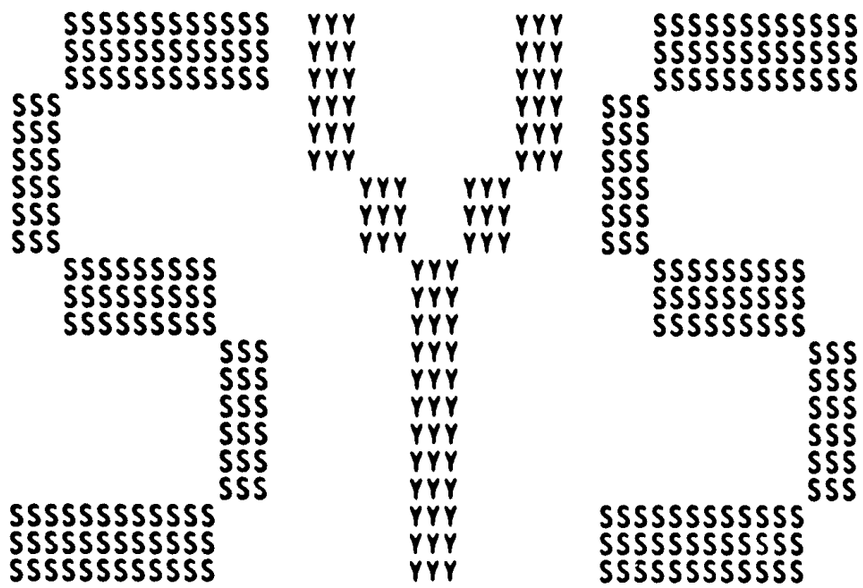

FILEID**WRTMFYPAG

WR
VO

```
WW      WW  RRRRRRRR    TTTTTTTTTT  MM      MM  FFFFFFFFFF  YY      YY  PPPPPPPP      AAAAAA      GGGGGGGG
WW      WW  RRRRRRRR    TTTTTTTTTT  MM      MM  FFFFFFFFFF  YY      YY  PPPPPPPP      AAAAAA      GGGGGGGG
WW      WW  RR      RR      TT      MMMM  MMMM  FF          YY      YY  PP      PP    AA      AA    GG
WW      WW  RR      RR      TT      MMMM  MMMM  FF          YY      YY  PP      PP    AA      AA    GG
WW      WW  RR      RR      TT      MM  MM  MM  FF            YY  YY    PP      PP    AA      AA    GG
WW      WW  RR      RR      TT      MM  MM  MM  FF            YY  YY    PP      PP    AA      AA    GG
WW      WW  RRRRRRRR        TT      MM      MM  FFFFFFFF        YY      PPPPPPPP    AA      AA    GG
WW      WW  RRRRRRRR        TT      MM      MM  FFFFFFFF        YY      PPPPPPPP    AA      AA    GG
WW  WW  WW  RR  RR          TT      MM      MM  FF              YY      PP          AAAAAAAAAA    GG  GGGGGG
WW  WW  WW  RR  RR          TT      MM      MM  FF              YY      PP          AAAAAAAAAA    GG  GGGGGG
WWWW  WWWW  RR    RR        TT      MM      MM  FF              YY      PP          AA      AA    GG      GG
WWWW  WWWW  RR    RR        TT      MM      MM  FF              YY      PP          AA      AA    GG      GG
WW      WW  RR      RR      TT      MM      MM  FF              YY      PP          AA      AA      GGGGGG
WW      WW  RR      RR      TT      MM      MM  FF              YY      PP          AA      AA      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  WRTMFYPAG - WRITE MODIFIED PAGES
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: 3-JAN-77
0000    35 ;
0000    36 ; MODIFIED BY:
0000    37 ;
0000    38 ;       V03-002 LJK0201         Lawrence J. Kenah       18-Apr-1983
0000    39 ;               Clear "disable limit check" flag in upper byte of SCH$GL_MFYLIM
0000    40 ;               along error path taken when page file has no room.
0000    41 ;
0000    42 ;       V03-001 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    43 ;               Add $RSNDEF.
0000    44 ;
0000    45 ;--
```

```
                     47          .SBTTL  DECLARATIONS
                     48 ;
                     49 ; INCLUDE FILES:
                     50 ;
                     51          $CADEF                          ;CONDITIONAL ASSEMBLY DEFINITIONS
                     52          $IRPDEF                         ;I/O REQUEST PACKET DEFINITIONS
                     53          $IPLDEF                         ;PROCESSOR PRIORITY LEVELS
                     54          $OPDEF                          ;DEFINE OPCODE EQUIVALENT VALUES
                     55          $PCBDEF                         ;PROCESS CONTROL BLOCK DEFINITIONS
                     56          $PFLDEF                         ;PAGE FILE CONTROL BLOCK DEFINITIONS
                     57          $PFNDEF                         ;PAGE FRAME NUMBER DATA BASE DEFINITIONS
                     58          $PHDDEF                         ;PROCESS HEADER DEFINITIONS
                     59          $PRDEF                          ;PROCESSOR REGISTER DEFINITIONS
                     60          $PRIDEF                         ;PRIORITY INCREMENT CLASS DEFINITIONS
                     61          $PTEDEF                         ;PAGE TABLE ENTRY DEFINITIONS
                     62          $RSNDEF                         ;RESOURCE NUMBER DEFINITIONS
                     63          $SECDEF                         ;SECTION TABLE ENTRY DEFINITIONS
                     64          $VADEF                          ;VIRTUAL ADDRESS FIELD DEFINITIONS
                     65 ;
                     66 ; MACROS:
                     67 ;
                     68
                     69 ;
                     70 ; EQUATED SYMBOLS:
                     71 ;
                     72          $VIELD  MPW,0,<-
                     73                  <SUCCESS,,M>,-          ;SUCCESSFUL COMPLETION BIT
                     74                  <BADPAG,,M>,-           ;THIS PAGE HAD A WRITE ERROR
                     75                  <NOTDONE,,M> -          ;THESE PAGES WERE NOT WRITTEN
                     76                  >
                     77 ;
                     78 ; OWN STORAGE:
                     79 ;
            00000000 80          .PSECT  $$$210,LONG
                     81          .ALIGN  LONG
                     82 MPW$AL_PTE::
00000000     0000    83          .LONG   0                       ;ADDRESS OF PAGE TABLE ENTRY ARRAY
                     84 MPW$AW_PHVINDEX::
00000000     0004    85          .LONG   0                       ;ADDRESS OF PROCESS HEADER VECTOR INDEX ARRA
                     86 MPW$GL_BADPAGTOTAL::
00000000     0008    87          .LONG   0                       ;TOTAL PAGES EVER PUT ON BAD PAGE LIST
                     88
00000001             89          .IF EQ  1
                     90          .PSECT  $$$075,LONG
                     91          .ALIGN  LONG
                     92 MPW$A_PGFLCLUSTERS::
                     93          .REPT   121
                     94          .LONG   0
                     95          .ENDR
                     96 MPW$A_SECTCLUSTERS::
                     97          .REPT   121
                     98          .LONG   0
                     99          .ENDR
                    100
                    101          .LONG   0                      ;BACKWARD FAILURE COUNT
                    102 MPW$L_BACKUPFAIL::
                    103          .LONG   0                      ;UNUSED
```

```
                000C   104                .LONG   0                       ;FORWARD FAILURE COUNT
                000C   105 MPW$L_COUNT::
                000C   106                .LONG   0
                000C   107                .ENDC
                000C   108 ;
                000C   109 ; ************************************************************************
                000C   110 ;
                000C   111 ; ***************** THIS ENTIRE MODULE MUST BE RESIDENT ******************
                000C   112 ;
            00000000   113                .PSECT  $MMGCOD
                0000   114 ;
                0000   115 ; ************************************************************************
                0000   116 ;
```

```
                     0000   118          .SBTTL  MODIFIED PAGE WRITE COMPLETION AST
                     0000   119 ;++
                     0000   120 ; FUNCTIONAL DESCRIPTION:
                     0000   121 ;
                     0000   122 ;       THIS ROUTINE IS A KERNEL MODE AST WHICH DOES THE CLEANUP OPERATIONS
                     0000   123 ; TO COMPLETE THE WRITING OF THE MODIFIED PAGES.  IT ALSO STARTS
                     0000   124 ; THE NEXT MODIFIED PAGE WRITE IF THERE IS ANY TO DO.
