F 1

```
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG   GGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG   GGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG   GGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
```

```
**FILE**ID**RESETSSI                                  M 16
```

```
RRRRRRRR    EEEEEEEEEE    SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS    SSSSSSSS    IIIIII
RRRRRRRR    EEEEEEEEEE    SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS    SSSSSSSS    IIIIII
RR      RR  EE          SS          EE              TT      SS          SS              II
RR      RR  EE          SS          EE              TT      SS          SS              II
RR      RR  EE          SS          EE              TT      SS          SS              II
RR      RR  EE          SS          EE              TT      SS          SS              II
RRRRRRRR    EEEEEEEE      SSSSSS    EEEEEEEE        TT        SSSSSS      SSSSSS        II
RRRRRRRR    EEEEEEEE      SSSSSS    EEEEEEEE        TT        SSSSSS      SSSSSS        II
RR  RR      EE                  SS  EE              TT              SS          SS      II
RR  RR      EE                  SS  EE              TT              SS          SS      II
RR    RR    EE                  SS  EE              TT              SS          SS      II          ....
RR    RR    EE                  SS  EE              TT              SS          SS      II          ....
RR      RR  EEEEEEEEEE  SSSSSSSS    EEEEEEEEEE      TT      SSSSSSSS    SSSSSSSS    IIIIII      ....
RR      RR  EEEEEEEEEE  SSSSSSSS    EEEEEEEEEE      TT      SSSSSSSS    SSSSSSSS    IIIIII      ....

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
```

1

Table of contents

R
V

```
0000                   .TITLE  RESETSSV -- Reset System Service Vector
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
0000           ;++
0000           ; Facility: VAX/VMS System Service Monitor
0000           ;
0000           ; Abstract:
0000           ;
0000           ;       This routine puts the systerm service vector back in shape and
0000           ;       deletes the P0 space.
0000           ;       (This routine is called when SYS$RUNDWN is encountered and
0000           ;        System service vector has been modified.)
0000           ;
0000           ; Environment:  VAX/VMS
0000           ;
0000           ;--
0000           ;
0000           ; Author: D.W. Thiel,    Creation Date: 30-Dec-1981
0000           ;
0000           ; Modified By:
0000           ;
0000           ;         P. Sager,      20-Sep-1983
0000           ;
0000           ;**
0000           ;
0000
0000
0000                   .SBTTL  DECLARATIONS
0000
0000                   .ENABLE SUPPRESSION
0000
0000                   .EXTERNAL INTERCEPT,RANGE,SSV_MUNGED_FLAG
```

```
    0000
    0000      ;
    0000      ; EQUATED SYMBOLS:
    0000      ;
    0000
    0000              $SGNDEF
    0000              $IPLDEF
    0000
    0000      ;
    0000      ; OWN STORAGE:
    0000      ;
00000000              .PSECT  $RESETDATA$, NOEXE, LONG, PIC
    0000
00000000              .PSECT  $RESETCODE$, NOWRT, BYTE, PIC, EXE
```

```
0000            .SBTTL  RESET_SSV
0000
0000    ;++
0000    ;
0000    ; FUNCTIONAL DESCRIPTION:
0000    ;       This routine is called at image run down time to copy the
0000    ;       original system service vector back and get rid of the P0.
0000    ; CALLING SEQUENCE:
0000    ;       This routine is called from SSI_USS by passing SYS$RUNDWN in as
0000    ;       second parameter.  All the other parameters are ignored.
0000    ;--
0000
```

