
SSSSSSSSSSS DDDDDDDDDDD AAAAAAAAA
SSSSSSSSSSS DDDDDDDDDDD AAAAAAAAA
SSSSSSSSSSS DDDDDDDDDDD 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
SSSSSSSSSSS DDDDDDDDDDD AAA AAA
SSSSSSSSSSS DDDDDDDDDDD AAA AAA
SSSSSSSSSSS DDDDDDDDDDD AAA AAA

```
MM       MM  MM       MM    GGGGGGGG
MM       MM  MM       MM    GGGGGGGG
MMMM   MMMM  MMMM   MMMM  GG
MMMM   MMMM  MMMM   MMMM  GG
MM  MM  MM  MM  MM  MM  GG
MM  MM  MM  MM  MM  MM  GG
MM      MM  MM      MM  GG
MM      MM  MM      MM  GG
MM      MM  MM      MM  GG  GGGGGG
MM      MM  MM      MM  GG  GGGGGG
MM      MM  MM      MM  GG      GG      ....
MM      MM  MM      MM  GG      GG      ....
MM      MM  MM      MM    GGGGGG        ....
MM      MM  MM      MM    GGGGGG        ....


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  MMG     PAGE TABLE FORMATTING ROUTINES
0000     2          .SBTTL  COPYRIGHT NOTICE
0000     3          .IDENT  'V04-000'
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     29          .SBTTL  PROGRAM DESCRIPTION
0000     30 ;++
0000     31 ;   FACILITY
0000     32 ;
0000     33 ;       SYSTEM DUMP ANALYZER
0000     34 ;
0000     35 ;   ABSTRACT
0000     36 ;
0000     37 ;       THIS MODULE CONTAINS ROUTINES RELATING TO FORMATTED
0000     38 ;       A SPECIFIED PAGE TABLE.
0000     39 ;
0000     40 ;   ENVIRONMENT
0000     41 ;
0000     42 ;       NATIVE MODE, USER MODE
0000     43 ;
0000     44 ;   AUTHOR
0000     45 ;
0000     46 ;       TIM HALVORSEN, JULY 1978
0000     47 ;
0000     48 ;   MODIFIED BY
0000     49 ;
0000     50 ;       V03-001 WMC0001         Wayne Cardoza   19-Aug-1982
0000     51 ;               Correct the use of MAXPFN to allow the last page.
0000     52 ;
0000     53 ;--
```

```
                        .SBTTL  DECLARATIONS
                ;
                ;       SYMBOL DEFINTIONS
                ;
                        $DMPDEF                         ; Dump file definitions
                        $OPDEF                          ; Define opcode equivalences
                        $OPTDEF                         ; Options definitions
                        $PFNDEF                         ; Page frame data definitions
                        $PHDDEF                         ; Process header definitions
                        $PTEDEF                         ; Page table entry definitions
                        $TPADEF                         ; TPARSE definitions
                        $VADEF                          ; Virtual address definitions
                        $WSLDEF                         ; Working set list definitions
```

```
                     0000    71          .SBTTL  STORAGE DEFINTIONS
                     0000    72 ;
                     0000    73 ;        WRITABLE STORAGE DEFINITIONS
                     0000    74 ;
                     0000    75
                00000000     76          .PSECT  SDADATA,NOEXE,WRT
                     0000    77
                     0000    78 BUFFER:
00000040             0000    79          .BLKL   16                      ; GETMEM WORK BUFFER
                     0040    80
                     0040    81 SDA$GL_MAXPFN:
00000044             0040    82          .BLKL   1                       ; VALUE OF MMG$GL_MAXPFN
00000042             0044    83          MMG$GW_BIGPFN = SDA$GL_MAXPFN + 2
                     0044    84 SDA$AB_STATE:
00000048             0044    85          .BLKL   1                       ; VALUE OF PFN$AB_STATE
                     0048    86 SDA$AB_TYPE:
0000004C             0048    87          .BLKL   1                       ; PFN$AB_TYPE
                     004C    88 SDA$AW_REFCNT:
00000050             004C    89          .BLKL   1                       ; PFN$AW_REFCNT
                     0050    90 SDA$AL_BAK:
00000054             0050    91          .BLKL   1                       ; PFN$AL_BAK
                     0054    92 SDA$AL_PTE:
00000058             0054    93          .BLKL   1                       ; PFN$AL_PTE
                     0058    94 SDA$Ax_FLINK:
0000005C             0058    95          .BLKL   1                       ; PFN$AW_FLINK
                     005C    96 SDA$Ax_BLINK:
00000060             005C    97          .BLKL   1                       ; PFN$AW_BLINK
                     0060    98 SDA$Ax_WSLX:
00000064             0060    99          .BLKL   1                       ; WORKING SET INDEX
                     0064   100
                00000000    101          .PSECT  MMG,EXE,NOWRT
                     0000   102
                     0000   103          .DEFAULT DISPLACEMENT,LONG
```

```
                        0000    106          .SBTTL  READ-ONLY DATA DEFINITIONS
                        0000    107
                        0000    108 ;
                        0000    109 ;       READ-ONLY DATA DEFINITIONS
                        0000    110 ;
                        0000    111
                        0000    112 PTECTL1:
                        0000    113          STRING  <!_!XL    !XL !XL    !AD !AD !AD !AD !AD>
                        002F    114 PTECTL2_WORD:
                        002F    115          STRING  <!_!XL    !XL !XL    !AD !AD !AD !AD !AD !AD !AD !XB   !XB !6UW   !X
                        008B    116 PTECTL2_LONG:
                        008B    117          STRING  <!_!XL    !XL !XL    !AD !AD !AD !AD !AD !AD !AD !XB   !XB !6UW   !X
                        00E7    118
                        00E7    119 PROT_TABLE:
45 4E 4F 4E             00E7    120          .ASCII  /NONE/
2A 2A 2A 2A             00EB    121          .ASCII  /****/
20 20 57 4B             00EF    122          .ASCII  /KW  /
20 20 52 4B             00F3    123          .ASCII  /KR  /
20 20 57 55             00F7    124          .ASCII  /UW  /
20 20 57 45             00FB    125          .ASCII  /EW  /
57 4B 52 45             00FF    126          .ASCII  /ERKW/
20 20 52 45             0103    127          .ASCII  /ER  /
20 20 57 53             0107    128          .ASCII  /SW  /
57 45 52 53             010B    129          .ASCII  /SREW/
57 4B 52 53             010F    130          .ASCII  /SRKW/
20 20 52 53             0113    131          .ASCII  /SR  /
57 53 52 55             0117    132          .ASCII  /URSW/
57 45 52 55             011B    133          .ASCII  /UREW/
57 4B 52 55             011F    134          .ASCII  /URKW/
20 20 52 55             0123    135          .ASCII  /UR  /
                        0127    136
                        0127    137 OWNER_TABLE:
55 53 45 4B             0127    138          .ASCII  /KESU/
                        012B    139
                        012B    140 TYPE_TABLE:
53 4E 41 52 54          012B    141          .ASCII  /TRANS/
20 58 54 50 47          0130    142          .ASCII  /GPTX /
4C 49 46 47 50          0135    143          .ASCII  /PGFIL/
20 20 58 54 53          013A    144          .ASCII  /STX  /
4F 52 45 5A 44          013F    145          .ASCII  /DZERO/
44 49 4C 41 56          0144    146          .ASCII  /VALID/
47 41 50 4F 49          0149    147          .ASCII  /IOPAG/
20 20 20 20 20          014E    148          .ASCII  /     /
                        0153    149
                        0153    150 LOC_TABLE:
20 54 53 4C 45 45 52 46 0153    151          .ASCII  /FREELST /
20 54 53 4C 59 46 44 4D 015B    152          .ASCII  /MDFYLST /
20 54 53 49 4C 44 41 42 0163    153          .ASCII  /BADLIST /
20 44 4E 45 50 4C 45 52 016B    154          .ASCII  /RELPEND /
20 52 4F 52 52 45 44 52 0173    155          .ASCII  /RDERROR /
20 54 55 4F 45 47 41 50 017B    156          .ASCII  /PAGEOUT /
20 20 4E 49 45 47 41 50 0183    157          .ASCII  /PAGEIN  /
20 20 45 56 49 54 43 41 018B    158          .ASCII  /ACTIVE  /
                        0193    159
                        0193    160 PAGTYP_TABLE:
20 53 53 45 43 4F 52 50 0193    161          .ASCII  /PROCESS /
20 20 4D 45 54 53 59 53 019B    162          .ASCII  /SYSTEM  /
```

```
20 20 4C 41 42 4F 4C 47  01A3   163          .ASCII  /GLOBAL  /
20 20 54 52 57 4C 42 47  01AB   164          .ASCII  /GBLWRT  /
20 20 4C 42 54 47 50 50  01B3   165          .ASCII  /PPGTBL  /
20 20 4C 42 54 47 50 47  01BB   166          .ASCII  /GPGTBL  /
20 20 20 20 20 20 20 20  01C3   167          .ASCII  /        /
20 20 20 20 20 20 20 20  01CB   168          .ASCII  /        /
                         01D3   169
                         01D3   170 MODIFY_TABLE:
                      4D 20 01D3   171          .ASCII  / M/
                         01D5   172
                         01D5   173 WSLOCK_TABLE:
                      4C 20 01D5   174          .ASCII  / L/
```

```
                        01D7   177          .SBTTL  INIT_PFN -- INITIALIZE FOR EXAMINING PFN DATA BASE
                        01D7   178 ;---
                        01D7   179 ;
                        01D7   180 ;       INIT_PFN
                        01D7   181 ;
                        01D7   182 ;       THIS ROUTINE MUST BE CALLED BEFORE ANY REFERENCES ARE
                        01D7   183 ;       MADE TO THE PFN DATA BASE.