                     0000   125 ;
                     0000   126 ; CALLING SEQUENCE:
                     0000   127 ;
                     0000   128 ;       BSBW    WRITEDONE
                     0000   129 ;
                     0000   130 ; INPUT PARAMETERS:
                     0000   131 ;
                     0000   132 ;       IPL = ASTDEL
                     0000   133 ;       R5 = AST CONTROL BLOCK = I/O REQUEST PACKET
                     0000   134 ;
                     0000   135 ; IMPLICIT INPUTS:
                     0000   136 ;
                     0000   137 ;       NONE
                     0000   138 ;
                     0000   139 ; OUTPUT PARAMETERS:
                     0000   140 ;
                     0000   141 ;       R4, R5 ALTERED
                     0000   142 ;
                     0000   143 ; IMPLICIT OUTPUTS:
                     0000   144 ;
                     0000   145 ;       NONE
                     0000   146 ;
                     0000   147 ; COMPLETION CODES:
                     0000   148 ;
                     0000   149 ;       NONE
                     0000   150 ;
                     0000   151 ; SIDE EFFECTS:
                     0000   152 ;
                     0000   153 ;       NONE
                     0000   154 ;
                     0000   155 ;--
                     0000   156
                     0000   157 WRITEDONE:
          1FC0 8F    BB   0000   158          PUSHR   #^M<R6,R7,R8,R9,R10,R11,AP> ;PRESERVE NON-VOLATILE REGISTERS
                     0004   159          DSBINT  #IPL$_SYNCH                 ;INTERLOCK FOR PFN DATA BASE MANIPULATION
                     000A   160                                              ;SAVE CALLED IPL ON STACK
     07    09    EF   000A   161          EXTZV   #VA$V_VPN,#<16-VA$V_VPN>,- ;GET ORIGINAL PAGE COUNT
  5A    44 A5         000D   162                  IRP$W_OBCNT(R5),R10
  5B    38 A5    D0   0010   163          MOVL    IRP$L_IOST1(R5),R11         ;GET ERROR STATUS AND TRANSFERRED BYTE COUNT
     5B    06    CA   0014   164          BICL    #<MPW$M_BADPAG ! MPW$M_NOTDONE>,R11 ;CLEAR FLAGS
     50    55    D0   0017   165          MOVL    R5,R0                       ;I/O REQUEST PACKET ADDRESS
        FFE3'    30   001A   166          BSBW    EXE$DEANONPAGED             ;DEALLOCATE THE PACKET
           56    D4   001D   167          CLRL    R6                          ;INIT PTE INDEX
57   5B   07    19    EF   001F   168          EXTZV   #<16+VA$V_VPN>,#<16-VA$V_VPN>,R11,R7 ;TRANSFERRED PAGE COUNT
              77    13   0024   169          BEQL    90$                         ;BRANCH IF NO PAGES SUCCESSFULLY TRANSFERRED
   50    0000'DF46    D0   0026   170 20$:     MOVL    @W^MPW$AL_PTE[R6],R0        ;GET PAGE FRAME NUMBER OF NEXT PAGE
   54    0004'DF46    32   002C   171          CVTWL   @W^MPW$AW_PHVINDEX[R6],R4 ;AND THE CORRESPONDING PROCESS
                     0032   172                                              ;HEADER VECTOR INDEX
              07    18   0032   173          BGEQ    22$                         ;BRANCH IF NOT SWAPVBN WRITE
54   54    0F    00    EF   0034   174          EXTZV   #0,#15,R4,R4                ;FIX UP THE PHV INDEX
```

```
                                                H 12

                0C   11  0039   175          BRB     25$                     ;AND IN THE CASE OF A WRITE ERROR
                                  0039   176                                  ;DO NOT JAM ON THE MODIFY BIT
             5B 06   D3  0039   177 22$:     BITL    #<MPW$M_BADPAG ! MPW$M_NOTDONE>,R11 ;NOT SUCCESSFULLY TRANSFERRED?
                07   13  003E   178          BEQL    25$                     ;BRANCH IF THIS PAGE IS OK
0000'DF40    80 8F   88  0040   179          BISB    #PFN$M_MODIFY,@W^PFN$AB_STATE[R0] ;NOTE PAGE STILL MODIFIED
   58  0000'DF40    90  0047   180 25$:     MOVB    @W^PFN$AB_TYPE[R0],R8   ;PAGE TYPE AND RPTEVT BIT
          03   00   ED  004D   181          CMPZV   #PFN$V_PAGTYP,#PFN$S_PAGTYP,- ;PROCESS PAGE TABLE?
          04   58       0050   182                  R8,#PFN$C_PPGTBL
               06   12  0052   183          BNEQ    40$                     ;BRANCH IF NOT
          51   54   D0  0054   184          MOVL    R4,R1                   ;PROCESS HEADER VECTOR INDEX
             FFA6'  30  0057   185          BSBW    MMG$DECPHDREF1          ;ONE LESS PROCESS HEADER REF
                                  005A   186                                  ;AT PAGE WRITE COMPLETION
                                  005A   187 40$:     DECREF  GTR=60$         ;ONE LESS REFERENCE
          0E 5B   01   E1  0066   188          BBC     #MPW$V_BADPAG,R11,50$   ;BRANCH IF NOT PAGE WRITE ERROR PAGE
             52   02   9A  006A   189          MOVZBL  #PFN$C_BADPAGLST,R2     ;PLACE THIS PAGE
             FF90'  30  006D   190          BSBW    MMG$INSPFNT             ;ON THE BAD PAGE LIST
   00000008'EF    D6  0070   191          INCL    MPW$GL_BADPAGTOTAL      ;COUNT IT
               03   11  0076   192          BRB     60$
             FF85'  30  0078   193 50$:     BSBW    MMG$RELPFN              ;RELEASE THE PAGE
54   0000'DF44    32  007B   194 60$:     CVTWL   @W^PHV$GL_PIXBAS[R4],R4 ;CALCULATE PCB ADDRESS FROM PHV INDEX
               16   19  0081   195          BLSS    80$                     ;BRANCH IF PCB IS GONE
54   0000'DF44    D0  0083   196          MOVL    @W^SCH$GL_PCBVEC[R4],R4 ;FETCH PCB ADR
 00 24 A4      10   E4  0089   197          BBSC    #PCB$V_SWPVBN,PCB$L_STS(R4),70$ ;DONE WITH SWPVBN WRITE IF ANY
                                  008E   198 ;
                                  008E   199 ; IF DELPAG IS WAITING FOR THIS WRITE COMPLETION, THE RPTEVT BIT IS SET
                                  008E   200 ;
07 58   06   E1  008E   201 70$:     BBC     #PFN$V_RPTEVT,R8,80$    ;BRANCH IF NO REPORT EVENT REQUESTED
          52   01   9A  0092   202          MOVZBL  #PRI$_IOCOM,R2          ;I/O COMPLETE PRIORITY CLASS
                         0095   203          RPTEVT  PFCOM                   ;REPORT PAGE FAULT COMPLETE
89 56   57   F2  0099   204 80$:     AOBLSS  R7,R6,20$               ;LOOP THROUGH ALL PAGES
     10 5B   E8  009D   205 90$:     BLBS    R11,100$                ;BRANCH IF PAGE WRITE HAD NO ERROR
          57   D6  00A0   206          INCL    R7                      ;ONE MORE PAGE FOR THE ERROR PAGE
80 5B   01   E3  00A2   207          BBCS    #MPW$V_BADPAG,R11,20$   ;BRANCH IF HAVEN'T PROCESSED THE ERROR PAGE
                                  00A6   208 ;
                                  00A6   209 ; NOW PROCESS THE UNWRITTEN PAGES IF ANY
                                  00A6   210 ;
   57   5A   D0  00A6   211          MOVL    R10,R7                  ;RESET LIMIT TO ORIGINAL PAGE COUNT
          56   D7  00A9   212          DECL    R6                      ;IN ORDER TO EXECUTE THE LOOP 0 OR MORE TIME
   5B   05   90  00AB   213          MOVB    #<MPW$M_SUCCESS ! MPW$M_NOTDONE>,R11 ;COMPLETE THE PAGES NOT TRANSFE
          E9   11  00AE   214          BRB     80$
                         00B0   215 100$:    SETIPL  (SP)                    ;BACK TO CALLED IPL
          00   DD  00B3   216          PUSHL   #0                      ;NO I/O PACKET ALLOCATED YET
0000'CF   0000'CF  D1  00B5   217          CMPL    W^SCH$GL_MFYCNT,W^SCH$GL_MFYLOLIM ;ENOUGH PAGES ON MODIFY LIST
                         00BC   218                                  ;TO TRY FOR ANOTHER CLUSTER?
          6F   1A  00BC   219          BGTRU   GET_NXT_CLUSTER         ;BRANCH IF YES
          10   11  00BE   220          BRB     NOMOREPAGES             ;NO, ALL DONE FOR NOW
```

```
                        00C0   222          .SBTTL  WRTMFYPAG - WRITE MODIFIED PAGES
                        00C0   223 ;++
                        00C0   224 ; FUNCTIONAL DESCRIPTION:
                        00C0   225 ;
                        00C0   226 ;       THIS ROUTINE GATHERS A CLUSTER OF PAGES OF LIKE KIND FROM THE
                        00C0   227 ; MODIFIED PAGE LIST, AND WRITES THEM BACK TO THEIR BACKING STORE ADDRESSES.
                        00C0   228 ; FOR PAGING FILE PAGES, THE ADDRESSES ARE REALLOCATED AS A CONTIGUOUS BLOCK
                        00C0   229 ; AND THE PAGES ARE WRITTEN BACK IN ONE OPERATION.
