F 5

```
   SSSSSSSSSSSS  DDDDDDDDDDDD         AAAAAAAAA
   SSSSSSSSSSSS  DDDDDDDDDDDD         AAAAAAAAA
   SSSSSSSSSSSS  DDDDDDDDDDDD         AAAAAAAAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
   SSSSSSSSS     DDD         DDD  AAA         AAA
   SSSSSSSSS     DDD         DDD  AAA         AAA
   SSSSSSSSS     DDD         DDD  AAA         AAA
            SSS  DDD         DDD  AAAAAAAAAAAAAAA
            SSS  DDD         DDD  AAAAAAAAAAAAAAA
            SSS  DDD         DDD  AAAAAAAAAAAAAAA
            SSS  DDD         DDD  AAA         AAA
            SSS  DDD         DDD  AAA         AAA
            SSS  DDD         DDD  AAA         AAA
SSSSSSSSSSSS     DDDDDDDDDDDD     AAA         AAA
SSSSSSSSSSSS     DDDDDDDDDDDD     AAA         AAA
SSSSSSSSSSSS     DDDDDDDDDDDD     AAA         AAA
```

```
**FILE**ID**VALIDATE                                   I 8
```

```
VV      VV    AAAAAA    LL            IIIIII    DDDDDDDD      AAAAAA    TTTTTTTTTT  EEEEEEEEEE
VV      VV    AAAAAA    LL            IIIIII    DDDDDDDD      AAAAAA    TTTTTTTTTT  EEEEEEEEEE
VV      VV  AA      AA  LL              II      DD      DD  AA      AA      TT      EE
VV      VV  AA      AA  LL              II      DD      DD  AA      AA      TT      EE
VV      VV  AA      AA  LL              II      DD      DD  AA      AA      TT      EE
VV      VV  AA      AA  LL              II      DD      DD  AA      AA      TT      EE
VV      VV  AA      AA  LL              II      DD      DD  AA      AA      TT      EEEEEEEE
VV      VV  AA      AA  LL              II      DD      DD  AA      AA      TT      EEEEEEEE
VV      VV  AAAAAAAAAA  LL              II      DD      DD  AAAAAAAAAA      TT      EE
VV      VV  AAAAAAAAAA  LL              II      DD      DD  AAAAAAAAAA      TT      EE
  VV  VV    AA      AA  LL              II      DD      DD  AA      AA      TT      EE              ....
  VV  VV    AA      AA  LL              II      DD      DD  AA      AA      TT      EE              ....
    VV      AA      AA  LLLLLLLLLL    IIIIII    DDDDDDDD    AA      AA      TT      EEEEEEEEEE      ....
    VV      AA      AA  LLLLLLLLLL    IIIIII    DDDDDDDD    AA      AA      TT      EEEEEEEEEE      ....

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

Table of contents

```
0000            .TITLE  VALIDATE - Struc.ure Validation module
0000            .IDENT  '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      ;++
0000      ;
0000      ; FACILITY:
0000      ;
0000      ;       SDA
0000      ;
0000      ; ABSTRACT:
0000      ;
0000      ;       This module contains code which verifies the consistency of
0000      ;       certain VMS internal structues.
0000      ;
000C      ; ENVIRONMENT:
0000      ;
000C      ;       VMS user mode.
0000      ;
0000      ;--
0000      ;
0000      ; AUTHOR: Jake VanNoy, CREATION DATE: 21-Jan-1983
0000      ;
0000      ; MODIFIED BY:
0000      ;
0000      ;       V03-000 JLV0226         Jake VanNoy             21-JAN-1983
0000      ;               Initial coding.
0000      ;**
0000
0000            .SBTTL  DECLARATIONS
0000      ;
0000      ; INCLUDE FILES:
C000      ;
0000
```