                        01D7   184 ;
                        01D7   185 ;   INPUTS:
                        01D7   186 ;
                        01D7   187 ;       NONE
                        01D7   188 ;
                        01D7   189 ;   OUTPUTS:
                        01D7   190 ;
                        01D7   191 ;       R0 = SUCCESS FLAG
                        01D7   192 ;       SDA$A... CELLS ARE INITIALIZED
                        01D7   193 ;
                        01D7   194 ;---
                        01D7   195
                        01D7   196 INIT_PFN::
                0000    01D7   197          .WORD   0
                        01D9   198
                        01D9   199          REQMEM  @MMG$GL_MAXPFN,SDA$GL_MAXPFN
                        01ED   200          REQMEM  @PFN$AB_STATE,SDA$AB_STATE
                        0201   201          REQMEM  @PFN$AB_TYPE,SDA$AB_TYPE
                        0215   202          REQMEM  @PFN$AW_REFCNT,SDA$AW_REFCNT
                        0229   203          REQMEM  @PFN$AL_BAK,SDA$AL_BAK
                        023D   204          REQMEM  @PFN$AL_PTE,SDA$AL_PTE
                        0251   205          REQMEM  @PFN$Ax_FLINK,SDA$Ax_FLINK
                        0265   206          REQMEM  @PFN$Ax_BLINK,SDA$Ax_BLINK
                        0279   207          REQMEM  @PFN$Ax_WSLX,SDA$Ax_WSLX
                  04    028D   208          RET
```

```
                        028E   211          .SBTTL  DISPLAY_PFN     DISPLAY MEMORY MANAGEMENT DATA
                        028E   212 ;---
                        028E   213 ;
                        028E   214 ;       DISPLAY_PFN
                        028E   215 ;
                        028E   216 ;       THIS ROUTINE IS RESPONSIBLE FOR PRINTING ALL INFORMATION
                        028E   217 ;       RELATING TO THE MEMORY MANAGEMENT DATA BASE.
                        028E   218 ;
                        028E   219 ;   INPUTS:
                        028E   220 ;
                        028E   221 ;       NONE
                        028E   222 ;
                        028E   223 ;   OUTPUTS:
                        028E   224 ;
                        028E   225 ;       NONE
                        028E   226 ;
                        028E   227 ;---
                        028E   228          .ENABL  LSB
                        028E   229
                   007C 028E   230 .ENTRY  DISPLAY_PFN,^M<R2,R3,R4,R5,R6>
                        0290   231
          FF42 CF   00   FB 0290   232          CALLS   #0,INIT_PFN             ; SETUP TO READ PFN DATA
38 00000000'EF      04   E1 0295   233          BBC     #OPT$V_SINGLEPFN,OPTIONS,20$    ; BRANCH IF LIST WANTED
                        029D   234 ;
                        029D   235 ;       DISPLAY A SINGLE SPECIFIED PFN ENTRY
                        029D   236 ;
             56   1C AC   D0 029D   237          MOVL    TPA$L_NUMBER(AP),R6     ; R6 = PFN TO DISPLAY
   00000040'EF       56   D1 02A1   238          CMPL    R6,SDA$GL_MAXPFN        ; CHECK IF PFN VALID
                     17   1A 02A8   239          BGTRU   10$                     ; BRANCH IF INVALID PFN
        047C'CF      00   FB 02AA   240          CALLS   #0,W^PFN_TITLE          ; DISPLAY THE TITLE LINE
        04CF'CF      00   FB 02AF   241          CALLS   #0,W^SHOW_PFN           ; DISPLAY THE PFN DATA
                        02B4   242          SKIP    1
                50   01   D0 02BD   243          MOVL    #1,R0
                     04   02C0   244          RET
                        02C1   245 10$:
          00000040'EF      DD 02C1   246          PUSHL   SDA$GL_MAXPFN
                        02C7   247          PRINT   1,<Invalid PFN number (maximum is !XL)>
                     04 02D4   248          RET
                        02D5   249 20$:
    52  00000000'EF      D0 02D5   250          MOVL    SCH$GL_FREECNT,R2       ; ADDRESS OF COUNT ARRAY
    53  00000000'EF      D0 02DC   251          MOVL    PFN$AL_LOLIMIT,R3       ; ADDRESS OF LOLIMIT ARRAY
    54  00000000'EF      D0 02E3   252          MOVL    PFN$AL_HILIMIT,R4       ; ADDRESS OF HILIMIT ARRAY
    55  00000000'EF      D0 02EA   253          MOVL    PFN$AL_HEAD,R5          ; ADDRESS OF LIST HEADS
                        02F1   254
19 00000000'EF      00   E1 02F1   255          BBC     #OPT$V_FREE,OPTIONS,30$ ; BRANCH IF NO FREE LIST
                        02F9   256          SUBHD   <Free page list>
                        0306   257          SKIP    PAGE
        03A3'CF      00   FB 030D   258          CALLS   #0,W^SHOW_PFN_LIST      ; DISPLAY FREE PAGE LIST
                        0312   259 30$:
                     82   D5 0312   260          TSTL    (R2)+
                     83   D5 0314   261          TSTL    (R3)+
                     84   D5 0316   262          TSTL    (R4)+
                     85   D5 0318   263          TSTL    (R5)+
18 00000000'EF      01   E1 031A   264          BBC     #OPT$V_MODIFIED,OPTIONS,40$     ; BRANCH IF NO MODIFIED
                        0322   265          SUBHD   <Modified page list>
                        032F   266          SKIP    PAGE
          A3'AF      00   FB 0336   267          CALLS   #0,B^SHOW_PFN_LIST      ; DISPLAY MODIFIED PAGE LIST
```

```
                                033A   268 40$:
                          82    D5 033A   269          TSTL    (R2)+
                          83    D5 033C   270          TSTL    (R3)+
                          84    D5 033E   271          TSTL    (R4)+
                          85    D5 0340   272          TSTL    (R5)+
                18 00000000'EF    02 E1 0342   273          BBC     #OPT$V_BAD,OPTIONS,50$  ; BRANCH IF NO BAD LIST
                                0034A   274          SUBHD   <Bad page list>
                                0357   275          SKIP    PAGE
                        A3'AF   00    FB 035E   276          CALLS   #0,B^SHOW_PFN_LIST      ; DISPLAY BAD PAGE LIST
                                0362   277 ;
                                0362   278 ;       PRINT ENTIRE PFN DATA FROM ENTRY 0 TO N
                                0362   279 ;
                                0362   280 50$:
                31 00000000'EF    03    E1 0362   281          BBC     #OPT$V_WHOLEPFN,OPTIONS,70$     ; BRANCH IF NOT WANTED
                                036A   282          SUBHD   <PFN data base>
             00000000'EF    047C'CF    9E 0377   283          MOVAB   W^PFN_TITLE,HEADING_ROUTINE     ; SET HEADING ROUTINE
                                0380   284          SKIP    PAGE
                          56    D4 0387   285          CLRL    R6                              ; START AT PFN 0
                000004CF'EF    00    FB 0389   286 60$:     CALLS   #0,SHOW_PFN             ; SHOW PFN IN R6
                          56    D6 0390   287          INCL    R6                              ; SKIP TO NEXT PFN
                00000040'EF    56    D1 0392   288          CMPL    R6,SDA$GL_MAXPFN        ; CHECK IF LAST PFN
                          EE    1B 0399   289          BLEQU   60$                             ; LOOP UNTIL DONE
                                039B   290 70$:
                                039B   291          STATUS  SUCCESS
                          04    03A2   292          RET
                                03A3   293
                                03A3   294          .DSABL  LSB
```

```
                     03A3   297          .SBTTL  SHOW_PFN_LIST, DISPLAY PFN LIST
                     03A3   298 ;---
                     03A3   299 ;
                     03A3   300 ;       SHOW_PFN_LIST
                     03A3   301 ;
                     03A3   302 ;       THIS ROUTINE DISPLAYS THE PFN DATA FOR THE FREE,
                     03A3   303 ;       MODIFIED AND BAD PAGE LISTS.
