     0000   07FD          .WORD   0
  0008'CF    0000'8F   AA   07FF          BICW    #SLPM_AUDIT,W^FLAGS       ; Switch off audit trail
                       04   0806          RET
                            0807  ;
                            0807  ;
                            0807  ACT_EXIT:
                     0000   0807          .WORD   0
    000A'CF    20 AC   D0   0809          MOVL    TPA$L_PARAM(AP),W^TYPE            ; Set return type
                       04   080F          RET
                            0810  ;
                            0810  ;
                            0810  ACT_LOC1:
                     0000   0810          .WORD   0
       000A'CF    01   B0   0812          MOVW    #1,W^TYPE                 ; Assume normal command
  0004'CF    0002'CF   B0   0817          MOVW    W^LOC,W^LOC1
                  05   13   081E          BEQL    10$                       ; If EQL is a normal command
       000A'CF    03   B0   0820          MOVW    #3,W^TYPE                 ; Set as data terminator command
                            0825  10$:
          0002'CF      B4   0825          CLRW    W^LOC
                       04   0829          RET
                            082A  ;
                            082A  ;
```

```
                                  082A  ACT_LOC2:
                          0000    082A          .WORD   0
       0006'CF   0002'CF    B0    082C          MOVW    W^LOC,W^LOC2
                            04    0833          RET
                                  0834  ;
                                  0834  ;
                                  0834  ACT_DOT:
                          0000    0834          .WORD   0
       0002'CF   0000'CF    B0    0836          MOVW    W^DOT,W^LOC
                            04    083D          RET
                                  083E  ;
                                  083E  ;
                                  083E  ACT_LOCNUM:
                          0000    083E          .WORD   0
       0002'CF     1C AC    B0    0840          MOVW    TPA$L_NUMBER(AP),W^LOC
       0000'CF   0002'CF    B0    0846          MOVW    W^LOC,W^DOT
                            04    084D          RET
                                  084E  ;
                                  084E  ;
                                  084E  ACT_PLUS:
                          0000    084E          .WORD   0
       0002'CF     1C AC    A0    0850          ADDW2   TPA$L_NUMBER(AP),W^LOC
       0000'CF   0002'CF    B0    0856          MOVW    W^LOC,W^DOT
                            04    085D          RET
                                  085E
                                  085E  ;
                                  085E  ;
                                  085E  ACT_AUDIT:
                          0000    085E          .WORD   0
       001A'CF     0C AC    D0    0860          MOVL    TPA$L_STRINGPTR(AP),W^AUDIT+4
       00 04 AC       00    E2    0866          BBSS    #TPA$V_BLANKS,TPA$L_OPTIONS(AP),10$ ; Make blanks significant
                                  086B  10$:
                            04    086B          RET
                                  086C  ;
                                  086C  ;
                                  086C  ACT_AUDCH:
                          0000    086C          .WORD   0
                 0016'CF    D6    086E          INCL    W^AUDIT
                            04    0872          RET
                                  0873  ;
                                  0873  ;
                                  0873  ACT_AUDEND:
                          0000    0873          .WORD   0
       00 04 AC       00    E5    0875          BBCC    #TPA$V_BLANKS,TPA$L_OPTIONS(AP),10$ ; Switch off blank processing
                                  087A  10$:
                            04    087A          RET
                                  087B  ;
                                  087B  ;
                                  087B  ACT_CMNT:
                          0000    087B          .WORD   0
       001E'CF     08 AC    7D    087D          MOVQ    TPA$L_STRINGCNT(AP),W^COMMENT
                            04    0883          RET
                                  0884  ;
                                  0884  ;
                                  0884  ACT_SUPPRESS::
                          0000    0884          .WORD   0
       0008'CF     00'8F    88    0886          BISB2   #SLPM_SUPPRESS,W^FLAGS  ; set clash messages suppressed flag
```

