



```

SSSSSSSS  YY  YY  SSSSSSSS  GGGGGGGG  EEEEEEEEE  TTTTTTTTTT  PPPPPPPP  TTTTTTTTTT  IIIIII
SSSSSSSS  YY  YY  SSSSSSSS  GGGGGGGG  EEEEEEEEE  TTTTTTTTTT  PPPPPPPP  TTTTTTTTTT  IIIIII
SS  YY  YY  SS  GG  EE  TT  PP  PP  TT  II
SS  YY  YY  SS  GG  EE  TT  PP  PP  TT  II
SS  YY  YY  SS  GG  EE  TT  PP  PP  TT  II
SS  YY  YY  SS  GG  EE  TT  PP  PP  TT  II
SSSSSSS  YY  YY  SSSSSS  GG  GGGGGG  EE  TT  PP  TT  II
SSSSSSS  YY  YY  SSSSSS  GG  GGGGGG  EE  TT  PP  TT  II
SS  YY  YY  SS  GG  GGGGGG  EE  TT  PP  TT  II
SS  YY  YY  SS  GG  GGGGGG  EE  TT  PP  TT  II
SS  YY  YY  SS  GG  GGGGGG  EE  TT  PP  TT  II
SSSSSSSS  YY  SSSSSSSS  GGGGGG  EEEEEEEEE  TT  PP  TT  IIIIII
SSSSSSSS  YY  SSSSSSSS  GGGGGG  EEEEEEEEE  TT  PP  TT  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

```

....  
....  
....  
....

(2) 43  
(3) 72  
(4) 150

DECLARATIONS  
GETPTI - Get Page Table Information  
GETPTIPAG - Get Page Table Info for Single Page

```
0000 1 .TITLE SYSGETPTI - Get Page Table Information System Service
0000 2 .IDENT 'V04-000'
0000 3
0000 4 :*****
0000 5 :
0000 6 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 7 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 8 :* ALL RIGHTS RESERVED. *
0000 9 :
0000 10 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 11 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 12 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 13 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 14 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 15 :* TRANSFERRED. *
0000 16 :
0000 17 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 18 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 19 :* CORPORATION. *
0000 20 :
0000 21 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 22 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 23 :
0000 24 :
0000 25 :*****
0000 26 :
0000 27 :++
0000 28 : FACILITY:
0000 29 :
0000 30 : ABSTRACT:
0000 31 :
0000 32 : ENVIRONMENT:
0000 33 :
0000 34 : AUTHOR: PETER H. LIPMAN , CREATION DATE: 24-APR-78
0000 35 :
0000 36 : MODIFIED BY:
0000 37 :
0000 38 : V03-001 WMC0001 Wayne Cardoza 02-Mar-1983
0000 39 : MMG$CRECOM1 has gone away
0000 40 :
0000 41 :--
```

```
0000 43      .SBTTL  DECLARATIONS
0000 44      :
0000 45      : INCLUDE FILES:
0000 46      :
0000 47      $MMGDEF      ; Offsets from FP into scratch area
0000 48      $PRDEF      ; PROCESSOR REGISTER DEFINITIONS
0000 49      $PTEDEF     ; PAGE TABLE ENTRY DEFINITIONS
0000 50      $SSDEF      ; SYSTEM STATUS CODE DEFINITIONS
0000 51      :
0000 52      : EXTERNAL SYMBOLS:
0000 53      :
0000 54      :
0000 55      : MACROS:
0000 56      :
0000 57      :
0000 58      : EQUATED SYMBOLS:
0000 59      :
0000 60      : OFFSET FROM AP
0000 61      :
00000004 0000 62      INADR      = 4      ;Offset to input range
00000008 0000 63      RETADR     = 8      ;Offset to return range
0000000C 0000 64      ACMODE     = 12     ;Access mode
00000010 0000 65      MASK       = 16     ;Mask of PTI attributes
00000014 0000 66      PAGCNT     = 20     ;Minimum number of pages to return
0000 67      :as having the specified attributes
0000 68      :
0000 69      : OWN STORAGE:
0000 70      :
```

```

0000 72      .SBTTL  GETPTI - Get Page Table Information
0000 73      :++
0000 74      : FUNCTIONAL DESCRIPTION:
0000 75      :
0000 76      : CALLING SEQUENCE:
0000 77      :
0000 78      :     CALLG  ARGLIST,@#SYSS$GETPTI
0000 79      :
0000 80      :
0000 81      : INPUT PARAMETERS:
0000 82      :
0000 83      :     INADR(AP) = Address of 2 long words the 1st of which specifies
0000 84      :                 the starting virtual address, the 2nd specifies the ending
0000 85      :                 virtual address (inclusive) of the pages to operate on.
0000 86      :     RETADR(AP) = Address of a 2 longword array into which is returned
0000 87      :                 the starting and ending virtual addresses (inclusive)
0000 88      :                 of the pages operated on.
0000 89      :     ACMODE(AP) = The access mode (maximized with calling mode)
0000 90      :                 against which the page ownership is checked.
0000 91      :     MASK(AP)   = Mask of Page Table Information control bits
0000 92      :                 bit 0 - Pages which are demand zero
0000 93      :     PAGCNT(AP) = Minimum page count.  If the next range of pages which
0000 94      :                 satisfies one or more of the PTI criteria contains
0000 95      :                 fewer pages than PAGCNT specifies, then the range
0000 96      :                 does not satisfy the criteria and another range
0000 97      :                 is sought.
0000 98      :
0000 99      : IMPLICIT INPUTS:
0000 100     :
0000 101     :     NONE
0000 102     :
0000 103     : OUTPUT PARAMETERS:
0000 104     :
0000 105     :     R0 = System Status Code
0000 106     :
0000 107     : IMPLICIT OUTPUTS:
0000 108     :
0000 109     :     NONE
0000 110     :
0000 111     : COMPLETION CODES:
0000 112     :
0000 113     :     SSS_NORMAL           ;Successful Completion
0000 114     :     SSS_ACCVIO         ;Access Violation
0000 115     :     SSS_PAGOWNVIO      ;Page Owner Violation
0000 116     :
0000 117     : SIDE EFFECTS:
0000 118     :
0000 119     :     NONE
0000 120     :
0000 121     : --
0000 122     :
0000 123     : *****
0000 124     :
0000 125     : ***** THE FOLLOWING CODE MAY BE PAGED *****
0000 126     :
0000 127     :     .PSECT  Y$EXEPAGED
0000 128     :