                     03A3   304 ;
                     03A3   305 ;   INPUTS:
                     03A3   306 ;
                     03A3   307 ;       R2 = ADDRESS OF COUNT LONGWORD
                     03A3   308 ;       R3 = ADDRESS OF LOLIMIT LONGWORD
                     03A3   309 ;       R4 = ADDRESS OF HILIMIT LONGWORD
                     03A3   310 ;       R5 = ADDRESS OF LIST HEAD LONGWORD
                     03A3   311 ;
                     03A3   312 ;---
                     03A3   313
                     03A3   314          .ENABL  LSB
                     03A3   315
                     03A3   316 SHOW_PFN_LIST:
             0040    03A3   317          .WORD   ^M<R6>
                     03A5   318
                     03A5   319          SKIP    1
                     03AE   320          GETMEM  (R2),-(SP)              ; GET LIST COUNT
          0D 50  E9  03BA   321          BLBC    R0,10$
                     03BD   322          PRINT   1,<Count:       !12SL>
                     03CA   323 10$:
                     03CA   324          GETMEM  (R3),-(SP)              ; GET LIST LOLIMIT
          0D 50  E9  03D6   325          BLBC    R0,20$
                     03D9   326          PRINT   1,<Lolimit:     !12SL>
                     03E6   327 20$:
                     03E6   328          GETMEM  (R4),-(SP)              ; GET LIST HILIMIT
          0D 50  E9  03F2   329          BLBC    R0,30$
                     03F5   330          PRINT   1,<High limit:  !12SL>
                     0402   331 30$:
       7C'AF   00  FB  0402   332          CALLS   #0,B^PFN_TITLE          ; PRINT HEADING LINE
00000000'EF  0000047C'EF  9E  0406   333          MOVAB   PFN_TITLE,HEADING_ROUTINE       ; SET HEADING ROUTINE
                     0411   334          GETMEM  (R5),R6                 ; GET LIST HEAD
          03 50  E8  041D   335          BLBS    R0,35$
           0052  31  0420   336 80$:     BRW     90$
                     0423   337 35$:
             10  12  0423   338          BNEQ    40$                     ; BRANCH IF NON-EMPTY LIST
                     0425   339          PRINT   0,<*** List is empty ***>
           0040  31  0432   340          BRW     90$
                     0435   341 40$:
     04CF'CF   00  FB  0435   342          CALLS   #0,W^SHOW_PFN           ; DISPLAY PFN IN R6
                     043A   343                  PFN_REFERENCE   -
                     043A   344          MOVAW   <@SDA$Ax_FLINK[R6],R1>,-
                     043A   345                  LONG_OPCODE=MOVAL,-
                     043A   346                  IMAGE=SDA
                     0454   347          GETMEM  (R1)
          15 50  E9  045D   348          BLBC    R0,90$                  ; SKIP IF ERROR
                     0460   349                  PFN_REFERENCE   -
                     0460   350          MOVZWL  <R1,R6>,-               ; SKIP TO NEXT ENTRY IN LIST
                     0460   351                  LONG_OPCODE=MOVL,-
                     0460   352                  IMAGE=SDA
             03  13  0470   353          BEQL    90$                     ; LOOP UNTIL END OF LIST
```

```
                FFC0    31  0472    354          BRW     40$
                            0475    355 90$:
00000000'EF     D4      0475    356          CLRL    HEADING_ROUTINE         ; CLEAR HEADING ROUTINE ADDRESS
                04      047B    357          RET
                        047C    358
                        047C    359          .DSABL  LSB
```

```
                        047C    361          .SBTTL  PFN_TITLE, DISPLAY PFN HEADING LINE
                        047C    362 ;---
                        047C    363 ;
                        047C    364 ;       PFN_TITLE
                        047C    365 ;
                        047C    366 ;       DISPLAY THE HEADING LINE FOR THE PFN DATA DISPLAY
                        047C    367 ;
                        047C    368 ;---
                        047C    369
                        047C    370          .ENABLE         LOCAL_BLOCK
                        047C    371
                        047C    372 PFN_TITLE:
                  0000  047C    373          .WORD   0
                        047E    374
                        047E    375          SKIP    1
                        0487    376          PFN_DISP_IF_BIGPFN_THEN
                        048F
                        048F                 ;This code executes if the PFN link arrays are longword arrays.
                        048F    377                  PRINT   0,<  PFN      PTE ADDRESS    BAK      REFCNT    FLINK      BL
                        049C    378                  PRINT   0,<  ----     -----------  --------   ------    -----      --
                        04A9    379          PFN_DISP_ELSE
                        04AB
                        04AB                 ;This code executes if the PFN link arrays are word arrays.
                        04AB    380                  PRINT   0,<PFN    PTE ADDRESS    BAK      REFCNT  FLINK BLINK     TY
                        04B8    381                  PRINT   0,<----   -----------  --------   ------  ----- -----    ----
                        04C5    382          PFN_DISP_ENDIF
                        04C5
                        04C5                 ;End of code that depends on size of PFN link arrays
                        04C5    383          SKIP    1
                  04    04CE    384          RET
                        04CF    385
                        04CF    386          .DISABLE        LOCAL_BLOCK
```

```
                                04CF    388          .SBTTL  SHOW_PFN, SHOW DATA ON A SINGLE PFN ENTRY
                                04CF    389 ;---
                                04CF    390 ;
                                04CF    391 ;       SHOW_PFN
                                04CF    392 ;
                                04CF    393 ;       THIS ROUTINE DISPLAYS THE PFN DATA BASE ASSOCIATED
                                04CF    394 ;       WITH A SINGLE PAGE FRAME NUMBER.
                                04CF    395 ;
                                04CF    396 ; INPUTS:
                                04CF    397 ;
                                04CF    398 ;       R6 = PAGE FRAME NUMBER
                                04CF    399 ;
                                04CF    400 ; OUTPUTS:
                                04CF    401 ;
                                04CF    402 ;       THE ENTRY IS DISPLAYED.