                        00C0   230 ;
                        00C0   231 ; CALLING SEQUENCE:
                        00C0   232 ;
                        00C0   233 ;       BSBW    MMG$WRTMFYPAG
                        00C0   234 ;
                        00C0   235 ; INPUT PARAMETERS:
                        00C0   236 ;
                        00C0   237 ;       IPL = 0
                        00C0   238 ;
                        00C0   239 ; IMPLICIT INPUTS:
                        00C0   240 ;
                        00C0   241 ;       NONE
                        00C0   242 ;
                        00C0   243 ; OUTPUT PARAMETERS:
                        00C0   244 ;
                        00C0   245 ;       R4, R5 ALTERED
                        00C0   246 ;
                        00C0   247 ; IMPLICIT OUTPUTS:
                        00C0   248 ;
                        00C0   249 ;       NONE
                        00C0   250 ;
                        00C0   251 ; COMPLETION CODES:
                        00C0   252 ;
                        00C0   253 ;       NONE
                        00C0   254 ;
                        00C0   255 ; SIDE EFFECTS:
                        00C0   256 ;
                        00C0   257 ;       NONE
                        00C0   258 ;
                        00C0   259 ;--
                        00C0   260 ;
                        00C0   261 ; NEED TO ALLOCATE AN I/O PACKET AND WAIT IF NONE AVAILABLE
                        00C0   262 ; ***** IT IS ASSUMED THAT THIS PROCESS IS RUNNING WITH RESOURCE WAIT ENABLED
                        00C0   263 ;
                        00C0   264 GET_IRP:
          FF3D'  30     00C0   265          BSBW    EXE$ALLOCIRP            ;ALLOCATE AN I/O REQUEST PACKET
        73 50    E8     00C3   266          BLBS    R0,GOT_IRP              ;BRANCH IF GOT ONE, MAY HAVE WAITED
                        00C6   267          BUG_CHECK MPWA[CIRP,FATAL       ;FAILED TO ALLOCATE I/O REQUEST PACKET
                        00CA   268
                        00CA   269          .ENABLE         LOCAL_BLOCK
                        00CA   270
                        00CA   271 NOSPACE:
     0003'CF    94     00CA   272          CLRB    W^SCH$GL_MFYLIM + 3     ;ENABLE THRESHOLD CHECKING AGAIN
          18    11     00CE   273          BRB     5$
                        00D0   274
                        00D0   275 NOMOREPAGES:
0000'CF 0000'CF  D0     00D0   276          MOVL    W^SCH$GL_MFYLIMSV,W^SCH$GL_MFYLIM ;RE-ENABLE LIMIT CHECK
0000'CF 0000'CF  D0     00D7   277          MOVL    W^SCH$GL_MFYLOSV,W^SCH$GL_MFYLOLIM ;RESTORE LOW LIMIT
     50 0000'CF  D0     00DE   278          MOVL    W^SCH$GL_CURPCB,R0      ;GET PCB ADDRESS
```

```
                00 24 A0    0C   E2  00E3   279          BBSS    #PCB$V_WAKEPEN,PCB$L_STS(R0),5$ ;SET WAKE PENDING TO FORCE
                                      00E8   280                                                  ; SWAP SCHEDULE RE-EVALUATION
             00 0000'CF    00'   E7  00E8   281 5$:      BBCCI   S^#SCH$V_MPW,W^SCH$GB_SIP,10$ ;MODIFIED PAGE WRITER INACTIVE
                     50    0C    9A  00EE   282 10$:     MOVZBL  #RSN$_MPWBUSY,R0            ;SET RESOURCE WAIT THAT IS SATISFIED
                        FF0C'      30  00F1   283          BSBW    SCH$RAVAIL                  ;DECLARE MODIFIED PAGE WRITER DONE
                        0000'CF    D5  00F4   284          TSTL    W^SCH$GL_MFYCNT             ;IS LIST EMPTY
                              06   12  00F8   285          BNEQ    15$
                     50    08    9A  00FA   286          MOVZBL  #RSN$_MPLEMPTY,R0           ;SET RESOURCE WAIT THAT IS SATISFIED
                        FF00'      30  00FD   287          BSBW    SCH$RAVAIL                  ;DECLARE MODIFIED LIST EMPTY
                     50    8E    D0  0100   288 15$:     MOVL    (SP)+,R0                    ;UNUSED I/O PACKET TO RELEASE?
                              08   13  0103   289          BEQL    20$                         ;BRANCH IF NOT
                08 A0    C4 8F    9A  0105   290          MOVZBL  #IRP$C_LENGTH,IRP$W_SIZE(R0) ;SET SIZE OF PACKET
                           FEF3'    30  010A   291          BSBW    EXE$DEANONPAGED             ;DEALLOCATE THE I/O REQUEST PACKET
                                      010D   292 20$:     ENBINT                              ;RESTORE CALLER'S IPL
                        1FC0 8F    BA  0110   293          POPR    #^M<R6,R7,R8,R9,R10,R11,AP> ;RESTORE NON-VOLATILE REGISTERS
                              05   05  0114   294          RSB                                 ;AND RETURN
                                      0115   295
                                      0115   296          .DISABLE        LOCAL_BLOCK
                                      0115   297
                                      0115   298 MMG$WRTMFYPAG::
          0000'CF    15    00    ED  0115   299          CMPZV   #PTE$V_PFN,#PTE$S_PFN,W^SCH$GL_MFYLIM,- ;ENOUGH PAGES ON MODIFIED
                        0000'CF      011B   300                  W^SCH$GL_MFYCNT             ;PAGE LIST TO CONSIDER WRITING?
                              01   15  011E   301          BLEQ    20$                         ;BRANCH IF YES
                              05   05  0120   302 10$:     RSB                                 ;NO, NOTHING TO DO FOR NOW
             F9 0000'CF    00'   E6  0121   303 20$:     BBSSI   S^#SCH$V_MPW,W^SCH$GB_SIP,10$ ;DO NOTHING IF ALREADY ACTIVE
                                      0127   304 ;
                                      0127   305 ; FIRST ENTRY TO MODIFIED PAGE WRITER
                                      0127   306 ;
                        1FC0 8F    BB  0127   307          PUSHR   #^M<R6,R7,R8,R9,R10,R11,AP> ;SAVE NON-VOLATILE REGISTERS
                              7E   7C  012B   308          CLRQ    -(SP)                       ;SAVED IPL = 0, NO IRP ALLOCATED
                                      012D   309 ;
                                      012D   310 ; SEE IF THERE ARE ANY MORE PAGES TO BE WRITTEN
                                      012D   311 ; 0(SP) = SAVED I/O REQUEST PACKET ADDRESS OR ZERO IF NONE ALLOCATED YET
                                      012D   312 ; 4(SP) = CALLERS IPL
                                      012D   313 ; IPL = ASTDEL OR LOWER
                                      012D   314 ;
                                      012D   315 GET_NXT_CLUSTER:
          54    0000'CF    D0  012D   316          MOVL    W^SCH$GL_CURPCB,R4          ;GET PCB ADDRESS OF THIS PROCESS
                                      0132   317 ;
                                      0132   318 ; ***** NOTE THAT PCB$W_DIOCNT(R4) IS ASSUMED GREATER THAN 0
                                      0132   319 ; ***** SIMULTANEOUS I/O REQUESTS WILL INVALIDATE THIS ASSUMPTION
                                      0132   320 ;
          52    0000'DF    0F  0132   321          REMQUE  @W^IOC$GL_IRPFL,R2          ;GET AN I/O REQUEST PACKET
                              87   1D  0137   322          BVS     GET_IRP                     ;BRANCH IF NONE ON SIDE LIST
                                      0139   323 GOT_IRP:
                     6E    52    D0  0139   324          MOVL    R2,(SP)                     ;SAVE IRP ADDRESS
                                      013C   325          SETIPL  #IPL$_SYNCH                 ;RAISE TO SYNCH FOR THE DURATION
          50    0000'CF    D0  013F   326          MOVL    W^PFN$AL_MFYLSTHD,R0        ;FIRST PFN IN MODIFIED PAGE LIST
                                      0144   327 GET_PAGE_TYPE:
                        8A   13  0144   328          BEQL    NOMOREPAGES                 ;BRANCH IF LIST IS EMPTY
                           014B   30  0146   329          BSBW    GETPFNCTX                   ;SET UP TO PROCESS THIS PFN
                              56   D4  0149   330          CLRL    R6                          ;INIT INDEX TO PTE ARRAY
                                      014B   331          CASE    R1,<-                       ;DISPATCH ON BACKING STORE TYPE
                                      014B   332                  PAGEFILE,-                  ;PAGING FILE PAGE
                                      014B   333                  SECTION,-                   ;SECTION PAGE (PROCESS OR GLOBAL)
                                      014B   334                  BADBAKADR,-                 ;GLOBAL BACKING STORE ADDRESS
                                      014B   335                  SWPVBN  >                   ;SWPVBN, WRITE BACK TO SWAP FILE
```

```
                        0157   336 BADBAKADR:
                        0157   337         BUG_CHECK IVBAKADIO,FATAL       ;INVALID BACKING STORE ADDRESS FOR I/O