```
04  088C  1158          RET
```

```
088D  1160 ;
088D  1161 ;
088D  1162          .END
```

```
$$                              = 00000002
$$$CNT                          = 00000003
$$$FLG                          = FFFFFFFF
$$$KEY                          = FFFFFFFF
$$$KFG                          = FFFFFFFF
$$$MOD                          = 00000000
$$.TMP1                         = 00000001
$$.TMP2                         = 00000052
$$KEYTAB                        = 00000000 R        05
$$T2                            = 00000004
..AFLG                          = 00000000
..FLG                           = 00000002
..TYP                           = 00000054
.LEN                            = 00000004
ACCESS_EDIT                       00000508 R        02
ACT_ATSIGN                        000007D3 R        02
ACT_AUDCH                         0000086C R        02
ACT_AUDEND                        00000873 R        02
ACT_AUDIT                         0000085E R        02
ACT_BACKSLASH                     000007FD R        02
ACT_BLANKS_NSIG                   000007C1 R        02
ACT_BLANKS_SIG                    000007B9 R        02
ACT_CMNT                          0000087B R        02
ACT_DOT                           00000834 R        02
ACT_EXIT                          00000807 R        02
ACT_LOC1                          00000810 R        02
ACT_LOC2                          0000082A R        02
ACT_LOCNUM                        0000083E R        02
ACT_PERCENT                       000007C9 R        02
ACT_PLUS                          0000084E R        02
ACT_SUPPRESS                      00000884 RG       02
ADDR                              0000000E R        03
ADJUST_COUNTS                     000006C7 R        02
AUDCH                             000000AA R        04
AUDIT                             00000016 R        03
AUDIT_OFF                         ********      X   02
AUDIT_ON                          ********      X   02
AUDIT_TRAIL                       ********      X   02
BIT...                          = 00000004
BUF_SIZE                        = 00000200
CLASH_LINE                        0000023C R        02
CMD_SIZE                        = 00000084
CMND                              00000040 R        04
CMNT                              000000C4 R        04
COMMA                           = 0000002C
COMMAND_CHECK                     00000721 R        02
COMMENT                           0000001E R        03
COMMENTS                          ********      X   02
DATA                              00000033 R        04
DOT                               00000000 R        03
ED$B_FILENO                     = 00000019
ED$B_FLAGS                      = 00000018
ED$K_BLN                        = 0000001A
ED$L_BWD                        = 00000004
ED$L_FILE                       = 00000014
ED$L_FWD                        = 00000000
ED$W_LINES                      = 0000000C
ED$W_LOC1                       = 00000008
ED$W_LOC2                       = 0000000A
ED$W_RFA                        = 0000000E
EDIT                              0000005A R        04
EDIT_NODES                        ********      X   02
ED_SIZE                           ********      X   02
END_EDIT                          ********      X   02
FAB$L_NAM                       = 00000028
FAB$L_STV                       = 0000000C
FILE_NODES                        ********      X   02
FLAGS                             00000008 R        03
GET_NODE                          0000017B R        02
INPUT_FAB                         ********      X   02
INPUT_RAB                         ********      X   02
INSERT_NODE                       000001A8 R        02
LIB$FREE_VM                       ********      X   02
LIB$GET_VM                        ********      X   02
LIB$SIGNAL                        ********      X   02
LIB$TPARSE                        ********      X   02
LOC                               00000002 R        03
LOC1                              00000004 R        03
LOC2                              00000006 R        03
LOCATOR                           000000C8 R        04
LOC_ONE                           ********      X   02
LOC_TWO                           ********      X   02
MER$_ATSIGN                       ********      X   02
MER$_CLASH                        ********      X   02
MER$_CLOSER                       ********      X   02
MER$_CLSHLN                       ********      X   02
MER$_CONNEC                       ********      X   02
MER$_DISCON                       ********      X   02
MER$_INPEOF                       ********      X   02
MER$_OUTERR                       ********      X   02
MER$_READER                       ********      X   02
MER$_REOPEN                       ********      X   02
MER$_SYNTAX                       ********      X   02
MERGE_FILES                       00000000 RG       02
MERGE_FLAGS                       ********      X   02
MERM_OUTPUT                       ********      X   02
NAM$B_RSL                       = 00000003
NAM$K_BLN                       = 00000060
NAM$L_RSA                       = 00000004
OUTDES                            ********      X   02
OUTLEN                            ********      X   02
OUTPUT_BUF                        ********      X   02
OUTPUT_RAB                        ********      X   02
PASS_ONE                          0000000B R        02
PASS_TWO                          000003F9 R        02
PROCESS_EDIT                      00000421 R        02
RAB$B_RAC                       = 0000001E
RAB$C_RFA                       = 00000002
RAB$C_SEQ                       = 00000000
RAB$L_RBF                       = 00000028
RAB$L_STV                       = 0000000C
RAB$W_RFA                       = 00000010
RAB$W_RSZ                       = 00000022
RANDOM_FAB                        ********      X   02
```