                                04CF    403 ;
                                04CF    404 ;---
                                04CF    405          .ENABL  LSB
                                04CF    406
                                04CF    407 SHOW_PFN:
                        0000    04CF    408          .WORD   0
                                04D1    409
     51  00000044'FF46    9E    04D1    410          MOVAB   @SDA$AB_STATE[R6],R1    ; GET PFN STATE
                                04D9    411          GETMEM  (R1)
                   55 50    E9  04E2    412          BLBC    R0,70$                  ; SKIP IF ERROR
      5D   51    03   00    EF  04E5    413          EXTZV   #PFN$V_LOC,#PFN$S_LOC,R1,R0
              FC64 CF40    7F  04EA    414          PUSHAQ  LOC_TABLE[R0]           ; ADDRESS OF STRING
                     07    DD  04EF    415          PUSHL   #7                      ; LENGTH OF STRING
                  7E   51    9A  04F1    416          MOVZBL  R1,-(SP)
     51  00000048'FF46    9E    04F4    417          MOVAB   @SDA$AB_TYPE[R6],R1     ; GET PFN TYPE
                                04FC    418          GETMEM  (R1)
                   32 50    E9  0505    419          BLBC    R0,70$                  ; SKIP IF ERROR
     50   51    03   00    EF  0508    420          EXTZV   #PFN$V_PAGTYP,#PFN$S_PAGTYP,R1,R0
              FC81 CF40    7F  050D    421          PUSHAQ  PAGTYP_TABLE[R0]        ; ADDRESS OF STRING
                     07    DD  0512    422          PUSHL   #7                      ; LENGTH OF STRING
                  7E   51    9A  0514    423          MOVZBL  R1,-(SP)
                                0517    424                  PFN_REFERENCE    -
                                0517    425          MOVAW   <@SDA$Ax_BLINK[R6],R1>,-
                                0517    426                  LONG_OPCODE=MOVAL,-
                                0517    427                  IMAGE=SDA
                                0531    428          GETMFM  (R1)
                   33 50    E9  053A    429 70$:     BLBC    R0,80$                  ; SKIP IF ERROR
                                053D    430                  PFN_REFERENCE    -
                                053D    431          MOVZWL  <R1,-(SP)>,-            ; BACKWARD LINK
                                053D    432                  LONG_OPCODE=MOVL,-
                                053D    433                  IMAGE=SDA
                                054D    434                  PFN_REFERENCE    -
                                054D    435          MOVAW   <@SDA$Ax_FLINK[R6],R1>,-
                                054D    436                  LONG_OPCODE=MOVAL,-
                                054D    437                  IMAGE=SDA
                                0567    438          GETMEM  (R1)
                   7B 50    E9  0570    439 80$:     BLBC    R0,90$                  ; SKIP IF ERROR
                                0573    440                  PFN_REFERENCE    -
                                0573    441          MOVZWL  <R1,-(SP)>,-            ; FORWARD LINK
                                0573    442                  LONG_OPCODE=MOVL,-
                                0573    443                  IMAGE=SDA
     51  0000004C'FF46    3E    0583    444          MOVAW   @SDA$AW_REFCNT[R6],R1
```

```
                        05B8    445          GETMEM  (R1)
                 57 50  E9  05C4    446          BLBC    R0,90$                  ; SKIP IF ERROR
               7E   51  3C  05C7    447          MOVZWL  R1,-(SP)                ; REFERENCE COUNT
     51  00000050'FF46  DE  05CA    448          MOVAL   @SDA$AL_BAK[R6],R1
                        05A2    449          GETMEM  (R1),-(SP)              ; BACKING STORE ADDRESS
                 3D 50  E9  05AE    450          BLBC    R0,90$                  ; SKIP IF ERROR
     51  00000054'FF46  DE  05B1    451          MOVAL   @SDA$AL_PTE[R6],R1
                        05C9    452          GETMEM  (R1),-(SP)              ; ADDRESS OF PAGE TABLE ENTRY
                 26 50  E9  05C5    453          BLBC    R0,90$                  ; SKIP IF ERROR
                    56  DD  05C8    454          PUSHL   R6                      ; PFN INDEX
                        05CA    455          PFN_DISP_IF_BIGPFN_THEN         ; If greater than 32 Mbytes, then use longwo
                        05D2
                        05D2                 ;This code executes if the PFN link arrays are longword arrays.
                        05D2    456                  PRINT   12,<!XL     !XL   !XL   !5UW    !XL !XL    !XB !AD     !XB !
                        05DF    457          PFN_DISP_ELSE                   ; Otherwise, use word format
                        05E1
                        05E1                 ;This code executes if the PFN link arrays are word arrays.
                        05E1    458                  PRINT   12,<!XW     !XL   !XL   !5UW    !XW !XW    !XB !AD     !XB !
                        05EE    459          PFN_DISP_ENDIF
                        05EE
                        05EE                 ;End of code that depends on size of PFN link arrays
                     04 05EE    460 90$:     RET
                        05EF    461
                        05EF    462          .DSABL  LSB
```

```
                                05EF    465          .SBTTL  DISPLAY_SPT_RANGE -- DISPLAY SYSTEM PAGE TABLE W/RANGE
                                05EF    466 ;---
                                05EF    467 ;
                                05EF    468 ;       DISPLAY_SPT_RANGE
                                05EF    469 ;
                                05EF    470 ;       THIS ROUTINE FORMATS THE ENTIRE CONTENTS OF THE SYSTEM
                                05EF    471 ;       PAGE TABLE, OR ANY SUBRANGE THEREOF.
                                05EF    472 ;
                                05EF    473 ;   INPUTS:
                                05EF    474 ;
                                05EF    475 ;       OPTIONS = OPTIONS FLAGS (RANGE OR LENGTH BITS RELEVANT)
                                05EF    476 ;       ESP     = START OF PAGE TABLE VA
                                05EF    477 ;                 (OR, IF LENGTH BIT SET)
                                05EF    478 ;       ESP     = SIZE OF PAGE TABLE VA
                                05EF    479 ;       ESP+4   = HIGH LIMIT OF PAGE TABLE VA
                                05EF    480 ;
                                05EF    481 ;   OUTPUTS:
                                05EF    482 ;
                                05EF    483 ;       NONE
                                05EF    484 ;
                                05EF    485 ;---
                                05EF    486
                        003C    05EF    487 .ENTRY  DISPLAY_SPT_RANGE,^M<R2,R3,R4,R5>
                                05F1    488
    50  00000000'EF     9E      05F1    489          MOVAB   OPTIONS, R0             ; POINT TO OPTIONS WORD
          52     60     D0      05F8    490          MOVL    (R0), R2
    51  00000000'EF     D0      05FB    491 3$:      MOVL    ESP, R1                 ; POINT TO EXPRESSION STACK
         07 52     03     E0      0602    492          BBS     #OPT$V_RANGE, R2, 10$   ; RANGE SPECIFIED
         11 52     04     E0      0606    493          BBS     #OPT$V_LENGTH, R2, 20$  ; LENGTH SPECIFIED
                   50     D4      060A    494 5$:      CLRL    R0                      ; SYNTAX ERROR
                          04      060C    495          RET
                                  060D    496
        54     04 A1     D0      060D    497 10$:     MOVL    4(R1),R4                ; R4 = LOWEST ADDRESS
      53     61     54     C3      0611    498          SUBL3   R4,(R1),R3              ; R3 = SIZE
        05 60     04     E2      0615    499          BBSS    #OPT$V_LENGTH,(R0),30$  ; SET A SINGLE BIT FOR RANGE
                  03     11      0619    500          BRB     30$
                                  061B    501
            53     61     7D      061B    502 20$:     MOVQ    (R1),R3                 ; R4 = LOWEST ADDRESS
                                  061E    503
        54   01FF 8F     AA      061E    504 30$:     BICW    #^X1FF,R4               ; ROUND DOWN
    53   000001FF 8F     C0      0623    505          ADDL2   #^X1FF,R3
     53    53     F7 8F     78      062A    506          ASHL    #-9,R3,R3               ; MAKE NUMBER OF ENTRIES
                  0A      11      062F    507          BRB     DISP                    ; JOIN COMMON CODE
```

```
                                0631   509          .SBTTL  DISPLAY_SPT     DISPLAY SYSTEM PAGE TABLE
                                0631   510 ;---
                                0631   511 ;
                                0631   512 ;       DISPLAY_SPT
                                0631   513 ;
                                0631   514 ;       THIS ROUTINE FORMATS THE ENTIRE CONTENTS OF THE SYSTEM
                                0631   515 ;       PAGE TABLE.