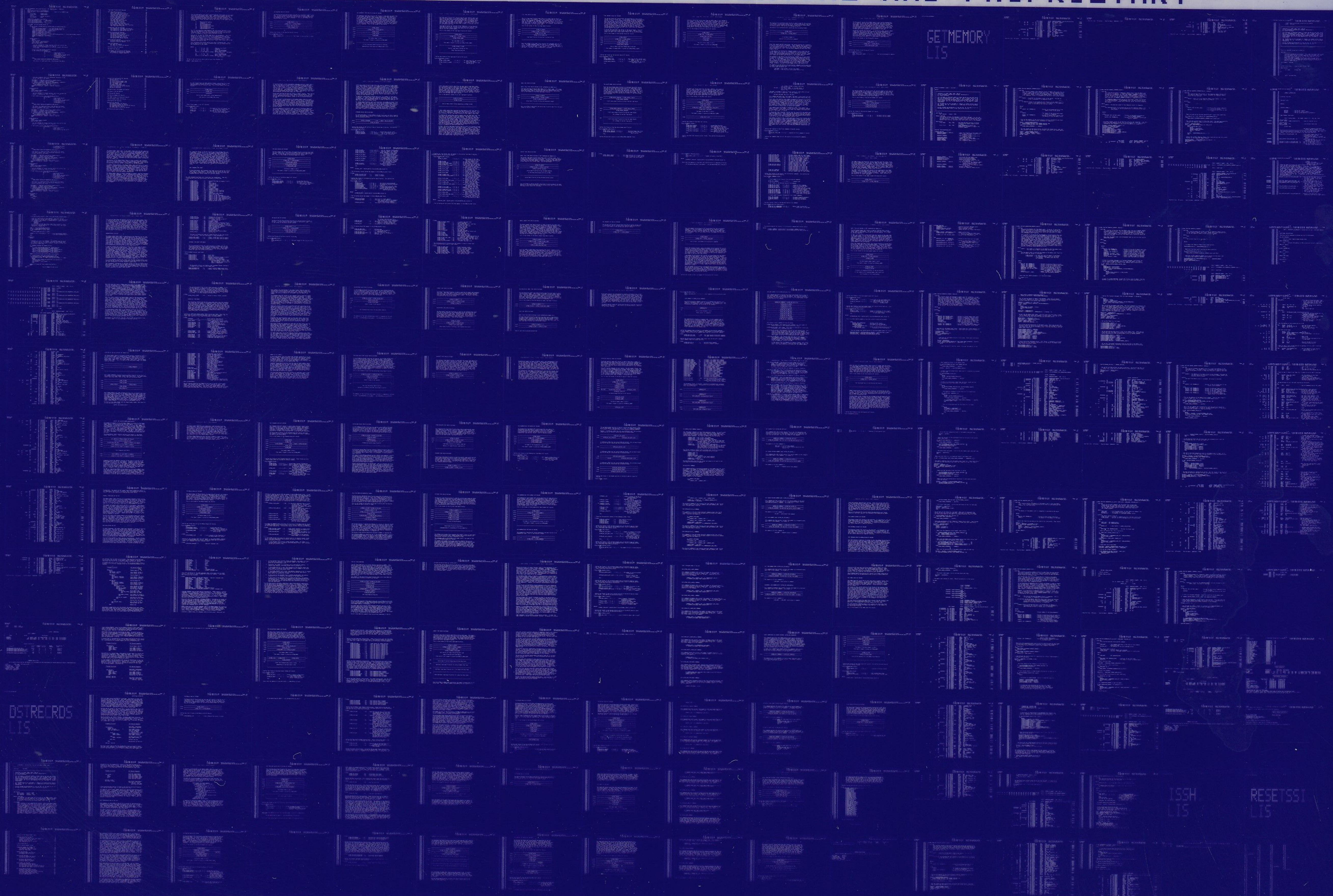
```
                         0010  0000          .ENTRY  RESET_SSV,^M<R4>
                               0002
         51  00000000'EF   D0  0002          MOVL    INTERCEPT,R1            ; Pointer to P0 saved copy
                               0009
                               0009   ; If this module is part of the installed privileged shareable
                               0009   ; image then $CHMKRL is not needed in here to copy the vector back.
                               0009   ; For in user defined system service vector already has CHMK
                               0009   ; instruction which grands the privilege.
                               0009   ;
                               0009   ; In testing stage, it is easier if we reduce the amount of code in kernel
                               0009   ; mode, strip off the interface module (SSIDISP.MAR which defines the user
                               0009   ; system service in KERNEL mode), call the essential routines via $CHMKRL,
                               0009   ; so that we still could test the remaining code in lesser mode environment.
                               0009   ; After testing is done, then we put interface module in place to execute
                               0009   ; the whole image in the proper mode environment.
                               0009   ;
                               0009   ; I have strip off all the other $CHMKRL in the other parts of this image
                               0009   ; except this one.  For this one has a different call interface, see
                               0009   ; the comment below, otherwise it would not work.  So I kept it here
                               0009   ; to save the need of finding out how to call $CHMKRL in here if there is a
                               0009   ; need to set up a test environment, since it does not hurt to keep it.
                               0009   ; -Ping
                               0009   ;
                      00   DD  0009          PUSHL   #0                      ; Just about to put original one back
                   14'AF   9F  000B          PUSHAB  B^COPY_SAVED_SSV        ; Routine to be executed in Kernel
                               000E                                          ;  Mode
                      02   FB  000E          CALLS   #2,-                    ; Call $CHMKRL through saved vector in
                 0000'C1       0010          W^<SYS$CMKRNL-SYS$QIOW>(R1); P0 (must be this way, for the
                               0013                                          ;    state is going to change after RET)
                           04  0013          RET                             ; Now, everything is back to normal
                               0014
                               0014   COPY_SAVED_SSV::
                         0000  0014          .WORD   ^M<>
         51  00000000'EF   D0  0016          MOVL    INTERCEPT,R1            ; Pointer to P0 saved copy
                      3C   BB  001D          PUSHR   #^M<R2,R3,R4,R5>        ; save volatile registers