```

```

0000 129 : *****
0000 130 :
07FC 0000 131 : .ENTRY EXE$GETPTI,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
0002 132 :
56 00000000'EF DE 0002 133 : MOVAL L^MMG$GETPTIPAG,R6 ;Per-page subroutine
58 10 AC D0 0009 134 : MOVL MASK(AP),R8 ;Type of information desired
59 01 CE 000D 135 : MNEGL #1,R9 ;No pages found yet
SA 14 AC 01 C3 0010 136 : SUBL3 #1,PAGCNT(AP),R10 ;Minimum page count - 1
02 18 0015 137 : BGEQ 10$ ;Branch if specified
5A 5A D4 0017 138 : CLRL R10 ;Default to 1 page
SE 1C C2 0019 139 10$: SUBL S^#-MMG$C LENGTH,SP ;Reserve scratch area
FFE1' 30 001C 140 : BSBW MMG$INADRINI ;Initialize return address arrays & scratch a
10 50 E9 001F 141 : BLBC R0,20$
52 54 7D 0022 142 : MOVQ R4,R2 ;R2 = Start of range, R3 = end
FFD8' 30 0025 143 : BSBW MMG$CREDEL ;Common Create code
50 DD 0028 144 : PUSHL R0 ;Save status
FFD3' 30 002A 145 : BSBW MMG$RETRANGE ;Return affected address range
02 50 E9 002D 146 : BLBC R0,20$ ;Use this bad status rather than CREDEL
50 BA 0030 147 : POPR R0
04 0032 148 20$: RET

