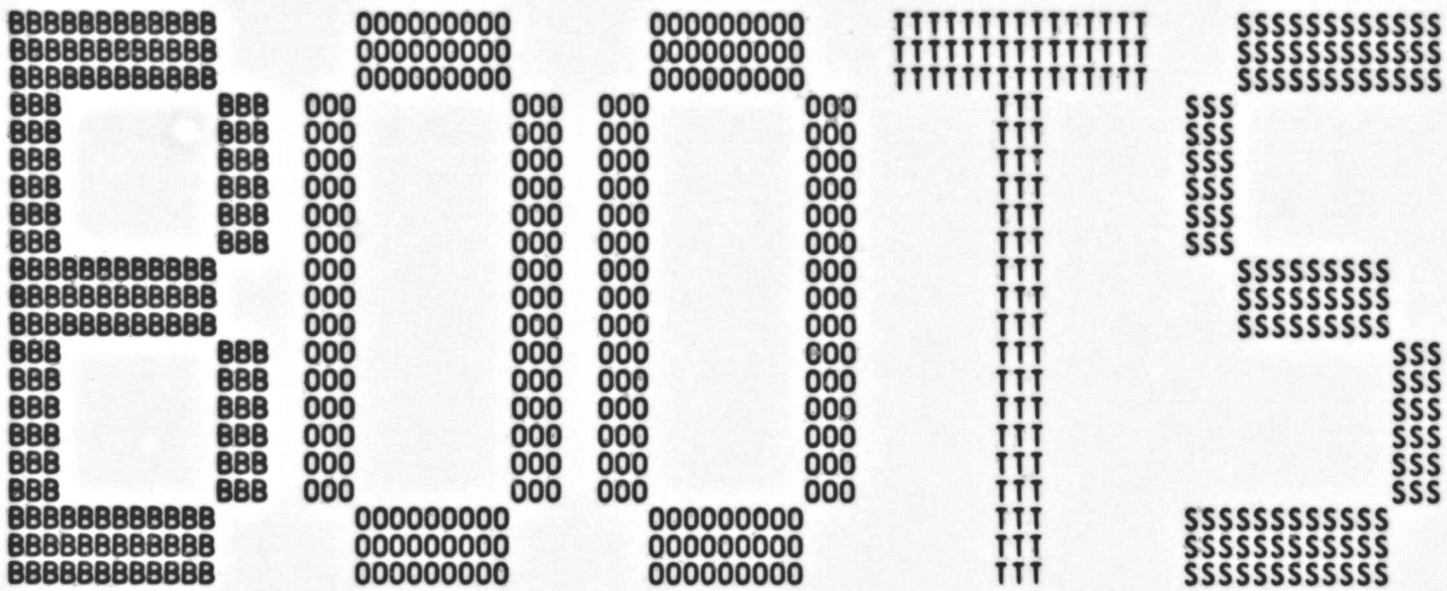
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTTTTTTTTTTTTTT SSSSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOO OOO OOO OOO TTT SSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTT SSSSSSSSSSS

FILEID**READDRIV

```
RRRRRRRR    EEEEEEEEEE    AAAAAA    DDDDDDDD    DDDDDDDD    RRRRRRRR      IIIIII    VV      VV
RRRRRRRR    EEEEEEEEEE    AAAAAA    DDDDDDDD    DDDDDDDD    RRRRRRRR      IIIIII    VV      VV
RR      RR  EE            AA  AA    DD    DD    DD    DD    RR      RR      II      VV      VV
RR      RR  EE            AA  AA    DD    DD    DD    DD    RR      RR      II      VV      VV
RR      RR  EE            AA  AA    DD    DD    DD    DD    RR      RR      II      VV      VV
RR      RR  EE            AA  AA    DD    DD    DD    DD    RR      RR      II      VV      VV
RRRRRRRR    EEEEEEEE      AA  AA    DD    DD    DD    DD    RRRRRRRR        II      VV      VV
RRRRRRRR    EEEEEEEE      AA  AA    DD    DD    DD    DD    RRRRRRRR        II      VV      VV
RR  RR      EE            AAAAAAAAAA    DD    DD    DD    DD    RR  RR      II      VV      VV
RR  RR      EE            AAAAAAAAAA    DD    DD    DD    DD    RR  RR      II      VV      VV
RR    RR    EE            AA  AA    DD    DD    DD    DD    RR    RR        II        VV  VV
RR    RR    EE            AA  AA    DD    DD    DD    DD    RR    RR        II        VV  VV
RR      RR  EEEEEEEEEE    AA  AA    DDDDDDDD    DDDDDDDD    RR      RR    IIIIII        VV
RR      RR  EEEEEEEEEE    AA  AA    DDDDDDDD    DDDDDDDD    RR      RR    IIIIII        VV

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     1          .TITLE  READDRIV - READ DRIVER INTO MEMORY
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28
0000    29  ;++
0000    30  ; FACILITY:     SYSGEN
0000    31  ;
0000    32  ; ABSTRACT:
0000    33  ;       THIS ROUTINE READS A DRIVER INTO MEMORY BY MAPPING IT ($CRMPSC)
0000    34  ;
0000    35  ; ENVIRONMENT:  USER MODE
0000    36  ;
0000    37  ; AUTHOR:  STEVE BECKHARDT,     CREATION DATE:  21-SEP-1979
0000    38  ;          (ORIGINAL AUTHOR - LEN KAWELL)
0000    39  ;
0000    40  ; MODIFICATION HISTORY:
0000    41  ;
0000    42  ;       V02-005 JLV0038         Jake VanNoy             13-Jul-1981
0000    43  ;                Add G^ to LIB$ routines. Return default file name
0000    44  ;                to SYS$SYSTEM:.EXE.
0000    45  ;
0000    46  ;       V04      JLV0013         Jake VanNoy             7-May-1981
0000    47  ;                Suppress error messages from unsupported device drivers.
0000    48  ;
0000    49  ;       V03      JLV0002         Jake VanNoy             9-Feb-1981