                               001F          DSBINT  #IPL$_ASTDEL            ; disable AST's
      61    0A00 8F        28  0025          MOVC3   #SGN$C_SYSVECPGS*512, - ; length of transfer vector
         00000000'9F           002A                  (R1),                 - ; saved transfer vector address
                               002F                  @#SYS$QIOW              ; Put system service vector back
         00000000'EF       D4  002F          CLRL    SSV_MUNGED_FLAG         ; Clear flag which says whether
                               0035                                          ;    the system service vector has
                               0035                                          ;    been modified by us
                               0035          ENBINT                          ; restore IPL
                      3C   BA  0038          POPR    #^M<R2,R3,R4,R5>        ; restore volatile registers
              50      01   D0  003A          MOVL    #1,R0
                           04  003D          RET
                               003E
                               003E          .END
```

```
COPY_SAVED_SSV   00000014 RG     03
INTERCEPT        ********    X   00
IPL$_ASTDEL    = 00000002
PR$_IPL          ********    X   03
RANGE            ********    X   00
RESET_SSV        00000000 RG     03
SGN$C_SYSVECPGS= 00000005
SSV_MUNGED_FLAG  ********    X   00
SYS$CMKRNL       ********    X   03
SYS$QIOW         ********    X   03


                                        +----------------+
                                        ! 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
$RESETDATA$                00000000  (     0.)   02 (  2.)    PIC    USR    CON    REL    LCL NOSHR NOEXE   RD     WRT NOVEC LONG
$RESETCODE$                0000003E  (    62.)   03 (  3.)    PIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   12    00:00:00.08    00:00:01.06
Command processing               83    00:00:00.73    00:00:03.41
Pass 1                          110    00:00:01.13    00:00:03.21
Symbol table sort                 0    00:00:00.03    00:00:00.02
Pass 2                           41    00:00:00.36    00:00:00.80
Symbol table output               2    00:00:00.02    00:00:00.02
Psect synopsis output             2    00:00:00.02    00:00:00.24
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            252    00:00:02.38    00:00:08.78

The working set limit was 1050 pages.
3836 bytes (8 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 38 non-local and 0 local symbols.
133 source lines were read in Pass 1, producing 18 object records in Pass 2.
11 pages of virtual memory were used to define 10 macros.

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

Macro library name                  Macros defined
------------------                  --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                   4
_$255$DUA28:[SYSLIB]STARLET.MLB;2                3
TOTALS (all libraries)                           7

112 GETS were required to define 7 macros.

There were no errors, warnings or information messages.
```