                                0631   516 ;
                                0631   517 ;   INPUTS:
                                0631   518 ;
                                0631   519 ;       NONE
                                0631   520 ;
                                0631   521 ;   OUTPUTS:
                                0631   522 ;
                                0631   523 ;       NONE
                                0631   524 ;
                                0631   525 ;---
                                0631   526          .ENABL  LSB
                                0631   527
                           003C 0631   528 .ENTRY  DISPLAY_SPT,^M<R2,R3,R4,R5>
                                0633   529
00 00000000'EF    04    E5    0633   530          BBCC    #OPT$V_LENGTH,OPTIONS,DISP ; CLEAR IT, IF SET BY /ALL
      FB97 CF     00    FB    063B   531 DISP:    CALLS   #0,INIT_PFN                ; SETUP TO READ PFN DATA
                                0640   532 ;
                                0640   533 ;       DISPLAY THE SYSTEM PAGE TABLE
                                0640   534 ;
7A 00000000'EF    02    E1    0640   535          BBC     #OPT$V_SYSTEM,OPTIONS,10$ ; BRANCH IF NOT SELECTED
                                0648   536          SUBHD   <System page table>
                                0655   537          SKIP    PAGE
                                065C   538          GETMEM  @MMG$GL_SYSPHD             ; ADDRESS OF SYSPHD
                  03 50         E8   0669   539          BLBS    R0,5$                      ;Branch if ok...else
                   00C1         31   066C   540          BRW     90$                        ;...Return
                     51         DD   066F   541 5$:      PUSHL   R1
   52    00000000'EF    9E    0671   542          MOVAB   BUFFER,R2
                                0678   543          GETMEM  PHD$L_POBR(R1),(R2),#8     ; GET VIRTUAL SBR,SLR
                  03 50         E8   0687   544          BLBS    R0,6$                      ; OKAY
                   00A3         31   068A   545          BRW     90$                        ; BRANCH IF ERROR
                     62         DD   068D   546 6$:      PUSHL   (R2)                       ; STARTING ADDRESS
   55    80000000    8F    D0    068F   547          MOVL    #^X80000000,R5             ; STARTING ADDRESS BEING MAPPED
09 00000000'EF    04    E0    0696   548          BBS     #OPT$V_LENGTH,OPTIONS,7$; IF RANGE NOT SPECIFIED...
53   04 A2    18    00    EF   069E   549          EXTZV   #PHD$V_POLR,#PHD$S_POLR,4(R2),R3   ; #ENTRIES
         54    55    D0   06A4   550          MOVL    R5,R4                      ; STARTING ADDRESS
      55 54    55    C3   06A7   551 7$:      SUBL3   R5,R4,R5                   ; OFFSET INTO AREA
      55 55     F9   8F   78   06AB   552          ASHL    #-7,R5,R5                  ; TURN INTO NUMBER OF ENTRIES TO SKIP
         6E    55    C0   06B0   553          ADDL    R5,(SP)                    ; UPDATE START ENTRY
         7E    53    7D   06B3   554          MOVQ    R3,-(SP)                   ; #ENTRIES,START ADDR
      38'AF    04    FB   06B6   555          CALLS   #4,B^DUMP_PTE              ; FORMAT PAGE TABLE
00 00000000'EF    04    E5    06BA   556          BBCC    #OPT$V_LENGTH,OPTIONS,10$ ; CLEAR IT OUT
                                06C2   557 ;
                                06C2   558 ;       DISPLAY THE GLOBAL PAGE TABLE
                                06C2   559 ;
66 00000000'EF    00    E1    06C2   560 10$:     BBC     #OPT$V_GLOBAL,OPTIONS,90$ ; BRANCH IF NOT SELECTED
                                06CA   561          SUBHD   <Global page table>
                                06D7   562          SKIP    PAGE
                                06DE   563          GETMEM  @MMG$GL_SYSPHD,-(SP)       ; ADDRESS OF PROCESS HEADER
                                06EE   564          GETMEM  @MMG$GL_GPTE,R2            ; ADDRESS OF FIRST GPTE
                  2F 50         E9   06FE   565          BLBC    R0,90$
```

```
                     0701    566          GETMEM  @MMG$GL_MAXGPTE          ; ADDRESS OF LAST+1 GPTE
                1F   50   E9 070E    567          BLBC    R0,90$
                     52   DD 0711    568          PUSHL   R2                       ; STARTING ADDRESS OF PAGE TABLE
06 00000000'EF       04   E0 0713    569          BBS     #OPT$V_LENGTH,OPTIONS,30$ ; IF RANGE NOT SPECIFIED...
        53   51      52   C3 071B    570          SUBL3   R2,R1,R3                 ; LENGTH OF PAGE TABLE
                     54   D4 071F    571          CLRL    R4                       ; FIRST PAGETABLE ENTRY
    55   54    F9    8F   78 0721    572 30$:     ASHL    #-7,R4,R5                ; TURN INTO NUMBER OF ENTRIES TO SKIP
               6E    55   C0 0726    573          ADDL    R5,(SP)                  ; UPDATE START ENTRY
               7E    53   7D 0729    574          MOVQ    R3,-(SP)
          38'AF    04   FB 072C    575          CALLS   #4,B^DUMP_PTE            ; FORMAT PAGE TABLE
                        0730    576 90$:
                        0730    577          STATUS  SUCCESS
                     04 0737    578          RET
                        0738    579
                        0738    580          .DSABL  LSB
```

```
                     0738   583          .SBTTL  DUMP_PTE -- FORMAT THE PAGE TABLE
                     0738   584 ;---
                     0738   585 ;
                     0738   586 ;       DUMP_PTE
                     073A   587 ;
                     0738   588 ;       THIS ROUTINE FORMATS AND PRINTS A SPECIFIED PAGE
                     0738   589 ;       TABLE GIVEN ITS ADDRESS AND LENGTH.  THE ADDRESS
                     0738   590 ;       OF THE PROCESS HEADER MUST ALSO BE GIVEN TO ACCESS
                     0738   591 ;       THE WORKING SET LIST.
                     0738   592 ;
                     0738   593 ;   INPUTS:
                     0738   594 ;
                     0738   595 ;        4(AP)  = ENTRIES OF PAGE TABLE TO DUMP
                     0738   596 ;        8(AP)  = STARTING ADDRESS OF REGION BEING MAPPED
                     0738   597 ;       12(AP)  = STARTING ADDRESS OF PAGE TABLE
                     0738   598 ;       16(AP)  = ADDRESS OF PROCESS HEADER
                     0738   599 ;
                     0738   600 ;       ASSUMES THAT INIT_PFN HAS ALREADY BEEN CALLED.
                     0738   601 ;
                     0738   602 ;   OUTPUTS:
                     0738   603 ;
                     0738   604 ;       THE PAGE TABLE IS FORMATTED AND PRINTED.