0000    50  ;                Changed psects to allow paged and non-paged segments
0000    51  ;                is SYSGEN image. Add SHOW/CONFIGURATION and
0000    52  ;                CONFIGURE.
0000    53  ;
0000    54  ;       V02      LMK0001         LEN KAWELL
0000    55  ;                CHANGE TO READ IMAGE HEADER AS WELL.
0000    56  ;--
```

```
                     58          .SBTTL  DECLARATIONS
                     59 ;
                     60 ; INCLUDE FILES:
                     61 ;
                     62
                     63 ;
                     64 ; MACROS:
                     65 ;
                     66
                     67 ;
                     68 ; EQUATED SYMBOLS:
                     69 ;
                     70
                     71          $ACFDEF
                     72          $SECDEF                             ; DEFINE $CRMPSC FLAGS
                     73 ;
                     74 ; OWN STORAGE:
                     75 ;
                     76
                     77          .PSECT  PAGED_DATA      rd,wrt,noexe,quad
                     78
                     79 IN_FAB:                                      ; INPUT FILE FAB
                     80          $FAB    FOP = UFO,-                 ; RMS WILL JUST OPEN IT
                     81                  NAM = IN_NAM,-              ; NAME BLOCK ADDRESS
                     82                  DNM = <SYS$SYSTEM:.EXE>     ; DEFAULT FILE NAME
                     83
                     84 IN_NAM:                                      ; INPUT FILE NAME BLOCK
                     85          $NAM    ESA = NAME,-                ; EXPANDED STRING BUFFER ADDRESS
                     86                  ESS = NAM$C_MAXRSS          ; EXPANDED STRING BUFFER SIZE
                     87
                     88 NAME:                                        ; EXPANDED STRING BUFFER
                     89          .BLKB   NAM$C_MAXRSS
                     90
                     91 MAP_RANGE:                                   ; MAP RANGE ARRAY
00000000             92          .LONG   0                           ; JUST USED TO INDICATE P0 SPACE
00000000             93          .LONG   0
                     94
                     95
                     96          .PSECT  PAGED_CODE      rd,nowrt,exe,long
```

```
                        98          .SBTTL  READ DRIVER
                        99 ;++
                       100 ; FUNCTIONAL DESCRIPTION:
                       101 ;
                       102 ;       This routine maps a driver into memory by doing a create and
                       103 ;       map section.