```
                0000            $OPTDEF
                0000            $TPADEF
                0000
                0000     ;
                0000     ; MACROS:
                0000     ;
                0000
                0000     ;
                0000     ; EQUATED SYMBOLS:
                0000     ;
      00000000  0000     NEWFL = 0
      00000004  0000     NEWBL = 4
      00000008  0000     HEADFL = 8
      0000000C  0000     HEADBL = 12
      00000010  0000     LOCAL_STORAGE = 16
                0000
                0000     ;
                0000     ; OWN STORAGE:
                0000     ;
                0000
      00000190  0000     VALIDATE_MAX:   .LONG   400
                0004
                0004
                0004            .SBTTL  VAL_SET_MAX - Set maximum number of links to traverse
                0004
                0004     ;++
                0004     ;
                0004     ; FUNCTIONAL DESCRIPTION:
                0004     ;
                0004     ;       Tparse action routine which sets value of VALIDATE_MAX.
                0004     ;       *** Note that the VALIDATE_QUEUE code does not use this
                0004     ;       *** as input. It has not been proven that there are ''infinite''
                0004     ;       *** queues which the algorithm does not detect.
                0004     ;
                0004     ; CALLING SEQUENCE:
                0004     ;       NONE
                0004     ;
                0004     ; INPUT PARAMETERS:
                0004     ;       NONE
                0004     ;
                0004     ; IMPLICIT INPUTS:
                0004     ;       NONE
                0004     ;
                0004     ; OUTPUT PARAMETERS:
                0004     ;       NONE
                0004     ;
                0004     ; IMPLICIT OUTPUTS:
                0004     ;       NONE
                0004     ;
                0004     ; COMPLETION CODES:
                0004     ;       NONE
                0004     ;
                0004     ; SIDE EFFECTS:
                0004     ;       NONE
                0004     ;
                0004     ;--
                0004
```

```
                         0004
                   0000  0004   .Entry  VAL_SET_MAX, 0
                         0006
F5 AF    1C AC     D0    0006           MOVL    TPA$L_NUMBER(AP),VALIDATE_MAX ; set max
     50    01      D0    000B           MOVL    #1,R0
                   04    000E           RET
```