```

```
0033 150 .SBTTL GETPTIPAG - Get Page Table Info for Single Page
0033 151
0033 152 :++
0033 153 : FUNCTIONAL DESCRIPTION:
0033 154 :
0033 155 :
0033 156 : CALLING SEQUENCE:
0033 157 :
0033 158 :     BSBW     MMG$GETPTIPAG
0033 159 :
0033 160 :
0033 161 : INPUT PARAMETERS:
0033 162 :
0033 163 :     R0 = Access Mode for page ownership check
0033 164 :     R2 = Virtual Address
0033 165 :     R4 = Current PCB address
0033 166 :     R5 = Process Header Address - P1 or System Space
0033 167 :     R6 = Count - 1 of pages to be processed including this one
0033 168 :     R7 = +^X200 if going forward in the address space
0033 169 :     = -^X200 if going backwards in the address space
0033 170 :     R8 = Page table information mask, specifying the criteria
0033 171 :         against which the pages are tested.
0033 172 :     R9 = count - 1 of pages found so far that satisfy the criteria
0033 173 :     R10 = threshold page count - 1. Once a run of pages is found
0033 174 :         that meets the criteria, a further check is made to see if
0033 175 :         the range of pages is large enough.
0033 176 :
0033 177 :     IPL = ASTDEL
0033 178 :
0033 179 : IMPLICIT INPUTS:
0033 180 :     NONE
0033 181 :
0033 182 : OUTPUT PARAMETERS:
0033 183 :
0033 184 :     R0 = Status Code
0033 185 :     R2 = Preserved
0033 186 :
0033 187 : IMPLICIT OUTPUTS:
0033 188 :     NONE
0033 189 :
0033 190 : COMPLETION CODES:
0033 191 :
0033 192 :     SSS_NORMAL                ;Successful Completion
0033 193 :     SSS_PAGOWNVIO            ;Page Owner Violation
0033 194 :     SSS_LENvio              ;Length Violation
0033 195 :     SSS_ACCVIO              ;Access Violation
0033 196 :
0033 197 : SIDE EFFECTS:
0033 198 :
0033 199 :     NONE
0033 200 :
0033 201 : --
0033 202 :
0033 203 : *****
0033 204 :
0033 205 : ***** THE FOLLOWING CODE MUST BE RESIDENT *****
0033 206 :
```



```

00000000 207 .PSECT $MMGCODE
0000 208 :
0000 209 : *****
0000 210 :
0000 211 MMG$GETPTIPAG::
0000 212 SAVIPL ;Save caller's IPL
50 DD 0003 213 PUSHL R0 ;Save access mode
FFF8' 30 0005 214 BSBW MMG$PTEREF ;Reference PTE, return SVAPTE
0008 215 ;return at IPL=SYNCH
69 50 E9 0008 216 BLBC R0,90$ ;Branch if Length Violation
50 0C 3C 000B 217 MOVZWL #SS$_ACCVIO,R0 ;In case this page is not accessible
51 63 D0 000E 218 MOVL (R3),R1 ;Fetch the page table entry
50 61 13 0011 219 BEQL 90$ ;Branch if deleted, access violation
50 6E D0 0013 220 MOVL (SP),R0 ;Access Mode
FFE7' 30 0016 221 BSBW MMG$PAGETYPE ;Check access, return page type
58 50 E9 0019 222 BLBC R0,90$ ;Branch if Page Owner Violation
50 63 16 00 223 EXTZV #PTESV_PGFLVB,#PTESS_PGFLVB,(R3),R0 ;PFN, PGFLVB, GPTX
7E 57 7D 0021 224 MOVQ R7,-(SP) ;Push R8, reserve a scratch location
6E 58 20 00 EA 0024 225 20$: FFS #0,#32,R8,(SP) ;Convert next bit to its bit number
09 13 C029 226 BEQL 40$ ;Branch if no more bits
00 58 6E E5 002B 227 BBCC (SP),R8,30$ ;Shut off the bit
49 10 002F 228 30$: BSBB 100$ ;See if this page has desired attribute
F0 50 E9 0031 229 BLBC R0,20$ ;Branch if not, check next attribute
0102 8F BA 0034 230 40$: POPR #*M<R1,R8> ;Clean off scratch loc, restore R8
59 50 C0 0038 231 ADDL R0,R9 ;If match found, count it
OF 19 003B 232 BLSS 60$ ;Branch if haven't found first page
08 50 E9 003D 233 BLBC R0,50$ ;Branch if end of "run"
0A 12 0040 234 BNEQ 60$ ;Branch if not first page of "run"
EC AD 52 D0 0042 235 MOVL R2,B*MMG$_SVSTARTVA(FP) ;Save address of first page
04 11 0046 236 BRB 60$
00 56 1F E2 0048 237 50$: BBSS #31,R6,60$ ;Indicate end of run
50 01 3C 004C 238 60$: MOVZWL #SS$_NORMAL,R0 ;Set success indication
56 D5 004F 239 TSTL R6 ;Check for end of range or end of "run"
21 14 0051 240 BGTR 90$ ;Branch if not
59 D5 0053 241 TSTL R9 ;At end range or end "run"
1A 19 0055 242 BLSS 80$ ;Branch if no "run" found
5A 59 D1 0057 243 CMLP R9,R10 ;Is the "run" large enough?
0B 19 005A 244 BLSS 70$ ;Branch if not, start over
005C 245 :
005C 246 : Found a large enough range, return it
005C 247 :
56 D5 005C 248 TSTL R6 ;At end of range?
14 13 005E 249 BEQL 90$ ;Branch if yes, this VA is in the range
52 56 D4 0060 250 CLRL R6 ;Force end of range
57 C2 0062 251 SUBL R7,R2 ;and back off 1 page
0D 11 0065 252 BRB 90$
0067 253 :
0067 254 : This "run" was too small, start looking for another
0067 255 :
59 01 CE 0067 256 70$: MNEGL #1,R9 ;No pages in the "run"
56 56 1E 00 257 EXTZV #0,#30,R6,R6 ;Restore count - 1 of pages left in range
03 12 006F 258 BNEQ 90$ ;Branch if still some to do
F4 AD D4 0071 259 80$: CLRL B*MMG$_SAVRETADR(FP) ;Force null return range
02 BA 0074 260 90$: POPR #*M<R1> ;Clean off saved access mode
0076 261 ENBINT ;Restore called IPL
05 0079 262 RSB ;and return
007A 263 :