                     0738   605 ;
                     0738   606 ;---
                     0738   607
00000060             0738   608 SCRATCH_SIZE    = 24*4                          ; 24 LONGWORDS
                     0738   609
     07FC            0738   610 .ENTRY  DUMP_PTE,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
                     073A   611
                     073A   612          .ENABL  LSB
                     073A   613
        0A0B'CF   00 FB 073A   614          CALLS   #0,W^PTE_TITLE              ; PRINT SUB-HEADING LINE
00000000'EF  0A0B'CF 9E 073F   615          MOVAB   W^PTE_TITLE,HEADING_ROUTINE     ; SET HEADING ROUTINE
                  59 7C 0748   616          CLRQ    R9                          ; INITIALIZE STATE TO NORMAL
               04 AC D5 074A   617          TSTL    4(AP)                       ; CHECK IF ANY TO DUMP
                  01 14 074D   618          BGTR    10$                         ; BRANCH IF SO
                     04 074F   619          RET
                     0750   620 10$:
           5E    A0 AE 9E 0750   621          MOVAB   -SCRATCH_SIZE(SP),SP        ; RESERVE SPACE FOR FAO PARAMS
              52    5E D0 0754   622          MOVL    SP,R2                       ; R2 USED TO STORE PARAMS
                     0757   623 ;
                     0757   624 ;       FORMAT THE PAGE TABLE ENTRY
                     0757   625 ;
           82    08 AC D0 0757   626          MOVL    8(AP),(R2)+                 ; MAPPING ADDRESS
           82    0C AC D0 075B   627          MOVL    12(AP),(R2)+                ; VIRTUAL ADDRESS OF ENTRY
                     075F   628          TRYMEM  @12(AP)                     ; GET PAGE TABLE ENTRY
                  09 50 E8 0769   629          BLBS    R0,20$                      ; IF ENTRY FOUND
              50    01 D0 076C   630          MOVL    #1,R0
                 02D2 30 076F   631          BSBW    PTE_STATE                   ; SET STATE = 1 (INVALID MEMORY)
                 0271 31 0772   632          BRW     80$                         ; AND SKIP THIS ENTRY
                     0775   633 20$:
              53    51 D0 0775   634          MOVL    R1,R3                       ; SAVE PTE IN R3
                  09 12 0778   635          BNEQ    30$                         ; BRANCH IF NOT NULL PAGE
              50    02 D0 077A   636          MOVL    #2,R0
                 02C4 30 077D   637          BSBW    PTE_STATE                   ; SET STATE = 2 (NULL PAGES)
                 0263 31 0780   638          BRW     80$                         ; AND SKIP THIS ENTRY
                     0783   639 30$:
```

```
                50   D4  0783   640          CLRL    R0
              02BC   30  0785   641          BSBW    PTE_STATE                ; SET STATE TO NORMAL
                         0788   642
58   53   15   00   EF  0788   643          EXTZV   #PTE$V_PFN,#PTE$S_PFN,R3,R8  ; GET PFN IF PRESENT
             54   05   D0  078D   644          MOVL    #5,R4                    ; TYPE CODE FOR VALID
             82   53   D0  0790   645          MOVL    R3,(R2)+                 ; STORE PTE IN FAO LIST
                  18   18  0793   646          BGEQ    32$                      ; BRANCH IF NOT VALID
 00000000'EF      58   D1  0795   647          CMPL    R8,PHYS_PAGES            ; CHECK IF LEGAL
                  0B   18  079C   648          BGEQ    31$                      ; BRANCH IF INVALID PFN
 00000040'EF      58   D1  079E   649          CMPL    R8,SDA$GL_MAXPFN         ; CHECK IF WITHIN PFN DATABASE
                  1D   1A  07A5   650          BGTRU   36$                      ; BYPASS PFN LOOKUP IF SO
                  1E   11  07A7   651          BRB     40$                      ; GOOD PFN
                           07A9   652 31$:
                  54   D6  07A9   653          INCL    R4                       ; TYPE CODE FOR I/O PAGE
                  17   11  07AB   654          BRB     36$                      ; AND INDICATE INVALID PFN
                           07AD   655 32$:
54   53   01     16   EF  07AD   656          EXTZV   #PTE$V_TYP0,#1,R3,R4     ; BRING TYP0 AND TYP1
       03 53      1A   E1  07B2   657          BBC     #PTE$V_TYP1,R3,34$       ; TOGETHER
             54   02   C8  07B6   658          BISL    #2,R4                    ; SET HIGH ORDER BIT
                           07B9   659 34$:
             54   D5  07B9   660          TSTL    R4                       ; 0 = TRANSITION OR DZERO
                  07   12  07BB   661          BNEQ    36$                      ; BRANCH IF NOT
                  58   D5  07BD   662          TSTL    R8                       ; PFN SHOULD BE 0 FOR DZERO
                  06   12  07BF   663          BNEQ    40$                      ; BRANCH IF TRANSITION
             54   04   D0  07C1   664          MOVL    #4,R4                    ; TYPE CODE FOR DZERO
                           07C4   665 36$:
             58   01   CE  07C4   666          MNEGL   #1,R8                    ; INDICATE NO PFN
                           07C7   667 ;
                           07C7   668 ;        R3 = PTE LONGWORD
                           07C7   669 ;        R4 = PTE TYPE CODE
                           07C7   670 ;        R8 = PFN OR -1 IF NONE
                           07C7   671 ;
                           07C7   672 40$:
             54   05   C4  07C7   673          MULL2   #5,R4                    ; INDEX INTO TYPE TABLE
             82   05   D0  07CA   674          MOVL    #5,(R2)+                 ; LENGTH OF STRING
   82   F959 CF44   9E  07CD   675          MOVAB   TYPE_TABLE[R4],(R2)+     ; ADDRESS OF STRING
                  56   7C  07D3   676          CLRQ    R6                       ; ASSUME MODIFY/LOCK BITS OFF
      4C 53       1F   E1  07D5   677          BBC     #PTE$V_VALID,R3,45$      ; BRANCH IF NOT VALID
57   53   01     1A   EF  07D9   678          EXTZV   #PTE$V_MODIFY,#1,R3,R7   ; GET MODIFY BIT FROM PTE
        43 58     1F   E0  07DE   679          BBS     #31,R8,45$               ; BRANCH IF NO PFN
    3E 08 AC      1F   E0  07E2   680          BBS     #31,8(AP),45$            ; BRANCH IF SPT
                           07E7   681                  PFN_REFERENCE    -
                           07E7   682          MOVAW   <@SDA$Ax_WSLX[R8],R1>,-  ; ADDRESS OF WSLX FIELD
                           07E7   683                  LONG_OPCODE=MOVAL,-
                           07E7   684                  IMAGE=SDA
                           0801   685          GETMEM  (R1)                     ; GET LONGWORD
             18 50   E9  080A   686          BLBC    R0,45$                   ; IF NOT FOUND
             51   51   32  080D   687          CVTWL   R1,R1                    ; EXTEND FIELD
                  13   13  0810   688          BEQL    45$                      ; BRANCH IF NOT A WSL INDEX
      51     10 BC41   DE  0812   689          MOVAL   @16(AP)[R1],R1           ; ADDRESS OF WSL ENTRY
                           0817   690          GETMEM  (R1)                     ; GET WSL LONGWORD
56   51   01   05   EF  0820   691          EXTZV   #WSL$V_WSLOCK,#1,R1,R6   ; WSL LOCK BIT
                           0825   692 45$:
51   53   04   1B   EF  0825   693          EXTZV   #PTE$V_PROT,#PTE$S_PROT,R3,R1    ; GET PROTECTION CODE
             82   04   D0  082A   694          MOVL    #4,(R2)+                 ; LENGTH OF STRING
   82   F8B5 CF41   DE  082D   695          MOVAL   PROT_TABLE[R1],(R2)+     ; PAGE PROTECTION
             82   01   D0  0833   696          MOVL    #1,(R2)+                 ; SIZE OF MODIFY STRING
```

```
                82    F998 CF47    9E  0836    697          MOVAB   MODIFY_TABLE[R7],(R2)+  ; ADDRESS OF STRING
                   82    01    D0  083C    698          MOVL    #1,(R2)+                ; SIZE OF WSLOCK STRING
                82    F991 CF46    9E  083F    699          MOVAB   WSLOCK_TABLE[R6],(R2)+  ; ADDRESS OF STRING
            51     53    02    17    EF  0845    700          EXTZV   #PTE$V_OWN,#PTE$S_OWN,R3,R1   ; GET PAGE OWNER
                   82    01    D0  084A    701          MOVL    #1,(R2)+                ; LENGTH OF STRING
                82    F8D5 CF41    9E  084D    702          MOVAB   OWNER_TABLE[R1],(R2)+   ; ADDRESS OF STRING
                   24 58    1F    E1  0853    703          BBC     #31,R8,50$              ; BRANCH IF PFN VALID
                                        0857    704          $FAOL_S PTECTL1,LIST+RAB$W_RSZ,LINE_DESCR,-SCRATCH_SIZE(FP)
             00000000'EF    00    FB  0871    705          CALLS   #0,PUT_LINE             ; OUTPUT LINE
                      016B    31  0878    706          BRW     80$                     ; SKIP TO NEXT ENTRY
                                        087B    707 ;
                                        087B    708 ;       FORMAT THE PFN STRUCTURES ASSOCIATED WITH THIS PTE
                                        087B    709 ;
                                        087B    710 50$:
            51    00000048'FF48    9E  087B    711          MOVAB   @SDA$AB_TYPE[R8],R1
                                        0883    712          GETMEM  (R1)
                      5B 50    E9  088C    713          BLBC    R0,70$                  ; SKIP IF ERROR
             51    51    03    00    EF  088F    714          EXTZV   #PFN$V_PAGTYP,#PFN$S_PAGTYP,R1,R1   ; GET PAGE TYPE
                   82    07    D0  0894    715          MOVL    #7,(R2)+                ; LENGTH OF STRING