                       104 ;
                       105 ; CALLING SEQUENCE:
                       106 ;
                       107 ;       BSBW    IOGEN$READDRIV
                       108 ;
                       109 ; INPUT PARAMETERS:
                       110 ;
                       111 ;       R0      Address of filename counted string
                       112 ;       R3      Address of location to store channel number
                       113 ;       R4      Address of two longword array to return address range
                       114 ;               created by $CRMPSC
                       115 ;       AP      Address of ACF block
                       116 ;
                       117 ; IMPLICIT INPUTS:
                       118 ;
                       119 ;       NONE
                       120 ;
                       121 ; OUTPUT PARAMETERS:
                       122 ;
                       123 ;       R0      Completion code
                       124 ;
                       125 ; IMPLICIT OUTPUTS:
                       126 ;
                       127 ;       NONE
                       128 ;
                       129 ; COMPLETION CODES:
                       130 ;
                       131 ;       Those returned by $OPEN and $CRMPSC
                       132 ;
                       133 ; SIDE EFFECTS:
                       134 ;
                       135 ;       R0 - R2 are used as scratch registers
                       136 ;
                       137 ;--
                       138
                       139 IOGEN$READDRIV::
52   0000'CF   DE      140         MOVAL   W^IN_FAB,R2             ; Get address of input FAB
  34 A2   80   90      141         MOVB    (R0)+,FAB$B_FNS(R2)     ; Store filename size in FAB
  2C A2   60   9E      142         MOVAB   (R0),FAB$L_FNA(R2)      ; Store filename address in FAB
                       143 ;
                       144 ; Open the image file
                       145 ;
                       146         $OPEN   (R2)                    ; Open the file
        32 50   E8     147         BLBS    R0,20$                  ; Br. if success
                       148
51 00000000'EF   DE    149         moval   Boo$al_acf,R1           ; Set address of acf block (not in AP)
25 0B A1   03   E0     150         bbs     #acf$v_noload_db,acf$b_aflag(R1),10$ ; Branch if Load driver call
20 0B A1   04   E0     151         bbs     #acf$v_support,acf$b_aflag(R1),10$   ; Branch if NOSUPPORT
                       152
      00B0'CF   9F     153         PUSHAB  W^NAME                  ; Create expanded name string
7E    005B'CF   9A     154         MOVZBL  W^NAM$B_ESL+IN_NAM,-(SP); descriptor
```

```
                51   5E   D0  0033   155          MOVL    SP,R1                   ; Save address of descriptor