```
                              000F          .SBTTL  VALIDATE_QUEUE - Validate queue structure
                              000F
                              000F    ;++
                              000F    ;
                              000F    ; FUNCTIONAL DESCRIPTION:
                              000F    ;
                              000F    ;       The algorithm used to validate a doubly linked list queue
                              000F    ;       starts by copying the head (or starting point specified)
                              000F    ;       into a known location. It then proceeds by following the
                              000F    ;       forward link checking that the backward link points to where
                              000F    ;       the last forward link was.
                              000F    ;
                              000F    ; CALLING SEQUENCE:
                              000F    ;       CALLS from tparse.
                              000F    ;
                              000F    ; INPUT PARAMETERS:
                              000F    ;       TPA$L_NUMBER(AP) - address to start search from
                              000F    ;
                              000F    ; IMPLICIT INPUTS:
                              000F    ;
                              000F    ;       OPTIONS - can specify SELF RELATIVE QUEUE
                              000F    ;       *** code not writeen for this as yet.
                              000F    ;
                              000F    ; OUTPUT PARAMETERS:
                              000F    ;       NONE
                              000F    ;
                              000F    ; IMPLICIT OUTPUTS:
                              000F    ;       NONE
                              000F    ;
                              000F    ; COMPLETION CODES:
                              000F    ;
                              000F    ;       no such memory, or success
                              000F    ;
                              000F    ; SIDE EFFECTS:
                              000F    ;       NONE
                              000F    ;
                              000F    ;--
                              000F
                              000F
                              000F
                        00FC  000F    .Entry  VALIDATE_QUEUE, ^M<R2,R3,R4,R5,R6,R7>
                              0011
        50    1C AC     D0    0011            MOVL    TPA$L_NUMBER(AP),R0     ; Address to start at
 00000000'EF    50      D0    0015            MOVL    R0,ADDRESS              ; Set "current"
                              001C
                              001C    ;       MOVL    VALIDATE_MAX,R1         ; Max number of links *** not used
                              001C
           5E   10      C2    001C            SUBL2   #local_storage,SP       ; allocate storage from stack
           52   5E      D0    001F            MOVL    SP,R2                   ; Allocate or stack
                56      D4    0022            CLRL    R6                      ; Counter
                              0024
           53   50      D0    0024            MOVL    R0,R3                   ; init last pointer
                              0027            TRYMEM  (R0),HEADFL(R2),#8      ; try memory at head of queue
        54    08 A2     D0    0035            MOVL    HEADFL(R2),R4           ; next address
                              0039            ;
                              0039            ; Loop through flinks
                              0039            ;
```

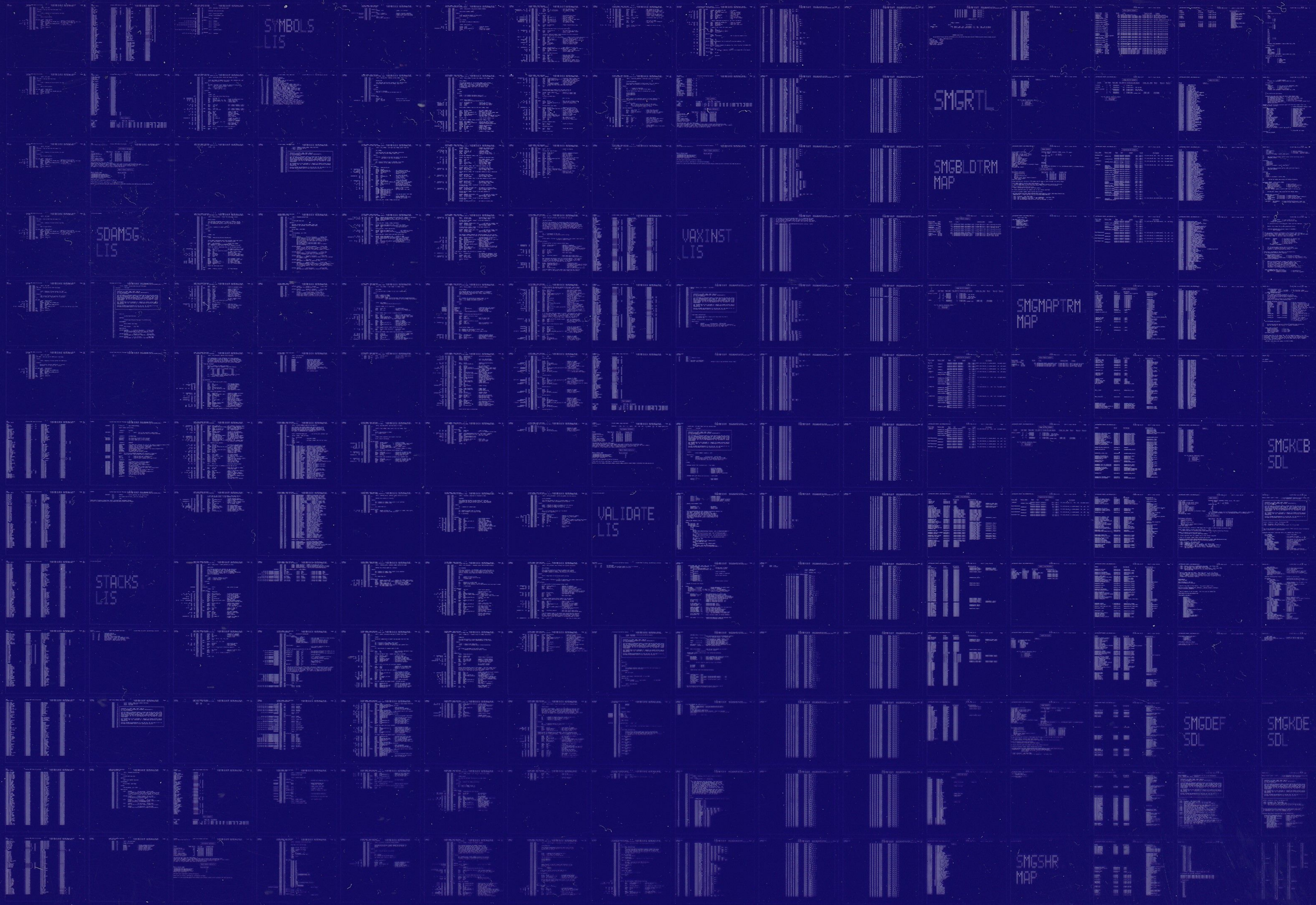
```
                          0039      20$:
                          0039                  TRYMEM  (R4),NEWFL(R2),#8         ; try memory
              37 50   E9  0046                  BLBC    R0,mem_err                ; Error
        08 A2    62   D1  0049                  CMPL    newfl(R2),headfl(R2)      ; Same as listhead?
                 10   13  004D                  BEQL    100$                      ; Done with flinks
                 56   D6  004F                  INCL    R6                        ; Increment counter
        04 A2    53   D1  0051                  CMPL    R3,NEWBL(R2)              ; back link ok?
                 3E   12  0055                  BNEQ    bad_blink                 ; branch if not
           53    54   D0  0057                  MOVL    R4,R3                     ; save last pointer
           54    62   D0  005A                  MOVL    NEWFL(R2),R4              ; move to next element
                 DA   11  005D                  BRB     20$                       ; Loop
                          005F                  ;
                          005F                  ; Search completed successfully, do final validation
                          005F                  ;
  0C A2     04 A2     D1  005F      100$:       CMPL    NEWBL(R2),HEADBL(R2)      ; Same as listhead?
                 2F   12  0064                  BNEQ    bad_blink                 ; Done with list
        0C A2    53   D1  0066                  CMPL    R3,HEADBL(R2)             ; does this check out?
                 29   12  006A                  BNEQ    bad_blink
                          006C                  ;
                          006C                  ; Queue is ok, check for empty queue
                          006C                  ;
                 56   D5  006C      110$:       TSTL    R6
                 47   12  006E                  BNEQ    queue_ok
                          0070                  PRINT   0,<The queue is empty>  ;
            0046      31  007D                  BRW     VAL_Q_EXIT
                          0080
                          0080      mem_err:
                 54   DD  0080                  PUSHL   R4
                          0082                  SIGNAL  1,NOTINPHYS               ; Not in physical memory error
                      04  0094                  RET
                          0095
                          0095      bad_blink:
                 53   DD  0095                  PUSHL   R3
                          0097                  PRINT   1,-
                          0097      <Error comparing backward link to previous structure address (!XL)>
                 56   DD  00A4                  PUSHL   R6
                 54   DD  00A6                  PUSHL   R4
                          00A8                  PRINT   2,-
                          00A8      <Error occured in queue element at address !XL, after tracing !UL element!%S>
                 0F   11  00B5                  brb     val_q_exit
                          00B7
                          00B7      queue_ok:
                 56   DD  00B7                  PUSHL   R6                        ; Count
                          00B9                  PRINT   1, <Queue is complete, total of !UL element!%S in the queue>
                          00C6
                          00C6      val_q_exit:
           50    01   D0  00C6                  MOVL    #1,R0
                      04  00C9                  RET
                          00CA
                          00CA      .END        ;       VALIDATE
```