                82    F8F7 CF41    7E  0897    716          MOVAQ   PAGTYP_TABLE[R1],(R2)+  ; ADDRESS OF STRING
            51    00000044'FF48    9E  089D    717          MOVAB   @SDA$AB_STATE[R8],R1
                                        08A5    718          GETMEM  (R1)
                      39 50    E9  08AE    719          BLBC    R0,70$                  ; SKIP IF ERROR
             51    51    03    00    EF  08B1    720          EXTZV   #PFN$V_LOC,#PFN$S_LOC,R1,R1     ; GET PAGE LOCATION
                   82    07    D0  08B6    721          MOVL    #7,(R2)+                ; LENGTH OF STRING
                82    F895 CF41    7E  08B9    722          MOVAQ   LOC_TABLE[R1],(R2)+     ; ADDRESS OF STRING
            51    00000044'FF48    9E  08BF    723          MOVAB   @SDA$AB_STATE[R8],R1
                                        08C7    724          GETMEM  (R1)                    ; GET STATE FIELD
                      17 50    E9  08D0    725          BLBC    R0,70$                  ; SKIP IF ERROR
                   82    51    9A  08D3    726          MOVZBL  R1,(R2)+
            51    00000048'FF48    9E  08D6    727          MOVAB   @SDA$AB_TYPE[R8],R1
                                        08DE    728          GETMEM  (R1)                    ; GET TYPE FIELD
                      03 50    E8  08E7    729          BLBS    R0,71$
                       00F9    31  08EA    730 70$:     BRW     80$                     ; SKIP IF ERROR
                   82    51    9A  08ED    731 71$:     MOVZBL  R1,(R2)+
            51    0000004C'FF48    3E  08F0    732          MOVAW   @SDA$AW_REFCNT[R8],R1
                                        08F8    733          GETMEM  (R1)                    ; COUNT OF PAGE REFERENCES
                      E6 50    E9  0901    734          BLBC    R0,70$                  ; SKIP IF ERROR
                   82    51    3C  0904    735          MOVZWL  R1,(R2)+
            51    00000050'FF48    DE  0907    736          MOVAL   @SDA$AL_BAK[R8],R1
                                        090F    737          GETMEM  (R1)                    ; BACKING STORE ADDRESS
                      CF 50    E9  0918    738          BLBC    R0,70$                  ; SKIP IF ERROR
                   82    51    D0  091B    739          MOVL    R1,(R2)+
            51    00000054'FF48    DE  091E    740          MOVAL   @SDA$AL_PTE[R8],R1
                                        0926    741          GETMEM  (R1)                    ; ADDRESS OF PTE
                      B8 50    E9  092F    742          BLBC    R0,70$                  ; SKIP IF ERROR
                   82    51    D0  0932    743          MOVL    R1,(R2)+
                                        0935    744                  PFN_REFERENCE    -
                                        0935    745          MOVAW   <@SDA$Ax_FLINK[R8],R1>,-
                                        0935    746                  LONG_OPCODE=MOVAL,-
                                        0935    747                  IMAGE=SDA
                                        094F    748          GETMEM  (R1)                    ; FORWARD PAGE LIST LINK
                      8F 50    E9  0958    749          BLBC    R0,70$                  ; SKIP IF ERROR
                                        095B    750                  PFN_REFERENCE    -
                                        095B    751          MOVZWL  <R1,(R2)+>,-
                                        095B    752                  LONG_OPCODE=MOVL,-
                                        095B    753                  IMAGE=SDA
```

```
                                 096B   754                     PFN_REFERENCE   -
                                 096B   755             MOVAW   <@SDA$Ax_BLINK[R8],R1>,-
                                 096B   756                     LONG_OPCODE=MOVAL,-
                                 096B   757                     IMAGE=SDA
                                 0985   758             GETMEM  (R1)                    ; BACKWARD PAGE LIST LINK
                      55 50   E9 098E   759             BLBC    R0,80$                  ; SKIP IF ERROR
                                 0991   760                     PFN_REFERENCE   -
                                 0991   761             MOVZWL  <R1,(R2)+>,-
                                 0991   762                     LONG_OPCODE=MOVL,-
                                 0991   763                     IMAGE=SDA
                                 09A1   764                                             ; For larger than 32 Mbytes, use longword fo
                                 09A1   765             PFN_DISP_IF_BIGPFN_THEN         END_BIGPFN_CODE=74$
                                 09A9
                                 09A9                   ;This code executes if the PFN link arrays are longword arrays.
                                 09A9   766                     $FAOL_S PTECTL2_LONG,LIST+RAB$W_RSZ,LINE_DESCR,-SCRATCH_SIZE(FP)
                                 09C3   767                                             ; Otherwise, use word format
                                 09C3   768             PFN_DISP_ELSE                   ELSE_CODE=74$ , COMMON_CODE=77$
                                 09C5
                                 09C5                   ;This code executes if the PFN link arrays are word arrays.
                                 09C5   769                     $FAOL_S PTECTL2_WORD,LIST+RAB$W_RSZ,LINE_DESCR,-SCRATCH_SIZE(FP)
                                 09DF   770             PFN_DISP_ENDIF          COMMON_CODE=77$
                                 09DF
                                 09DF                   ;End of code that depends on size of PFN link arrays
          00000000'EF    00   FB 09DF   771             CALLS   #0,PUT_LINE             ; OUTPUT LINE
                                 09E6   772 ;
                                 09E6   773 ;           SKIP TO NEXT PAGE TABLE ENTRY
                                 09E6   774 ;
                                 09E6   775 80$:
                5E    60 AE    9E 09E6   776             MOVAB   SCRATCH_SIZE(SP),SP     ; DEALLOCATE FAO SPACE
                0C AC    04    C0 09EA   777             ADDL2   #4,12(AP)               ; NEXT PTE
          08 AC    00000200 8F   C0 09EE   778             ADDL2   #512,8(AP)              ; INCREMENT MAPPING ADDRESS
                      04 AC    D7 09F6   779             DECL    4(AP)                   ; DECREMENT REPEAT COUNT
                         03    15 09F9   780             BLEQ    90$                     ; EXIT IF DONE
                       FD52    31 09FB   781             BRW     10$
                                 09FE   782 90$:
                         50    D4 09FE   783             CLRL    R0
                       0041    30 0A00   784             BSBW    PTE_STATE               ; TERMINATE CURRENT STATE
                                 0A03   785             STATUS  SUCCESS
                         04    0A0A   786             RET
                                 0A0B   787
                                 0A0B   788             .DSABL  LSB
                                 0A0B   789
                                 0A0B   790 ;
                                 0A0B   791 ;           SUBROUTINE TO PRINT THE SUB-HEADING LINE
                                 0A0B   792 ;
                                 0A0B   793
                                 0A0B   794             .ENABLE         LOCAL_BLOCK
                                 0A0B   795
                                 0A0B   796 PTE_TITLE:
                        0000     0A0B   797             .WORD   0
                                 0A0D   798
                                 0A0D   799             SKIP    1
                                 0A16   800             PFN_DISP_IF_BIGPFN_THEN         ; For larger than 32 Mbytes, use longword fo
                                 0A1E
                                 0A1E                   ;This code executes if the PFN link arrays are longword arrays.
                                 0A1E   801                     PRINT   0,-
                                 0A1E   802                     <!_ ADDRESS       SVAPTE    PTE     TYPE  PROT  BITS PAGTYP    LOC S
```

```
                0A2B    803         PFN_DISP_ELSE                       ; Otherwise, use word format
                0A2D
                0A2D                ;This code executes if the PFN link arrays are word arrays.
                0A2D    804                 PRINT   0,-
                0A2D    805                 <!  ADDRESS       SVAPTE    PTE     TYPE  PROT  BITS PAGTYP    LOC S
                0A3A    806         PFN_DISP_ENDIF
                0A3A
                0A3A                ;End of code that depends on size of PFN link arrays
                0A3A    807         SKIP    1
          04    0A43    808         RET
                0A44    809
                0A44    810         .DISABLE        LOCAL_BLOCK
```

```
                     0A44   813          .SBTTL  PTE_STATE       SET STATE OF PTE DISPLAY
                     0A44   814 ;---
                     0A44   815 ;
                     0A44   816 ;       PTE_STATE
                     0A44   817 ;
                     0A44   818 ;       SET STATE OF RUNNING SCAN OF PAGE TABLE AND PRINT ANY
                     0A44   819 ;       STATUS MESSAGES FROM THE PREVIOUS STATE.