                        015B   338 ;
                        015B   339 ; NO PAGE FILE VBN'S AVAILABLE IN THIS PAGE FILE
                        015B   340 ;
                        015B   341 NO_PAGEFILE:
              00F9 8F   BA  015B   342         POPR    #^M<R0,R3,R4,R5,R6,R7>  ;RECOVER REGISTERS FROM STACK
                 FF68   31  015F   343         BRW     NOSPACE                 ;EXIT, IF NO SPACE TO WRITE STUFF
                        0162   344 NEXT_MFYPAG:
                        0162   345                 PFN_REFERENCE   -
                        0162   346         MOVZWL  <@W^PFN$Ax_FLINK[R0],R0>,-      ;CHAIN TO NEXT PAGE
                        0162   347                 LONG_OPCODE=MOVL,-
                        0162   348                 IMAGE=SYS_NONPAGED
                     DA 11  0168   349         BRB     GET_PAGE_TYPE           ;AND PROCESS IT
                        016A   350 ;
                        016A   351 ; PAGE FILE PAGE, GATHER A CLUSTER FROM THE SAME PAGE FILE
                        016A   352 ;
                        016A   353 PAGEFILE:
      57   52   08   18  EF  016A   354         EXTZV   #PFN$V_PGFLX,#PFN$S_PGFLX,R2,R7 ;PAGE FILE INDEX
              00F9 8F   BB  016F   355         PUSHR   #^M<R0,R3,R4,R5,R6,R7>  ;SAVE PFN,SVAPTE,PCB,PHD,PTE INDEX,FILE
                   50   7C  0173   356         CLRQ    R0                      ;NOTHING TO FREE
         53  0000'DF47   D0  0175   357         MOVL    @W^MMG$GL_PAGSWPVC[R7],R3 ;GET PAGE FILE CONTROL BLOCK ADDRESS
             52   22 A3   9A  017B   358         MOVZBL  PFL$B_ALLOCSIZ(R3),R2   ;DESIRED CLUSTER SIZE
                FE7E'   30  017F   359 5$:      BSBW    MMG$ALLOCPAGFIL1        ;ALLOCATE A CLUSTER
                   13   12  0182   360         BNEQ    20$                     ;BRANCH IF WE HAVE ALLOCATION
                52   10   C2  0184   361         SUBL    #16,R2                  ;ALLOCATION FAILED, TRY SMALLER CLUSTER
                   09   15  0187   362         BLEQ    10$                     ;BRANCH IF ALREADY AT MINIMUM
             22 A3   52   90  0189   363         MOVB    R2,PFL$B_ALLOCSIZ(R3)   ;SET NEW SIZE TO ATTEMPT FROM NOW ON
                04 A3   D4  018D   364         CLRL    PFL$L_STARTBYTE(R3)     ;START AT BEGINNING OF MAP
                   ED   11  0190   365         BRB     5$                      ;TRY AGAIN
                        0192   366
                FE6B'   30  0192   367 10$:     BSBW    MMG$ALLOCPAGFIL2        ;ALLOCATE SPACE, ANY AMOUNT OK
                   C4   13  0195   368         BEQL    NO_PAGEFILE             ;BRANCH IF NONE, NO WRITING CAN BE DONE
      5B   52   D0  0197   369 20$:     MOVL    R2,R11                  ;SAVE NUMBER OF PAGES ALLOCATED
      58   50   D0  019A   370         MOVL    R0,R8                   ;AND THE STARTING PAGE FILE VBN
00F9  8F   BA  019D   371         POPR    #^M<R0,R3,R4,R5,R6,R7>  ;RECOVER PFN, SVAPTE
      59   5B   D0  01A1   372         MOVL    R11,R9                  ;NUMBER OF PAGEFILE PAGES ALLOCATED
                        01A4   373 ;
                        01A4   374 ; AT THIS POINT THE REGISTERS CONTAIN THE FOLLOWING VALUES:
                        01A4   375 ;       R0  = PFN
                        01A4   376 ;       R3  = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
                        01A4   377 ;       R4  = PCB ADDRESS FOR THE PROCESS IN WHICH THIS CODE IS RUNNING
                        01A4   378 ;       R5  = PHD ADDRESS OF THE PROCESS WHICH OWNS THE MODIFIED PAGE
                        01A4   379 ;       R6  = INDEX TO NEXT ENTRY TO USE IN PTE AND PHVINDEX ARRAYS
                        01A4   380 ;       R7  = PAGE FILE INDEX
                        01A4   381 ;       R8  = NEXT PAGE FILE VBN TO USE
                        01A4   382 ;       R9  = NUMBER OF PAGE FILE VBN'S NOT YET USED IN THE CLUSTER ALLOCATED
                        01A4   383 ;       R11 = NUMBER OF PAGE FILE VBN'S ALLOCATED IN THE CLUSTER
                        01A4   384 ;
                        01A4   385 PAGFILCLUSTER:
                     5B DD  01A4   386         PUSHL   R11                     ;SAVE COUNT OF ALLOCATED PAGE FILE
                        01A6   387                 PFN_REFERENCE   -
                        01A6   388         MOVZWL  <@W^PFN$Ax_BLINK[R0],-(SP) >,-  ;REMEMBER WHERE TO RESTART SCAN OF L
                        01A6   389                 LONG_OPCODE=MOVL,-
                        01A6   390                 IMAGE=SYS_NONPAGED
                        01AC   391 ;       CLRL    MPW$L_COUNT             ;INIT COUNT OF CLUSTER TO 0
                  016E   30  01AC   392         BSBW    PTESCAN                 ;TRY TO GET ADJACENT PAGES TO THIS ONE
```

```
                                01AF    393 ;       MOVL    MPW$L_COUNT,R0          ;GET THE COUNT
                                01AF    394 ;       INCL    MPW$A_PGFLCLUSTERS[R0]  ;BUMP THE COUNT
                                01AF    395
                                01AF    396 ;
                                01AF    397 ; DONE WITH THIS CLUSTER OF PAGE TABLE ENTRIES
                                01AF    398 ; IF MORE PAGE FILE VBN'S ARE LEFT, SCAN MODIFIED LIST FOR MORE PAGES
                                01AF    399 ; IN SAME PAGE FILE.
                                01AF    400 ; 0(SP) = SAVED PFN (OR 0) TO LINK FORWARD FROM
                                01AF    401 ; 4(SP) = NUMBER OF PAGE FILE VBN'S ALLOCATED
                                01AF    402 ;
                    0801 8F BA  01AF    403         POPR    #^M<R0,R11>             ;R0 = SAVED PFN TO LINK FORWARD
                                01B3    404                                         ;R11 = NUMBER OF PAGE FILE VBN'S ALLOCATED
                59  5B   56 C3  01B3    405         SUBL3   R6,R11,R9               ;NO. OF PAGE FILE VBN'S NOT USED
                                01B7    406 ; DISABLE THE FOLLOWING BECAUSE THIS MIGHT RETURN A SMALL ALLOCATION WITHOUT
                                01B7    407 ; WRITING ANY PAGES AT ALL.
                                01B7    408 ;       CMPW    R9,W^MPW$GW_MINLIM      ;ARE WE AT THRESHOLD TO SHUT OFF CLUSTER
                      24 15     01B7    409         BLEQ    100$                    ;BRANCH IF USED ALL THAT WE SHOULD
                      50 D5     01B9    410         TSTL    R0                      ;PFN 0 IS LIST HEAD
                      07 12     01BB    411         BNEQ    70$                     ;GET FLINK AS NEXT CANDIDATE
                50  0000'CF D0 01BD    412         MOVL    W^PFN$AL_MFYLSTHD,R0    ;NEXT CANDIDATE FROM FRONT OF LIST
                      06 11     01C2    413         BRB     75$
                                01C4    414 70$:                PFN_REFERENCE   -
                                01C4    415         MOVZWL  <@W^PFN$Ax_FLINK[R0],R0>,-      ;CHAIN TO NEXT PFN IN LIST
                                01C4    416                 LONG_OPCODE=MOVL,-
                                01C4    417                 IMAGE=SYS_NONPAGED
                      11 13     01CA    418 75$:    BEQL    100$                    ;BRANCH IF END OF LIST
                   00C5 30      01CC    419 80$:    BSBW    GETPFNCTX               ;SET UP TO PROCESS THIS PFN
                      51 D5     01CF    420         TSTL    R1                      ;PAGE FILE VBN?
                      F1 12     01D1    421         BNEQ    70$                     ;BRANCH IF NOT
            57  52  08  18 ED  01D3    422         CMPZV   #PFN$V_PGFLX,#PFN$S_PGFLX,R2,R7 ;SAME PAGE FILE INDEX?
                      EA 12     01D8    423         BNEQ    70$                     ;BRANCH IF NOT
                   FFC7 31      01DA    424         BRW     PAGFILCLUSTER           ;FIND ANOTHER PTE CLUSTER
                                01DD    425 ;
                                01DD    426 ; SET UP TO WRITE THIS CLUSTER OF PAGES
                                01DD    427 ;
                5B  58   56 C3  01DD    428 100$:   SUBL3   R6,R8,R11               ;FORM AND SAVE FIRST PAGE FILE VBN
             53  0000'DF47 D0    01E1    429         MOVL    @W^MMG$GL_PAGSWPVC[R7],R3 ;ADDRESS OF PAGE FILE CONTROL BLOCK
                   51  59 D0     01E7    430         MOVL    R9,R1                   ;ANY PAGE FILE VBN'S TO DEALLOCATE?
                      06 13     01EA    431         BEQL    140$                    ;BRANCH IF THEY WERE ALL USED
                   50  58 D0     01EC    432         MOVL    R8,R0                   ;SET VBN NUMBER OR AREA TO DEALLOCATE
                                01EF    433 ;
                                01EF    434 ; THERE ARE R1 PAGES OF PAGE FILE ALLOCATED BUT NOT USED STARTING WITH VBN R0.