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

0097 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



GETMEMORY
LIS

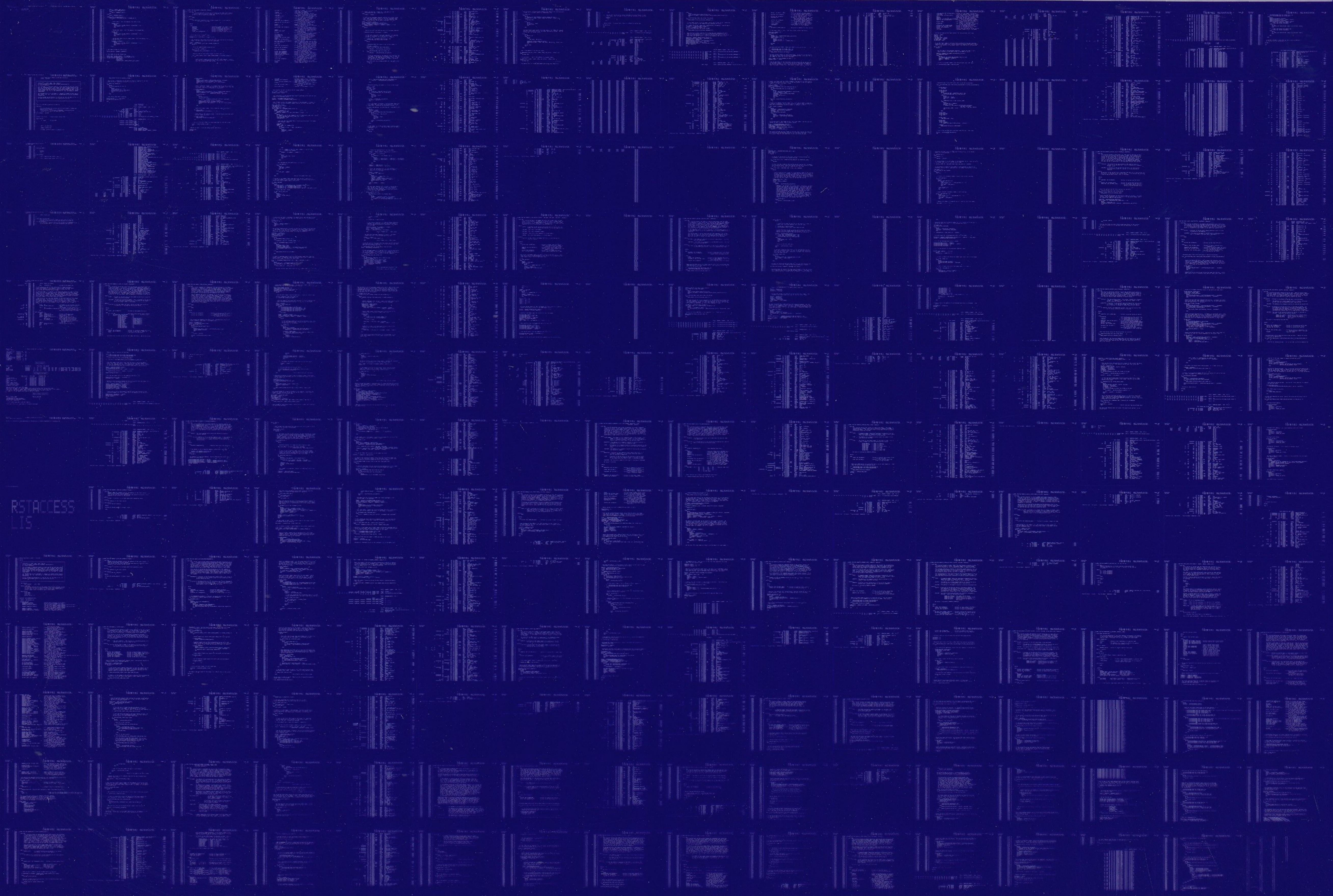
DSTRECRDS
LIS

ISSH
LIS

RESETSSI
LIS

0098 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



RSTACCESS
LIS