```
ADDRESS              ********    X    01
ARGS               = 00000003
BAD_BLINK            00000095 R       01
HEADBL             = 0000000C
HEADFL             = 00000008
LIB$SIGNAL           ********    X    01
LOCAL_STORAGE      = 00000010
MEM_ERR              00000080 R       01
MSG$_NOTINPHYS       ********    X    01
NEWBL              = 00000004
NEWFL              = 00000000
PRINT                ********    X    01
QUEUE_OK             000000B7 R       01
TPA$L_NUMBER       = 0000001C
TRYMEM               ********    X    01
VALIDATE_MAX         00000000 R       01
VALIDATE_QUEUE       0000000F RG      01
VAL_Q_EXIT           000000C6 R       01
VAL_SET_MAX          00000004 RG      01


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

PSECT name                          Allocation          PSECT No.  Attributes
----------                          ----------          ---------  ----------
.  ABS  .                           00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
. BLANK .                           000000CA  (  202.)  01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$ABS$                               00000000  (    0.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
LITERALS                            000000F5  (  245.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD  NOWRT NOVEC BYTE

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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                       29    00:00:00.05    00:00:00.80
Command processing                  111    00:00:00.48    00:00:02.85
Pass 1                              163    00:00:01.68    00:00:06.93
Symbol table sort                     0    00:00:00.12    00:00:00.12
Pass 2                               53    00:00:00.47    00:00:02.15
Symbol table output                   3    00:00:00.02    00:00:00.02
Psect synopsis output                 2    00:00:00.02    00:00:00.02
Cross-reference output                0    00:00:00.00    00:00:00.00
Assembler run totals                363    00:00:02.84    00:00:12.89

The working set limit was 1050 pages.
13151 bytes (26 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 182 non-local and 8 local symbols.
228 source lines were read in Pass 1, producing 21 object records in Pass 2.
13 pages of virtual memory were used to define 12 macros.
```

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

Macro library name                            Macros defined
------------------                            --------------
_$255$DUA28:[SDA.OBJ]SDALIB.MLB;1                   5
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                      0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                   4
TOTALS (all libraries)                              9

275 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

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

0354 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



SYMBOLS
LIS

SMGRTL

SMGBLDTRM
MAP

SDAMSG
LIS

VAXINST
LIS

SMGMAPTRM
MAP

SMGKCB
SDL

VALIDATE
LIS

STACKS
LIS

SMGDEF
SDL

SMGKDE
SDL

SMGSHR
MAP