                                01EF    435 ; MUST RETURN THEM TO THE PAGE FILE
                                01EF    436 ;
                   FE0E' 30     01EF    437         BSBW    MMG$DEALLOCPAGFIL       ;FREE THE PAGES IN THE FILE
                                01F2    438 ;
                                01F2    439 ; NOW SET UP TO DO THE CALL TO BUILDPKT
                                01F2    440 ;       R3 = PAGE FILE CONTROL BLOCK ADDRESS
                                01F2    441 ;       R6  = NO. OF PAGES TO TRANSFER
                                01F2    442 ;       R7  = PAGE FILE INDEX
                                01F2    443 ;       R11 = STARTING PAGEFILE VBN
                                01F2    444 ;
                                01F2    445         ASSUME  SEC$L_VBN EQ PFL$L_VBN
                                01F2    446         ASSUME  SEC$L_WINDOW EQ PFL$L_WINDOW
          50  10 A3   5B C1    01F2    447 140$:   ADDL3   R11,PFL$L_VBN(R3),R0    ;FORM VBN IN PAGE FILE
              52  0C A3   D0   01F7    448         MOVL    PFL$L_WINDOW(R3),R2     ;WINDOW ADDRESS
                                01FB    449 ;
```

```
                         01FB    450 ; 0(SP) = ADDRESS OF I/O REQUEST PACKET
                         01FB    451 ; 4(SP) = CALLER'S IPL
                         01FB    452 ; R0 = VBN IN FILE TO WRITE
                         01FB    453 ; R2 = ADDRESS OF WINDOW FOR FILE
                         01FB    454 ; R4 = PROCESS CONTROL BLOCK ADDRESS FOR THE PROCESS IN WHICH THIS CODE IS RUNNING
                         01FB    455 ; R6 = NUMBER OF PAGES TO WRITE
                         01FB    456 ;
                         01FB    457 MPW_BLDPKT:
          51    56   09   78   01FB    458         ASHL    #9,R6,R1                ;NUMBER OF BYTES TO WRITE
          53    0000'DF   DE   01FF    459         MOVAL   @W^MPW$AL_PTE,R3        ;SVAPTE FOR TRANSFER
                55    8E   D0   0204    460         MOVL    (SP)+,R5                ;GET I/O PACKET ADDRESS
       14 A5    FDF5 CF   DE   0207    461         MOVAL   W^WRITEDONE,IRP$L_ASTPRM(R5) ;ADDRESS OF KERNEL MODE AST
                         020D    462                                                 ;FOR WRITE COMPLETION PROCESSING
    23 A5   1F   0000'CF   83   020D    463         SUBB3   W^MPW$GB_PRIO,#31,IRP$B_PRI(R5) ;SET PRIORITY FOR TRANSFER
                   FDE9'   30   0214    464         BSBW    EXE$BLDPRTSWPW          ;BUILD AND QUEUE THE I/O REQUEST PACKET
                         0217    465 ;
                         0217    466 ; R4 NO LONGER HAS PCB ADDRESS IN IT
                         0217    467 ;
                         0217    468         ENBINT                              ;BACK TO CALLED IPL
                         021A    469
                00000002 021A    470         .IF     GT,CA$_MEASURE
          0000'CF   56   C0   021A    471         ADDL    R6,W^PMS$GL_PWRITES     ;COUNT THE PAGES WRITTEN
                0000'CF   D6   021F    472         INCL    W^PMS$GL_PWRITIO        ;AND THE NUMBER OF WRITE REQUESTS
                         0223    473         .ENDC
                         0223    474
                1FC0 8F   BA   0223    475         POPR    #^M<R6,R7,R8,R9,R10,R11,AP> ;RESTORE NON-VOLATILE REGISTERS
                         05   0227    476         RSB
                         0228    477 ;
                         0228    478 ; SECTION PAGE - TRY TO FORM A CLUSTER OF THESE
                         0228    479 ;
                         0228    480
                         0228    481         .ENABL  LSB
                         0228    482
                         0228    483 SECTION:
            57    52   D0   0228    484         MOVL    R2,R7                   ;BACKING STORE ADDRESS
       59   0000'CF   3C   022B    485         MOVZWL  W^MPW$GW_MPWPFC,R9      ;MAXIMUM NUMBER OF PAGES TO CLUSTER
                         0230    486 ;       CLRL    MPW$L_COUNT             ;INIT COUNT OF CLUSTER TO 0
                00EA   30   0230    487         BSBW    PTESCAN                 ;LOOK AT ADJACENT PTE'S FOR A CLUSTER TO WRI
                         0233    488 ;       MOVL    MPW$L_COUNT,R0          ;GET THE COUNT
                         0233    489 ;       INCL    MPW$A_SECTCLUSTERS[R0]  ;BUMP THE COUNT
          50   0000'DF   D0   0233    490         MOVL    @W^MPW$AL_PTE,R0        ;GET STARTING PAGE NUMBER
    52  0000'DF40   D0   0238    491         MOVL    @W^PFN$AL_BAK[R0],R2    ;GET ITS BACKING STORE ADDRESS
    53  0000'DF40   D0   023E    492         MOVL    @W^PFN$AL_PTE[R0],R3    ;AND ITS PAGE TABLE ENTRY ADDRESS
                         0244    493 ;
                         0244    494 ; R5 = PROCESS HEADER ADR ASSOCIATED WITH THIS SVAPTE FROM ORIGINAL GETPFNCTX CALL
                         0244    495 ;
                FDB9'   30   0244    496 10$:    BSBW    MMG$INIBLDPKT           ;TRANSLATE BACKING STORE TO VBN AND WINDOW
                  B2    11   0247    497         BRB     MPW_BLDPKT              ;GO QUEUE THE REQUEST
                         0249    498 ;
                         0249    499 ; THIS PAGE IS A SWPVBN PAGE AND IS TO BE WRITTEN BACK TO THE SWAP FILE
                         0249    500 ; RATHER THAN ITS NORMAL BACKING STORE ADDRESS
                         0249    501 ;
                         0249    502 SWPVBN:
 3E 0000'CF    00'   E0   0249    503         BBS     S^#SCH$V_SIP,W^SCH$GB_SIP,40$ ;DON'T PROCESS SWAP VBN'S
                         024F    504                                             ;IF SWAPPER IS ACTIVE
      51   42 A5   3C   024F    505         MOVZWL  PHD$W_PHVINDEX(R5),R1   ;GET PROCESS HEADER VECTOR INDEX
   51   0000'DF41   32   0253    506         CVTWL   @W^PHV$GL_PIXBAS[R1],R1 ;GET PROCESS INDEX
```

```
                35   19  0259   507          BLSS    PROCESS_GONE                ;BRANCH IF PROCESS IS NO LONGER AROUND
 57   0000'DF41   D0  025B   508          MOVL    @W^SCH$GL_PCBVEC[R1],R7 ;GET PCB ADDRESS
     59   30 A7   3C  0261   509          MOVZWL  PCB$W_APTCNT(R7),R9     ;FORM VBN IN SWAP SLOT
        59   52   C0  0265   510          ADDL    R2,R9                   ;R2 HAD THE SWAP VBN
     59   20 A7   C0  0268   511          ADDL    PCB$L_WSSWP(R7),R9      ;FORM SWAP FILE BACKING STORE ADDRESS
      0000'DF40   B4  026C   512          CLRW    @W^PFN$AW_SWPVBN[R0]    ;ALL DONE WITH THE SWPVBN FIELD
 00 24 A7   10   E2  0271   513          BBSS    #PCB$V_SWPVBN,PCB$L_STS(R7),20$ ;NOTE SWPVBN WRITE IN PROGRESS
        5C   53   D0  0276   514 20$:     MOVL    R3,AP                   ;STARTING PTE RANGE
        5A   53   D0  0279   515          MOVL    R3,R10                  ;ENDING PTE RANGE
        57   01   CE  027C   516          MNEGL   #1,R7                   ;SET PTE$V_TYPO
           0118   30  027F   517          BSBW    SCAN_DONE               ;PUT JUST THIS PAGE IN MPW PTE ARRAY
                      0282   518 ;
                      0282   519 ; R9, R10 ASSUMED PRESERVED
                      0282   520 ;
        52   59   7D  0282   521          MOVQ    R9,R2                   ;R2 = BACKING STORE ADR, R3 = SVAPTE
B9 0004'DF   0F   E3  0285   522          BBCS    #15,@W^MPW$AW_PHVINDEX,10$ ;INDICATE SWAP VBN PAGE WRITE
             B7   11  028B   523          BRB     10$                     ;SET UP FOR BUILDPKT
                      028D   524 ;
                      028D   525 ; DO NOT PROCESS THIS MODIFIED PAGE WITH SWPVBN SET SINCE SWAPPER IS ACTIVE
                      028D   526 ;
           FED2   31  028D   527 40$:     BRW     NEXT_MFYPAG             ;SKIP THIS MODIFIED PAGE
                      0290   528
                      0290   529          .DSABL  LSB
                      0290   530 ;
                      0290   531 ; PROCESS WAS DELETED AND SWAP VBN WAS SET IN PFN DATA BASE
                      0290   532 ;
                      0290   533 PROCESS_GONE:
                      0290   534          BUG_CHECK PROCGONE,FATAL        ;PROCESS NOT IN SYSTEM
```

```
                     0294   536          .SBTTL  GETPFNCTX
                     0294   537 ;
                     0294   538 ; CALLING SEQUENCE:
                     0294   539 ;
                     0294   540 ;          BSBW    GETPFNCTX
                     0294   541 ;
                     0294   542 ; INPUTS:
                     0294   543 ;
                     0294   544 ;          R0 = PFN
                     0294   545 ;
                     0294   546 ; OUTPUTS:
                     0294   547 ;
                     0294   548 ;          R0 = PFN (PRESERVED)
                     0294   549 ;          R1 = TYPE OF BACKING STORE ADDRESS
                     0294   550 ;             = 0 IF PAGING FILE
                     0294   551 ;             = 1 IF SECTION ADDRESS
                     0294   552 ;             = 2 IF ILLEGAL