                     51   DD  0036   156          PUSHL   R1                      ; Push addr. of file name desc.
                     01   DD  0038   157          PUSHL   #1                      ; Push FAO count
          007C0000'8F     DD  003A   158          PUSHL   #SHR$_OPENIN!<124@16>   ; Push error status
00000000'GF       03     FB  0040   159          CALLS   #3,G^LIB$SIGNAL         ; Signal the error
                5E   08   C0  0047   160          ADDL    #8,SP                   ; Pop descriptor off stack
                          05  004A   161 10$:     RSB
                              004B   162 ;
                              004B   163 ; Map the file
                              004B   164 ;
              63   0C A2    B0  004B   165 20$:     MOVW    FAB$L_STV(R2),(R3)      ; Store channel number
                              004F   166          $CRMPSC_S CHAN = (R3),-         ; Channel
                              004F   167                    INADR = W^MAP_RANGE,- ; Map in P0 space
                              004F   168                    FLAGS = #SEC$M_EXPREG,- ; Expand region
                              004F   169                    RETADR = (R4),-       ; Return address array
                              004F   170                    VBN = #1              ; Map entire image file
                              0073   171
                          05  0073   172          RSB                             ; Status in R0
                              0074   173
                              0074   174
                              0074   175          .END
```

```
$$.TAB                          = 00000050 R      02
$$.TABEND                       = 000000B0 R      02
$$.TMP                          = 00000000
$$.TMP1                         = 00000001
$$.TMP2                         = 00000062
$$.TMPX                         = 00000000 R      03
$$.TMPX1                        = 0000000F
$$T1                            = 00000000
ACF$B_AFLAG                     = 0000000B
ACF$V_NOLOAD_DB                 = 00000003
ACF$V_SUPPORT                   = 00000004
BOO$AC_ACF                        ********    X   04
FAB$B_DNS                       = 00000035
FAB$B_FNS                       = 00000034
FAB$C_BID                       = 00000003
FAB$C_BLN                       = 00000050
FAB$C_SEQ                       = 00000000
FAB$C_VAR                       = 00000002
FAB$L_ALQ                       = 00000010
FAB$L_DNA                       = 00000030
FAB$L_FNA                       = 0000002C
FAB$L_FOP                       = 00000004
FAB$L_STV                       = 0000000C
FAB$V_CHAN_MODE                 = 00000002
FAB$V_FILE_MODE                 = 00000004
FAB$V_LNM_MODE                  = 00000000
FAB$V_UFO                       = 00000011
FAB$W_GBC                       = 00000048
IN_FAB                            00000000 R      02
IN_NAM                            00000050 R      02
IOGEN$READDRIV                    00000000 RG     04
LIB$SIGNAL                        ********   X    04
MAP_RANGE                         000001AF R      02
NAM$B_ESL                       = 0000000B
NAM$B_ESS                       = 0000000A
NAM$B_NOP                       = 00000008
NAM$B_RSS                       = 00000002
NAM$C_BID                       = 00000002
NAM$C_BLN                       = 00000060
NAM$C_MAXRSS                    = 000000FF
NAM$L_ESA                       = 0000000C
NAM$L_RSA                       = 00000004
NAME                              000000B0 R      02
SEC$M_EXPREG                    = 00020000
SHR$_OPENIN                       ********    X   04
SYS$CRMPSC                        ********   GX   04
SYS$OPEN                          ********   GX   04
```

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
PAGED_DATA	000001B7 (439.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
$RMSNAM	0000000F (15.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
PAGED_CODE	00000074 (116.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.34
Command processing	107	00:00:00.65	00:00:02.38
Pass 1	219	00:00:05.06	00:00:12.91
Symbol table sort	0	00:00:00.53	00:00:00.65
Pass 2	48	00:00:00.89	00:00:01.72
Symbol table output	7	00:00:00.06	00:00:00.06
Psect synopsis output	3	00:00:00.04	00:00:00.06
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	415	00:00:07.30	00:00:18.13

The working set limit was 1200 pages.
24045 bytes (47 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 440 non-local and 2 local symbols.
175 source lines were read in Pass 1, producing 17 object records in Pass 2.
25 pages of virtual memory were used to define 20 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[BOOTS.OBJ]BOOTS.MLB;1	0
_255DUA28:[SYS.OBJ]LIB.MLB;1	1
_255DUA28:[SYSLIB]STARLET.MLB;2	16
TOTALS (all libraries)	17

702 GETS were required to define 17 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:READDRIV/OBJ=OBJ$:READDRIV MSRC$:READDRIV/UPDATE=(ENH$:READDRIV)+EXECML$/LIB+LIB$:BOOTS.MLB/LIB
```

READLBN
LIS

RTFILREAD
LIS

SHARE
LIS

QUSS
LIS

RWBTDRIVR
LIS

MBBTDRIVR
LIS

SCSLOADER
LIS

RMSCONIO
LIS

READDRIV
LIS

PABTDRIVR
LIS

PUBTDRIVR
LIS

PUTERROR
LIS

READPRMPT
LIS