```
RANDOM_FILE                     ********     X    02
RANDOM_OPEN                     000002AD R        02
RANDOM_RAB                      ********     X    02
RANGE_EDIT                      000004BC R        02
READ_LINE                       000003A4 R        02
READ_RECORD                     00000527 R        02
REPORT_CLASH                    000001F6 R        02
RESET_AUDIT                     00000568 R        02
RESET_CURRENT                   00000578 R        02
RETURN_NODE                     00000193 R        02
RMS$_EOF                        ********     X    02
SEMICOLON                     = 0000003B
SET_UP_NODES                    000000EF R        02
SIZ...                        = 00000001
SIZE                            00000012 R        03
SLP$B_FILENO                  = 0000000D
SLP$B_FLAGS                   = 0000000C
SLP$K_BLN                     = 000000A8
SLP$L_BWD                     = 00000004
SLP$L_FWD                     = 00000000
SLP$Q_AUCDS                   = 00000028
SLP$Q_AUDDS                   = 00000010
SLP$Q_CMNT                    = 00000040
SLP$T_AUCST                   = 00000030
SLP$T_AUDST                   = 00000018
SLP$T_NAM                     = 00000048
SLP$W_DOT                     = 0000000E
SLP$W_LOC1                    = 00000008
SLP$W_LOC2                    = 0000000A
SLPM_AUDIT                      ********     X    02
SLPM_SUPPRESS                   ********     X    02
SLPV_SUPPRESS                   ********     X    02
SLP_AUDDS                       ********     X    02
SLP_AUDST                       ********     X    02
SLP_FLAGS                       ********     X    02
SUM_MER_KEY                     00000000 RG       05
SUM_MER_STATE                   00000000 RG       04
SUM_M_1STEDIT                 = 00000002
SUM_M_EXPED                   = 00000001
SUM_M_SPRCDE                  = 00000008
SUM_M_SUPPRESS                = 00000004
SUM_V_1STEDIT                 = 00000001
SUM_V_EXPED                   = 00000000
SUM_V_SPRCDE                  = 00000003
SUM_V_SUPPRESS                = 00000002
SYS$CLOSE                       ********    GX    02
SYS$CONNECT                     ********    GX    02
SYS$DISCONNECT                  ********    GX    02
SYS$FAO                         ********     X    02
SYS$GET                         ********    GX    02
SYS$OPEN                        ********    GX    02
SYS$PUT                         ********    GX    02
TERM                            0000004D R        04
TPA$K_COUNT0                  = 00000008
TPA$K_LENGTH0                 = 00000024
TPA$L_NUMBER                  = 0000001C
TPA$L_OPTIONS                 = 00000004
TPA$L_PARAM                   = 00000020
TPA$L_STRINGCNT               = 00000008
TPA$L_STRINGPTR               = 0000000C
TPA$V_BLANKS                  = 00000000
TPA$_ALPHA                    = 000001EE
TPA$_ANY                      = 000001ED
TPA$_BLANK                    = 000001F2
TPA$_DECIMAL                  = 000001F3
TPA$_DIGIT                    = 000001EF
TPA$_EOS                      = 000001F7
TPA$_EXIT                     = FFFFFFFF
TPA$_FAIL                     = FFFFFFFE
TPA$_FILESPEC                 = 000001EA
TPA$_HEX                      = 000001F5
TPA$_IDENT                    = 000001EC
TPA$_KEYWORD                  = 00000100
TPA$_LAMBDA                   = 000001F6
TPA$_MAXKEY                   = 000000DC
TPA$_OCTAL                    = 000001F4
TPA$_STRING                   = 000001F0
TPA$_SUBXPR                   = 000001F8
TPA$_SYMBOL                   = 000001F1
TPA$_UIC                      = 000001EB
TPARSE_BLOCK                    00000026 R        03
TYPE                            0000000A R        03
VIRT_ADDR                       ********     X    02
WRITE_EDIT                      0000058D R        02
WRITE_LINE                      000006DB R        02
```

```
                                +----------------+
                                ! Psect synopsis !
                                +----------------+

PSECT name                      Allocation           PSECT No.  Attributes
----------                      ----------           ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                           00000000  (     0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$CODE                           0000088D  (  2189.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE
$DATA                           000000AA  (   170.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD     WRT NOVEC BYTE
_LIB$STATE$                     000000E8  (   232.)  04 (  4.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC BYTE
_LIB$KEY0$                      00000000  (     0.)  05 (  5.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC WORD
_LIB$KEY1$                      00000000  (     0.)  06 (  6.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC WORD

                             +------------------------+
                             ! Performance indicators !
                             +------------------------+

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                    37    00:00:00.08    00:00:00.36
Command processing               132    00:00:00.71    00:00:04.69
Pass 1                           441    00:00:21.50    00:00:49.99
Symbol table sort                  5    00:00:01.22    00:00:03.38
Pass 2                           212    00:00:05.22    00:00:11.81
Symbol table output               24    00:00:00.18    00:00:00.20
Psect synopsis output              3    00:00:00.03    00:00:00.03
Cross-reference output             0    00:00:00.00    00:00:00.00
Assembler run totals             856    00:00:28.95    00:01:10.46

The working set limit was 1800 pages.
114014 bytes (223 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 847 non-local and 86 local symbols.
1247 source lines were read in Pass 1, producing 33 object records in Pass 2.
54 pages of virtual memory were used to define 40 macros.

                            +--------------------------+
                            ! Macro library statistics !
                            +--------------------------+

Macro library name                         Macros defined
------------------                         --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                         0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     28
TOTALS (all libraries)                                28

1243 GETS were required to define 28 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SUMED/OBJ=OBJ$:SUMED MSRC$:SUMCOM/UPDATE=(ENH$:SUMCOM)+MSRC$:SUMED/UPDATE=(ENH$:SUMED)+EXECML$/LIB
```

0368 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SUMSLP
MAP

SRTTRNMSG
LIS

DATA
LIS

SUMSHR
MAP

SUMDATA
LIS

UTLCHKCLI
LIS

SUM

SUM
MAP

SUMDEF
MAP

MAIN
LIS

SUMED
LIS

SUMEDIT
LIS

GETCMD
LIS