                     0294   553 ;             = 3 IF SWPVBN
                     0294   554 ;             = 4 IF NOT A PAGE ON THE MODIFIED PAGE LIST (CHKNXTPTE)
                     0294   555 ;          R2 = BACKING STORE ADDRESS OR SWPVBN
                     0294   556 ;          R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
                     0294   557 ;          R4 = PRESERVED
                     0294   558 ;          R5 = PROCESS HEADER ADDRESS
                     0294   559 ;
                     0294   560 GETPFNCTX:
          53 0000'DF40 D0    0294   561          MOVL    @W^PFN$AL_PTE[R0],R3    ;SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
51 0000'DF40  03  00 EF    029A   562          EXTZV   #PFN$V_PAGTYP,#PFN$S_PAGTYP,@W^PFN$AB_TYPE[R0],R1 ;PAGE TYPE
                     02A2   563
                     02A2   564          ASSUME  PFN$C_PROCESS EQ 0
                     02A2   565          ASSUME  PFN$C_SYSTEM  EQ 1
                     02A2   566          ASSUME  PFN$C_GLOBAL  EQ 2
                     02A2   567          ASSUME  PFN$C_GBLWRT  EQ 3
                     02A2   568          ASSUME  PFN$C_PPGTBL  EQ 4
                     02A2   569          ASSUME  PFN$C_GPGTBL  EQ 5
                     02A2   570
                     02A2   571          CASE    R1,<-
                     02A2   572                  PROCESS,-               ;PROCESS PAGE
                     02A2   573                  SYSPHD,-                ;SYSTEM PAGE
                     02A2   574                  BADTYP,-                ;GLOBAL READ ONLY
                     02A2   575                  SYSPHD,-                ;GLOBAL WRITABLE
                     02A2   576                  PHDR,-                  ;PROCESS PAGE TABLE
                     02A2   577                  SYSPHD-                 ;GLOBAL PAGE TABLE
                     02A2   578          >
                     02B2   579 BADTYP:  BUG_CHECK BADPAGTYPE,FATAL      ;BAD PAGE TYPE
                     02B6   580
                     02B6   581 SYSPHD:
          55 0000'DF  DE     02B6   582          MOVAL   @W^MMG$GL_SYSPHD,R5     ;ADDRESS OF SYSTEM HEADER
                2E   11     02BB   583          BRB     GOTPHDR                 ;JOIN THE COMMON CODE
                     02BD   584
                     02BD   585 PHDR:
55   53 0000'CF  C3     02BD   586          SUBL3   W^SWP$GL_BALSPT,R3,R5   ;NO. OF BYTES INTO SPT BEYOND BALSET BASE
     55 55  07   78     02C3   587          ASHL    #7,R5,R5                ;NO. OF SPT ENTRIES * 512
           06    11     02C7   588          BRB     GETPHDR                 ;GET PROCESS HEADER ADDRESS
                     02C9   589
                     02C9   590 PROCESS:
55   53 0000'CF  C3     02C9   591          SUBL3   W^SWP$GL_BALBASE,R3,R5  ;NO. OF BYTES BEYOND FIRST BAL SET PAGE
                     02CF   592 GETPHDR:
```

```
                55   00000000'EF   C6   02CF    593          DIVL    SWP$GL_BSLOTSZ,R5       ;FORM PROCESS HEADER INDEX
                 55    55    F7 8F   78   02D6    594          ASHL    #-9,R5,R5               ;DIVIDE BY PAGE SIZE
                55   00000000'EF   C4   02DB    595          MULL    SWP$GL_BSLOTSZ,R5       ;CONVERT PROCESS INDEX
                    55    55    09   9C   02E2    596          ROTL    #9,R5,R5                ;MULL BY BYTES PER PAGE
                    55    0000'CF   C0   02E6    597          ADDL    W^SWP$GL_BALBASE,R5     ;TO PROCESS HEADER ADDRESS
                                    02EB    598 GOTPHDR:
                  52    0000'DF40   3C   02EB    599          MOVZWL  @W^PFN$AW_SWPVBN[R0],R2 ;IS SWPVBN SET?
                                 0C   12   02F1    600          BNEQ    40$                     ;BRANCH IF YES
                  52    0000'DF40   D0   02F3    601          MOVL    @W^PFN$AL_BAK[R0],R2    ;GET BACKING STORE ADDRESS
                                    02F9    602
                                    02F9    603          ASSUME  PFN$V_GBLBAK EQ PTE$V_TYP0+1
               51   52    02    16   EF   02F9    604          EXTZV   #PTE$V_TYP0,#2,R2,R1    ;GET BACKING STORE ADDRESS TYPE
                                 05   02FE    605          RSB
                                    02FF    606 40$:
                                    02FF    607          ASSUME  PFN$C_PROCESS EQ 0
                0000'DF40    07   93   02FF    608          BITB    #PFN$M_PAGTYP,@W^PFN$AB_TYPE[R0] ;REQUIRE SWPVBN PAGE TO BE PROCESS
                             04   12   0305    609          BNEQ    60$                     ;BRANCH IF NOT, ERROR
                        51   03   D0   0307    610          MOVL    #3,R1                   ;CODE FOR SWPVBN BACKING DESTINATION
                                 05   030A    611          RSB
                                    030B    612 60$:     BUG_CHECK BADSWPVBN,FATAL           ;SWAP VBN ONLY FOR PROCESS PAGES
```

```
                     030F    614          .SBTTL  PTESCAN - SCAN ADJACENT PTE'S
                     030F    615 ;
                     030F    616 ; PTESCAN, ROUTINE TO SCAN ADJACENT PAGE TABLE ENTRIES FOR TRANSITION
                     030F    617 ; PAGES THAT ARE ON THE MODIFIED PAGE LIST AND HAVE THE APPROPRIATE
                     030F    618 ; BACKING STOPE ADDRESS FOR CLUSTERING.
                     030F    619 ; A SCAN IS MADE BACKWARDS FROM THE SVAPTE AND THEN FORWARDS.
                     030F    620 ;
                     030F    621 ; CALLING SEQUENCE:
                     030F    622 ;
                     030F    623 ;       BSBW    PTESCAN
                     030F    624 ;
                     030F    625 ; INPUTS:
                     030F    626 ;
                     030F    627 ;       R3 = SVAPTE TO START SCANNING FROM
                     030F    628 ;       R5 = PROCESS HEADER ADDRESS ASSOCIATED WITH THAT SVAPTE
                     030F    629 ;       R6 = INDEX TO NEXT AVAILABLE SLOT IN MODIFIED PAGE WRITER
                     030F    630 ;            PTE AND PHVINDEX ARRAYS
                     030F    631 ;       R7 = PAGE FILE INDEX IF PAGE FILE PAGE, TYPO BIT CLEAR
                     030F    632 ;          = SECTION BACKING STORE ADDRESS IF TYPO BIT SET
                     030F    633 ;       R8 = PAGE FILE VBN TO USE IF SCANNING FOR PAGE FILE PAGES
                     030F    634 ;       R9 = MAXIMUM NUMBER OF PAGES TO FIND
                     030F    635 ;
                     030F    636 ; OUTPUTS:
                     030F    637 ;
                     030F    638 ;       R6 UPDATED TO POINT TO NEXT SLOT NOT USED
                     030F    639 ;       R8 = FIRST PAGE FILE VBN NOT USED
                     030F    640 ;       R4,R5,R7 PRESERVED
                     030F    641 ;       R0,R1,R2,R3 SCRATCHED
                     030F    642 ;       R9,R10,R11,AP SCRATCHED
                     030F    643 ;
                     030F    644 CHK_ACCESS:
     51    53   15   09   EF   030F    645          EXTZV   #VA$V_VPN,#VA$S_VPN,R3,R1 ;INDEX INTO SYSTEM PAGE TABLE
           0000'DF41   D5   0314    646          TSTL    @W^MMG$GL_SPTBASE[R1]   ;IS THIS PAGE OF PAGE TABLE ENTRIES VALID
                  6F   18   0319    647          BGEQ    SCAN_FORWARD            ;BRANCH IF NOT
                  18   11   031B    648          BRB     SCAN_NEXT1
                       031D    649 PTESCAN:
                  5C   53   D0   031D    650          MOVL    R3,AP                   ;SAVE SVAPTE
     51    53   15   09   EF   0320    651          EXTZV   #VA$V_VPN,#VA$S_VPN,R3,R1 ;KEEP PAGE NUMBER OF SVAPTE
                  5B   04   CE   0325    652          MNEGL   #4,R1T                  ;GOING BACKWARDS LOOKING AT PTE'S
                  58   11   0328    653          BRB     SCAN_AGAIN              ;LOOK AT NEXT PAGE TABLE ENTRY
                       032A    654 ;
                       032A    655 ; NOTE THAT R1 = PAGE NUMBER OF AN ACCESSABLE PTE
                       032A    656 ;
                       032A    657 SCAN_NEXT:
     53    5B   5A   C1   032A    658          ADDL3   R10,R11,R3              ;FORM NEXT SVAPTE TO CHECK
     51    53   15   09   ED   032E    659          CMPZV   #VA$V_VPN,#VA$S_VPN,R3,R1 ;PTE IN SAME PAGE?
                  DA   12   0333    660          BNEQ    CHK_ACCESS              ;BRANCH IF NOT, CHECK ACCESSABILITY
                       0335    661 SCAN_NEXT1:
50   63    7B800000 8F   CB   0335    662          BICL3   #^C<PTE$M_VALID !-
                       033D    663                  PTE$M_TYPT ! PTE$M_TYPO !-
                       033D    664                  PTE$M_PGFLVB>,(R3),R0   ;GET PFN IF TRANSITION PAGE
                  4B   13   033D    665          BEQL    SCAN_FORWARD            ;BRANCH IF DEMAND ZERO PAGE
     52    50    EA 8F   78   033F    666          ASHL    #-PTE$V_TYPO,R0,R2      ;VALID, TYP1, TYPO ALL ZERO?