```

007A 264 : Subroutine to determine whether the specified page is of the type  
007A 265 : described by the specified type code.

007A 266 :  
007A 267 : INPUTS:

007A 268 :  
007A 269 : R0 = PFN, PGFLVB, GPTX, OR SECX WITH SECTION BITS  
007A 270 : R1 = Page type  
007A 271 : = 0 - transition or demand zero  
007A 272 : = 1 - page file  
007A 273 : = 2 - global  
007A 274 : = 3 - section  
007A 275 : = 4 - valid  
007A 276 : R2 = Virtual address  
007A 277 : R3 = System Virtual Address of page table entry  
007A 278 : R4 = Current PCB address  
007A 279 : R5 = Process Header address  
007A 280 : 4(SP) = Page Table Information Code to CASE on

007A 281 :  
007A 282 : OUTPUTS:

007A 283 :  
007A 284 : R0 = 0 if page is not of the specified type  
007A 285 : = 1 if page is of the specified type  
007A 286 : R2 preserved

007A 287 :  
007A 288 : 100\$: CASE 4(SP), <-  
007A 289 : INF\_DZRO - ;0 = page is demand zero  
007A 290 : >

0081 291 :  
0081 292 : Page is not of the specified type

50 D4 0081 293 :  
0081 294 : INF\_NO: CLRL R0  
05 0083 295 : RSB

0084 296 :  
0084 297 : See if page is demand zero

0084 298 :  
0084 299 : INF\_DZRO:

0084 300 : CASE R1, <-  
0084 301 : 20\$, - ;Transition or Demand Zero  
0084 302 : INF\_NO, - ;Page file  
0084 303 : 40\$, - ;Global  
0084 304 : 60\$, - ;Section  
0084 305 : >, TYPE=B  
EF 11 0090 306 : BRB INF\_NO ;Valid

0092 307 :  
0092 308 : Transition or demand zero page

50 D5 0092 309 :  
0092 310 : 20\$: TSTL R0 ;Zero PFN means demand zero  
EB 12 0094 311 : BNEQ INF\_NO ;Branch if not dzro  
50 D6 0096 312 : INCL R0 ;Return 1 for dzro page  
05 0098 313 : RSB

0099 314 :  
0099 315 : Global page, see if it is a demand zero section

0099 316 :  
0099 317 : 40\$: BICL3 #^C<PTESM\_VALID !- ;Get valid,  
009A 318 : PTESM\_TYPT ! PTESM\_TYPO !- ;type bits,  
009A 319 : PTESM\_PGFLVB>, - ;and section bits  
50 0000'DF40 7B800000 8F 009A 320 : @W^M^3\$GL\_GPTBASE[R0],R0 ;from global PTE