                     0A44   820 ;
                     0A44   821 ;   INPUTS:
                     0A44   822 ;
                     0A44   823 ;       R0  =   REQUESTED NEW STATE
                     0A44   824 ;       R9  =   CURRENT STATE
                     0A44   825 ;       R10 =   REPITITION COUNT IN SAME STATE
                     0A44   826 ;
                     0A44   827 ;   OUTPUTS:
                     0A44   828 ;
                     0A44   829 ;       R9  =   NEW STATE
                     0A44   830 ;       R10 =   UPDATED REPITITION COUNT
                     0A44   831 ;
                     0A44   832 ;---
                     0A44   833
                     0A44   834          .ENABL  LSB
                     0A44   835
                     0A44   836 PTE_STATE:
 59   50   D1   0A44   837          CMPL    R0,R9                   ; CHECK IF ALREADY IN STATE
      03   12   0A47   838          BNEQ    10$                     ; BRANCH IF NOT
      5A   D6   0A49   839          INCL    R10                     ; INCREMENT REPITITION COUNT
           05   0A4B   840          RSB
                0A4C   841 10$:
      50   DD   0A4C   842          PUSHL   R0                      ; SAVE NEW STATE
 01   59   D1   0A4E   843          CMPL    R9,#1                   ; CHECK IF BYPASSING BAD MEMORY
      23   12   0A51   844          BNEQ    20$                     ; BRANCH IF NOT
                0A53   845          SKIP    1
      5A   DD   0A5C   846          PUSHL   R10
                0A5E   847          PRINT   1,<!_-------- !UL ENTRIES NOT IN MEMORY>
                0A6B   848          SKIP    1
      26   11   0A74   849          BRB     80$
                0A76   850 20$:
 02   59   D1   0A76   851          CMPL    R9,#2                   ; CHECK IF SKIPPING NULL PAGES
      21   12   0A79   852          BNEQ    80$                     ; BRANCH IF NOT
                0A7B   853          SKIP    1
      5A   DD   0A84   854          PUSHL   R10
                0A86   855          PRINT   1,<!_-------- !UL NULL PAGE!%S>
                0A93   856          SKIP    1
                0A9C   857 80$:
      59 8ED0   0A9C   858          POPL    R9                      ; SET NEW STATE
 5A   01   D0   0A9F   859          MOVL    #1,R10                  ; INITIALIZE REPITITION COUNTER
           05   0AA2   860          RSB
                0AA3   861
                0AA3   862          .DSABL  LSB
```

```
                0AA3    864
                0AA3    865             .END
```

```
ARGS                            = 00000001
BUFFER                            00000000 R      02
DISP                              0000063B R      03
DISPLAY_PFN                       0000028E RG     03
DISPLAY_SPT                       00000631 RG     03
DISPLAY_SPT_RANGE                 000005EF RG     03
DUMP_PTE                          00000738 RG     03
ESP                               ********    X   03
GETMEM                            ********    X   03
HEADING_ROUTINE                   ********    X   03
INIT_PFN                          000001D7 RG     03
LINE_DESCR                        ********    X   03
LIST                              ********    X   03
LOC_TABLE                         00000153 R      03
MMG$GL_GPTE                       ********    X   03
MMG$GL_MAXGPTE                    ********    X   03
MMG$GL_MAXPFN                     ********    X   03
MMG$GL_SYSPHD                     ********    X   03
MMG$GW_BIGPFN                   = 00000042 R      02
MODIFY_TABLE                      000001D3 R      03
MSG$_SUCCESS                      ********    X   03
NEW_PAGE                          ********    X   03
OPT$V_BAD                       = 00000002
OPT$V_FREE                      = 00000000
OPT$V_GLOBAL                    = 00000000
OPT$V_LENGTH                    = 00000004
OPT$V_MODIFIED                  = 00000001
OPT$V_RANGE                     = 00000003
OPT$V_SINGLEPFN                 = 00000004
OPT$V_SYSTEM                    = 00000002
OPT$V_WHOLEPFN                  = 00000003
OPTIONS                           ********    X   03
OWNER_TABLE                       00000127 R      03
PAGTYP_TABLE                      00000193 R      03
PFN$AB_STATE                      ********    X   03
PFN$AB_TYPE                       ********    X   03
PFN$AL_BAK                        ********    X   03
PFN$AL_HEAD                       ********    X   03
PFN$AL_HILIMIT                    ********    X   03
PFN$AL_LOLIMIT                    ********    X   03
PFN$AL_PTE                        ********    X   03
PFN$AW_REFCNT                     ********    X   03
PFN$AX_BLINK                      ********    X   03
PFN$AX_FLINK                      ********    X   03
PFN$AX_WSLX                       ********    X   03
PFN$S_LOC                       = 00000003
PFN$S_PAGTYP                    = 00000003
PFN$V_LOC                       = 00000000
PFN$V_PAGTYP                    = 00000000
PFN_TITLE                         0000047C R      03
PHD$L_POBR                      = 000000C8
PHD$S_POLR                      = 00000018
PHD$V_POLR                      = 00000000
PHYS_PAGES                        ********    X   03
PRINT                             ********    X   03
PROT_TABLE                        000000E7 R      03
PTE$S_OWN                       = 00000002
PTE$S_PFN                       = 00000015
PTE$S_PROT                      = 00000004
PTE$V_MODIFY                    = 0000001A
PTE$V_OWN                       = 00000017
PTE$V_PFN                       = 00000000
PTE$V_PROT                      = 0000001B
PTE$V_TYP0                      = 00000016
PTE$V_TYP1                      = 0000001A
PTE$V_VALID                     = 0000001F
PTECTL1                           00000000 R      03
PTECTL2_LONG                      0000008B R      03
PTECTL2_WORD                      0000002F R      03
PTE_STATE                         00000A44 R      03
PTE_TITLE                         00000A0B R      03
PUT_LINE                          ********    X   03
RAB$W_RSZ                         ********    X   03
REQMEM                            ********    X   03
SCH$GL_FREECNT                    ********    X   03
SCRATCH_SIZE                    = 00000060
SDA$AB_STATE                      00000044 R      02
SDA$AB_TYPE                       00000048 R      02
SDA$AL_BAK                        00000050 R      02
SDA$AL_PTE                        00000054 R      02
SDA$AW_REFCNT                     0000004C R      02
SDA$AX_BLINK                      0000005C R      02
SDA$AX_FLINK                      00000058 R      02
SDA$AX_WSLX                       00000060 R      02
SDA$GL_MAXPFN                     00000040 R      02
SET_HEADING                       ********    X   03
SHOW_PFN                          000004CF R      03
SHOW_PFN_LIST                     000003A3 R      03
SKIP_LINES                        ********    X   03
SYS$FAOL                          ********   GX   03
TPA$L_NUMBER                    = 0000001C
TRYMEM                            ********    X   03
TYPE_TABLE                        0000012B R      03
WSL$V_WSLOCK                    = 00000005
WSLOCK_TABLE                      000001D5 R      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
SDADATA	00000064 (100.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE
MMG	00000AA3 (2723.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE
LITERALS	00000462 (1122.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.03	00:00:01.48
Command processing	112	00:00:00.43	00:00:03.01
Pass 1	353	00:00:07.85	00:00:26.11
Symbol table sort	0	00:00:00.97	00:00:05.81
Pass 2	163	00:00:02.21	00:00:09.83
Symbol table output	11	00:00:00.06	00:00:00.33
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	674	00:00:11.57	00:00:46.59

The working set limit was 1500 pages.
76303 bytes (150 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 875 non-local and 96 local symbols.
865 source lines were read in Pass 1, producing 41 object records in Pass 2.
31 pages of virtual memory were used to define 29 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SDA.OBJ]SDALIB.MLB;1	9
_255DUA28:[SYS.OBJ]LIB.MLB;1	9
_255DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	26

1055 GETS were required to define 26 macros.

There were no errors, warnings or information messages.

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

HANDLER
LIS

MAPPING
LIS

DUMP
LIS

MAIN
LIS

EXAMPSL
LIS

MMG
LIS

INDEX
LIS

LOCK
LIS

PARSE
LIS