                  44   12   0344    667          BNEQ    SCAN_FORWARD            ;BRANCH IF NOT TRANSITION PAGE
        0000'CF   50   D1   0346    668          CMPL    R0,W^MMG$GL_MAXPFN      ;IS THIS A LEGAL PFN?
                  3D   14   034B    669          BGTR    SCAN_FORWARD            ;NO (ALSO REJECTS WINDOW PAGES)
           03   00   ED   034D    670          CMPZV   #PFN$V_LOC,#PFN$S_LOC,- ;SEE IF ON MODIFIED PAGE LIST
```

```
                01   0000'DF40        0350   671                     @W^PFN$AB_STATE[R0],#PFN$C_MFYPAGLST
                               33  12  0355   672             BNEQ    SCAN_FORWARD                ;BRANCH IF NOT ON MODIFIED PAGE LIST
                     0000'DF40  B5  0357   673             TSTW    @W^PFN$AW_SWPVBN[R0]        ;IF SWAP VBN PAGE,
                               2C  12  035C   674             BNEQ    SCAN_FORWARD                ;DON'T USE IT
                0000'DF40      53  D1  035E   675             CMPL    R3,@W^PFN$AL_PTE[R0]        ;CHECK FOR SPURIOUS MATCH
                               24  12  0364   676             BNEQ    SCAN_FORWARD                ;BRANCH IF SPURIOUS
                52   0000'DF40  D0  0366   677             MOVL    @W^PFN$AL_BAK[R0],R2        ;GET BACKING STORE ADDRESS
                   0D 57   16  E0  036C   678             BBS     #PTE$V_TYPO,R7,60$          ;BRANCH IF SECTION ADDRESS
                                  0370   679 ;
                                  0370   680 ; PAGE FILE PAGE, REQUIRE ANOTHER PAGE FILE PAGE WITH SAME PAGE FILE INDEX
                                  0370   681 ;
                     16 52   16   E0  0370   682             BBS     #PTE$V_TYPO,R2,SCAN_FORWARD ;BRANCH IF SECTION PAGE
             57   52    08   18   ED  0374   683             CMPZV   #PFN$V_PGFLX,#PFN$S_PGFLX,R2,R7 ;SAME PAGE FILE INDEX?
                               07  13  0379   684             BEQL    SCAN_AGAIN                  ;BRANCH IF YES
                               0D  11  037B   685             BRB     SCAN_FORWARD                ;NOT SAME PAGE FILE
                                  037D   686 ;
                                  037D   687 ; SECTION PAGE, MUST CHECK FOR SECTION BACKING STORE TYPE AND SAME BACKING STORE
                                  037D   688 ;
                         57   52  D1  037D   689 60$:    CMPL    R2,R7                       ;SAME SECTION?
                               08  12  0380   690             BNEQ    SCAN_FORWARD                ;BRANCH IF NOT
                                  0382   691 SCAN_AGAIN:
                         5A    53  D0  0382   692             MOVL    R3,R10                      ;ADDRESS OF LAST PTE CHECKED
                           A2  59  F5  0385   693             SOBGTR  R9,SCAN_NEXT                ;BRANCH IF MORE PAGE FILE VBN'S TO USE
                                  0388   694 ;           INCL    L^MPW$L_BACKUPFAIL(R11) ;COUNT FAILURE FOR EITHER DIRECTION
                               10  11  0388   695             BRB     SCAN_DONE                   ;ALLOCATED PAGE FILE EXHAUSTED,
                                  038A   696                                                     ;WRITE THE CLUSTER
                                  038A   697 ;
                                  038A   698 ; PTE SCAN CANNOT PROCEED IN CURRENT DIRECTION, SWITCH TO SCAN FORWARD
                                  038A   699 ; IF NOT ALREADY SCANNING FORWARD
                                  038A   700 ;
                                  038A   701 SCAN_FORWARD:
                         5B    5B  CE  038A   702             MNEGL   R11,R11                     ;SWITCH DIRECTION OF PTE SCAN
                               0B  19  038D   703             BLSS    SCAN_DONE                   ;PTE SCAN COMPLETE IF ALREADY SCANNED FORWAR
                         53    5C  D0  038F   704             MOVL    AP,R3                       ;GET STARTING PTE ADDRESS
                         5C    5A  D0  0392   705             MOVL    R10,AP                      ;RECORD THIS AS STARTING PTE ADDRESS
                                  0395   706                                                     ;SINCE IT IS LEQU THAN STARTING PTE
                         5A    53  D0  0395   707             MOVL    R3,R10                      ;START FORWARD FROM ORIGINAL START PTE
                               90  11  0398   708             BRB     SCAN_NEXT                   ;CONTINUE THE PTE SCAN
                                  039A   709 ;
                                  039A   710 ; AP = SVAPTE, R10 IS THE OTHER SVAPTE, NOT NECESSARILY IN ORDER
                                  039A   711 ; R7 = PAGE FILE INDEX OR SECTION BACKING STORE ADDRESS
                                  039A   712 ;       IF PTE$V_TYPO IS SET, NO BACKING STORE MANIPULATION
                                  039A   713 ; R9 PRESERVED FROM HERE ON
                                  039A   714 ; R10 PRESERVED FROM HERE ON IF AP LEQU R10
                                  039A   715 ;
                                  039A   716 SCAN_DONE:
                         5A    5C  D1  039A   717             CMPL    AP,R10                      ;GET PTE ADDRESSES IN ORDER
                               09  1B  039D   718             BLEQU   40$                         ;BRANCH IF R10 IS TOP OF RANGE
                         50    5A  D0  039F   719             MOVL    R10,R0                      ;SAVE BOTTOM OF RANGE
                         5A    5C  D0  03A2   720             MOVL    AP,R10                      ;HIGH END OF RANGE
                         5C    50  D0  03A5   721             MOVL    R0,AP                       ;LOW END OF RANGE
                                  03A8   722 ;
                                  03A8   723 ; AP = FIRST PTE ADDRESS, R10 = LAST PTE ADDRESS INCLUSIVE
                                  03A8   724 ;       ALL PFN'S IN THESE PTE'S ARE ON THE MODIFIED PAGE LIST AND
                                  03A8   725 ;       ARE FROM THE SAME PAGE FILE OR SECTION
                                  03A8   726 ;
          5B    8C    FFE00000  8F  CB  03A8   727 40$:    BICL3   #^C<PTE$M_PFN>,(AP)+,R11 ;GET PAGE FRAME NUMBER
```

```
                0000'DF46    5B   D0  03B0   728          MOVL    R11,@W^MPW$AL_PTE[R6]    ;STORE PFN IN PTE ARRAY
           0004'DF46    42 A5   B0  03B6   729          MOVW    PHD$W_PHVINDEX(R5),@W^MPW$AW_PHVINDEX[R6] ;STORE THE ASSOCIATED
                        56   D6  03BD   730          INCL    R6                       ;NEXT PTE INDEX
                                 03BF   731                                           ;PROCESS HEADER VECTOR INDEX
                                 03BF   732 ;        INCL    MPW$L_COUNT              ;COUNT PAGES IN THIS CLUSTER
                  19 57    16   E0  03BF   733          BBS     #PTE$V_TYP0,R7,65$       ;BRANCH IF SECTION PAGE
      0000'DF4B    007FFFFF 8F   D3  03C3   734          BITL    #PFN$M_BAK,@W^PFN$AL_BAK[R11] ;NO BACKING STORE?
                        43   12  03CD   735          BNEQ    100$                     ;BRANCH IF THERE IS ONE, BUGCHECK
                52   57   18   78  03CF   736          ASHL    #PFN$V_PGFLX,R7,R2       ;PAGE FILE INDEX TO ITS BACKING STORE FIELD
           0000'DF4B    58   52   C9  03D3   737          BISL3   R2,R8,@W^PFN$AL_BAK[R11] ;AND RECORD IT IN THE BACKING STORE ADR
                        58   D6  03DA   738          INCL    R8                       ;NEXT PAGE FILE VBN
                        50   5B  03DC   739 65$:     MOVL    R11,R0                   ;PFN TO CONVENTIONAL REGISTER
                        52   01  9A  03DF   740          MOVZBL  #PFN$C_MFYPAGLST,R2      ;INDEX TO MODIFIED PAGE LIST
                           FC1B' 30  03E2   741          BSBW    MMG$REMPFN               ;REMOVE PAGE FROM MODIFIED PAGE LIST
                     87 8F   8B  03E5   742          BICB3   #<PFN$M_MODIFY ! PFN$M_LOC>,- ;SHUT OFF MODIFY BIT
                51   0000'DF40       03E8   743                  @W^PFN$AB_STATE[R0],R1
           0000'DF40    51   05  89  03ED   744          BISB3   #PFN$C_WRTINPROG,R1,@W^PFN$AB_STATE[R0] ;SET WRITE IN PROGRESS
                     0000'DF40   B6  03F4   745          INCW    @W^PFN$AW_REFCNT[R0]     ;AND COUNT AN I/O REFERENCE
                        03   00  ED  03F9   746          CMPZV   #PFN$V_PAGTYP,#PFN$S_PAGTYP,- ;IF PROCESS PAGE TABLE PAGE
                04   0000'DF40       03FC   747                  @W^PFN$AB_TYPE[R0],#PFN$C_PPGTBL
                        09   12  0401   748          BNEQ    80$
                     51   42 A5   3C  0403   749          MOVZWL  PHD$W_PHVINDEX(R5),R1    ;THEN MUST COUNT A PROCESS HEADER REF
                      0000'DF41   B6  0407   750          INCW    @W^PHV$GL_REFCBAS[R1]
                        5A   5C  D1  040C   751 80$:     CMPL    AP,R10                   ;DONE LAST PTE IN RANGE?