51	50	EA	DB	19	00A4	321	BLSS	INF_NO		;Branch if valid
			8F	78	00A6	322	ASHL	#-PTESV_TYPO,R0,R1		;Transition?
			D4	13	00AB	323	BEQL	INF_NO		;Branch if yes
					00AD	324				
					00AD	325	ASSUME	<PTESV_TYP1-PTESV_TYPO>	LE 7	
	51	11		8C	00AD	326	XORB	#<PTESM_TYP1 ! PTESM_TYPO>	@-PTESV_TYPO,R1	;Section address?
		CF		12	00B0	327	BNEQ	INF_NO		;Branch if not
					00B2	328				
					00B2	329				
					00B2	330				
					00B2	331				
CB	50	11	E1	00B2	331	60\$:	BBC	#PTESV_DZRO,R0,INF_NO		;Branch if not demand zero section
	50	01	D0	00B6	332		MOVL	#1,R0		;It is a demand zero section
			05	00B9	333		RSB			
				00BA	334					
				00BA	335					
				00BA	336					
				00BA	337					
							.END			

SYSGETPTI  
Symbol table

```

ACMODE = 0000000C
EXESGETPTI = 00000000 RG 02
INADR = 00000004
INF_DZRO = 000C0084 R 03
INF_NO = 00000081 R 03
MASK = 00000010
MMGSCREDEL = ***** X 02
MMGSC_LENGTH = FFFFFFFE4
MMGSGETPTIPAG = 00000000 RG 03
MMGSGL_GPTBASE = ***** X 03
MMGSLNADRINI = ***** X 02
MMGSL_SAVRETADR = FFFFFFFF4
MMGSL_SVSTARTVA = FFFFFFFEC
MMGSPAGETYPE = ***** X 03
MMGSPTEREF = ***** X 03
MMGSPRETRANGE = ***** X 02
PAGCNT = 00000014
PRS_IPL = 00000012
PTESM_PGFLVB = 003FFFFFF
PTESM_TYPO = 00400000
PTESM_TYP1 = 04000000
PTESM_VALID = 80000000
PTESV_PGFLVB = 00000016
PTESV_DZRO = 00000011
PTESV_PGFLVB = 00000000
PTESV_TYPO = 00000016
PTESV_TYP1 = 0000001A
RETADR = 00000008
SS$_ACCVIO = 0000000C
SS$_NORMAL = 00000001
    
```

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

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
Y\$EXEPAGED	00000033 ( 51.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$MMGCOD	000000BA ( 186.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.08	00:00:00.42
Command processing	109	00:00:00.57	00:00:03.17
Pass 1	230	00:00:05.69	00:00:14.22
Symbol table sort	0	00:00:00.83	00:00:01.53
Pass 2	74	00:00:01.25	00:00:02.60
Symbol table output	5	00:00:00.05	00:00:00.07
Psect synopsis output	1	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	452	00:00:08.50	00:00:22.04

The working set limit was 1200 pages.  
31420 bytes (62 pages) of virtual memory were used to buffer the intermediate code.  
There were 30 pages of symbol table space allocated to hold 573 non-local and 18 local symbols.  
337 source lines were read in Pass 1, producing 18 object records in Pass 2.  
15 pages of virtual memory were used to define 14 macros.

↑-----↑  
! Macro library statistics !  
↑-----↑

Macro library name	Macros defined
-----	-----
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	5
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	11

656 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SYSGETPTI/OBJ=OBJ\$:SYSGETPTI MSRC\$:SYSGETPTI/UPDATE=(ENH\$:SYSGETPTI)+EXECMLS/LIB

0385 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

A grid of 100 small terminal windows, each displaying a different system utility or diagnostic tool. The windows are arranged in a 10x10 grid. Several windows are highlighted with larger, clearer text labels:

- SYSGETSYI LIS** (row 4, column 4)
- SYSGETPTI LIS** (row 5, column 3)
- SYSGETTIM LIS** (row 6, column 5)
- SYSGETLKI LIS** (row 7, column 1)
- SYSGETMSG LIS** (row 8, column 3)
- SYSIMGACT LIS** (row 9, column 5)
- SYSIMGFIX LIS** (row 5, column 10)

The other windows in the grid contain various data, including system status reports, error logs, and configuration parameters, all rendered in a monospaced font typical of early computer terminals.