                        97   1B  040F   752          BLEQU   40$                      ;BRANCH IF MORE TO DO
                             05  0411   753          RSB
                                 0412   754
                                 0412   755 100$:    BUG_CHECK MODRELNBAK,FATAL       ;BACKING STORE VBN FOR MODIFIED PAGE
                                 0416   756
                                 0416   757          .END
```

```
...PFN                  = 000001C4 R      03
BADBAKADR                 00000157 R      03
BADTYP                    000002B2 R      03
BIT...                  = 00000003
BUG$_BADPAGTYPE           ******** X    03
BUG$_BADSWPVBN            ******** X    03
BUG$_IVBAKADIO            ******** X    03
BUG$_MODRELNBAK           ******** X    03
BUG$_MPWALCIRP            ******** X    03
BUG$_PROCGONE             ******** X    03
CA$_MEASURE             = 00000002
CHK_ACCESS                0000030F R      03
EVT$_PFCOM                ******** X    03
EXE$ALLOCIRP              ******** X    03
EXE$BLDPKTSWPW            ******** X    03
EXE$DEANONPAGED           ******** X    03
GETPFNCTX                 00000294 R      03
GETPHDR                   000002CF R      03
GET_IRP                   000000C0 R      03
GET_NXT_CLUSTER           00000120 R      03
GET_PAGE_TYPE             00000144 R      03
GOTPHDR                   000002EB R      03
GOT_IRP                   00000139 R      03
IOC$GL_IRPFL              ******** X    03
IPL$_SYNCH              = 00000008
IRP$B_PRI               = 00000023
IRP$C_LENGTH            = 000000C4
IRP$L_ASTPRM            = 00000014
IRP$L_IOST1             = 00000038
IRP$W_OBCNT             = 00000044
IRP$W_SIZE              = 00000008
MMG$ALLOCPAGFIL1          ******** X    03
MMG$ALLOCPAGFIL2          ******** X    03
MMG$DEALLOCPAGFIL         ******** X    03
MMG$DECPHDREF1            ******** X    03
MMG$GL_MAXPFN             ******** X    03
MMG$GL_PAGSLPVC           ******** X    03
MMG$GL_SPTBASE            ******** X    03
MMG$GL_SYSPHD             ******** X    03
MMG$INIBLDPKT             ******** X    03
MMG$INSPFN                ******** X    03
MMG$RFFCNTNEG             ******** X    03
MMG$RELPFN                ******** X    03
MMG$REMPFN                ******** X    03
MMG$WRTMFYPAG             00000115 RG     03
MPW$AL_PTE                00000000 RG     02
MPW$AW_PHVINDEX           00000004 RG     02
MPW$GB_PRIO               ******** X    03
MPW$GL_BADPAGTOTAL        00000008 RG     02
MPW$GW_MPWPFC             ******** X    03
MPW$M_BADPAG            = 00000002
MPW$M_NOTDONE           = 00000004
MPW$M_SUCCESS           = 00000001
MPW$V_BADPAG            = 00000001
MPW$V_NOTDONE           = 00000002
MPW$V_SUCCESS           = 00000000
MPW_BLDPKT                000001FB R      03
NEXT_MFYPAG               00000162 R      03
NOMOREPAGES               000000D0 R      03
NOSPACE                   000000CA R      03
NO_PAGEFILE               00000158 R      03
OP$_MOVL                = 000000D0
OP$_MOVZWL              = 0000003C
PAGEFILE                  0000016A R      03
PAGFILCLUSTER             000001A4 R      03
PCB$L_STS               = 00000024
PCB$L_WSSWP             = 00000020
PCB$V_SWPVBN            = 00000010
PCB$V_WAKEPEN           = 0000000C
PCB$W_APTCNT            = 00000030
PFL$B_ALLOCSIZ          = 00000022
PFL$L_STARTBYTE         = 00000004
PFL$L_VBN               = 00000010
PFL$L_WINDOW            = 0000000C
PFN$AB_STATE              ********   X    03
PFN$AB_TYPE               ********   X    03
PFN$AL_BAK                ********   X    03
PFN$AL_MFYLSTHD           ********   X    03
PFN$AL_PTE                ********   X    03
PFN$AW_REFCNT             ********   X    03
PFN$AW_SWPVBN             ********   X    03
PFN$AX_BLINK              ********   X    03
PFN$AX_FLINK              ********   X    03
PFN$C_BADPAGLST         = 00000002
PFN$C_GBLWRT            = 00000003
PFN$C_GLOBAL            = 00000002
PFN$C_GPGTBL            = 00000005
PFN$C_MFYPAGLST         = 00000001
PFN$C_PPGTBL            = 00000004
PFN$C_PROCESS           = 00000000
PFN$C_SYSTEM            = 00000001
PFN$C_WRTINPROG         = 00000005
PFN$M_BAK               = 007FFFFF
PFN$M_LOC               = 00000007
PFN$M_MODIFY            = 00000080
PFN$M_PAGTYP            = 00000007
PFN$S_LOC               = 00000003
PFN$S_PAGTYP            = 00000003
PFN$S_PGFLX             = 00000008
PFN$V_GBLBAK            = 00000017
PFN$V_LOC               = 00000000
PFN$V_PAGTYP            = 00000000
PFN$V_PGFLX             = 00000018
PFN$V_RPTEVT            = 00000006
PHD$W_PHVINDEX          = 00000042
PHDR                      000002BD R      03
PHV$GL_PIXBAS             ********   X    03
PHV$GL_REFCBAS            ********   X    03
PMS$GL_PWRITES            ********   X    03
PMS$GL_PWRITIO            ********   X    03
PR$_IPL                 = 00000012
PRI$_IOCOM              = 00000001
PROCESS                   000002C9 R      03
PROCESS_GONE              00000290 R      03
```

```
PTE$M_PFN                       = 001FFFFF
PTE$M_PGFLVB                    = 003FFFFF
PTE$M_TYP0                      = 00400000
PTE$M_TYP1                      = 04000000
PTE$M_VALID                     = 80000000
PTE$S_PFN                       = 00000015
PTE$V_PFN                       = 00000000
PTE$V_TYP0                      = 00000016
PTESCAN                           0000031D R      03
RSN$_MPLEMPTY                   = 0000000B
RSN$_MPWBUSY                    = 0000000C
SCAN_AGAIN                        00000382 R      03
SCAN_DONE                         0000039A R      03
SCAN_FORWARD                      0000038A R      03
SCAN_NEXT                         0000032A R      03
SCAN_NEXT1                        00000335 R      03
SCH$GB_SIP                        ********    X   03
SCH$GL_CURPCB                     ********    X   03
SCH$GL_MFYCNT                     ********    X   03
SCH$GL_MFYLIM                     ********    X   03
SCH$GL_MFYLIMSV                   ********    X   03
SCH$GL_MFYLOLIM                   ********    X   03
SCH$GL_MFYLOSV                    ********    X   03
SCH$GL_PCBVEC                     ********    X   03
SCH$RAVAIL                        ********    X   03
SCH$RSE                           ********    X   03
SCH$V_MPW                         ********    X   03
SCH$V_SIP                         ********    X   03
SEC$L_VBN                       = 00000010
SEC$L_WINDOW                    = 0000000C
SECTION                           00000228 R      03
SIZ...                          = 00000001
SWP$GL_BALBASE                    ********    X   03
SWP$GL_BALSPT                     ********    X   03
SWP$GL_BSLOTSZ                    ********    X   03
SWPVBN                            00000249 R      03
SYSPHD                            00000286 R      03
TMP...                          = 00000001
VA$S_VPN                        = 00000015
VA$V_VPN                        = 00000009
WRITEDONE                         00000000 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
$$$210	0000000C (12.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
$MMGCOD	00000416 (1046.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
Z$INIT$PFN_FIXUP_TABLE	00000012 (18.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:00.89
Command processing	106	00:00:00.55	00:00:05.94
Pass 1	332	00:00:11.69	00:00:45.24
Symbol table sort	0	00:00:01.79	00:00:05.11
Pass 2	151	00:00:02.68	00:00:09.81
Symbol table output	19	00:00:00.14	00:00:00.23
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	641	00:00:16.94	00:01:07.25

The working set limit was 1500 pages.
68469 bytes (134 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1159 non-local and 39 local symbols.
757 source lines were read in Pass 1, producing 24 object records in Pass 2.
32 pages of virtual memory were used to define 30 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	19
_255DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	27

1289 GETS were required to define 27 macros.

There were no errors, warnings or information messages.

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

WRTMFYPAG
LIS

UCBCREDEL
LIS

USRVECTOR
LIS

SYSVECTOR
LIS

SYSINIT
LIS

VERSION
LIS

SYSINI

UPCASEDAT
LIS

SYSINIT
MAP

SYSWAIT
LIS

TIMESCHDL
LIS