L 12

```
   SSSSSSSSSSSS   YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS   YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS   YYY         YYY     SSSSSSSSSSSS
SSS               YYY         YYY  SSS
SSS               YYY         YYY  SSS
SSS               YYY         YYY  SSS
SSS                  YYY   YYY     SSS
SSS                  YYY   YYY     SSS
SSS                  YYY   YYY     SSS
   SSSSSSSSS            YYY           SSSSSSSSS
   SSSSSSSSS            YYY           SSSSSSSSS
   SSSSSSSSS            YYY           SSSSSSSSS
            SSS         YYY                    SSS
            SSS         YYY                    SSS
            SSS         YYY                    SSS
            SSS         YYY                    SSS
            SSS         YYY                    SSS
SSSSSSSSSSSS            YYY        SSSSSSSSSSSS
SSSSSSSSSSSS            YYY        SSSSSSSSSSSS
SSSSSSSSSSSS            YYY        SSSSSSSSSSSS
```

FILEID**SHELL

B 8

```
  SSSSSSSS  HH      HH  EEEEEEEEEE  LL          LL
  SSSSSSSS  HH      HH  EEEEEEEEEE  LL          LL
SS          HH      HH  EE          LL          LL
SS          HH      HH  EE          LL          LL
SS          HH      HH  EE          LL          LL
SS          HH      HH  EE          LL          LL
  SSSSSS    HHHHHHHHHH  EEEEEEEE    LL          LL
  SSSSSS    HHHHHHHHHH  EEEEEEEE    LL          LL
        SS  HH      HH  EE          LL          LL
        SS  HH      HH  EE          LL          LL
        SS  HH      HH  EE          LL          LL               ....
        SS  HH      HH  EE          LL          LL               ....
SSSSSSSS    HH      HH  EEEEEEEEEE  LLLLLLLLLL  LLLLLLLLLL       ....
SSSSSSSS    HH      HH  EEEEEEEEEE  LLLLLLLLLL  LLLLLLLLLL       ....

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
```

SHE
V04

Table of contents

(1)	216	DECLARATIONS
(1)	354	SHELL PROCESS HEADER
(1)	450	SHELL PROCESS PAGE TABLES
(1)	553	BODY OF SHELL PROCESS
(1)	880	PROCESS I/O SEGMENT
(1)	1049	INITIALZE SHELL WITH SYSGEN PARAMETERS
(1)	1302	FILLPHD - SETUP A VALID PHD PTE
(1)	1371	COMMAND LANGUAGE INTERPRETER DATA AREA
(1)	1405	COMPATIBILITY MODE EMULATOR CONTEXT PAGE
(1)	1414	GLOBAL USER-MODE CONTEXT PAGE
(1)	1434	IMAGE ACTIVATOR CONTEXT PAGE

SHE
V04

```
0000            .IF     NDF,LIBSWITCH
0000            .TITLE  SHELL PROCESS DEFINITION
0000            .IFF
0000            .TITLE  SYS$USRVECTOR - USER ACCESSIBLE P1 CELLS
0000            .ENDC
0000            .IDENT  'V04-000'
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000
0000    ;++
0000    ; FACILITY:     EXECUTIVE, CREATE PROCESS/SWAPPER DATA BASE
0000    ;
0000    ; ABSTRACT:     SHELL DEFINES THE SWAP IMAGE FOR THE INITIAL PROCESS
0000    ;               STARTED BY THE CREATE PROCESS SYSTEM SERVICE.
0000    ;
0000    ; ENVIRONMENT:
0000    ;
0000    ; AUTHOR: RICHARD I. HUSTVEDT, CREATION DATE: 30-NOV-76
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-039 RAS0319         Ron Schaefer            29-Jun-1984
0000    ;               Add queue header cells for logical name table name
0000    ;               translation cache.
0000    ;
0000    ;       V03-038 LJK0287         Lawrence J. Kenah       25-Jun-1984
0000    ;               Add three longword array to contain the counts of termination
0000    ;               handlers declared in each of the three outer access modes.
0000    ;               Add sequence number cell for process private logical name
0000    ;               directory.
0000    ;
0000    ;       V03-037 MSH0054         Michael S. Harvey       30-May-1984
0000    ;               Remove some obsolete symbols.
0000    ;
0000    ;       V03-036 LJK0282         Lawrence J. Kenah       9-May-1984
0000    ;               Add demand zero page after PIO page to accommodate long
```

```
0000 ;          directory names that straddle page boundary.
0000 ;
0000 ;  V03-035 TMK0001         Todd M. Katz            27-Apr-1984
0000 ;          Remove CTL$GL_LOGTBL which was required only under the old
0000 ;          logical name design.
0000 ;
0000 ;  V03-034 LJK0272         Lawrence J. Kenah       10-Apr-1984
0000 ;          Add image activator context cells privileged vectors
0000 ;          and shareable image initialization.
0000 ;
0000 ;  V03-033 MHB0133         Mark Bramhall            9-Apr-1984
0000 ;          Remove preset values for CTL$T_USERNAME and CTL$T_ACCOUNT.
0000 ;          Validate CLI information ordering.
0000 ;
0000 ;  V03-032 RAS0281         Ron Schaefer            28-Mar-1984
0000 ;          Add PIO$GB_DFNBC process-specific RMS parameter for
0000 ;          network block count transfer size.
0000 ;          Re-use the cell once used from RMS Compatibility.
0000 ;          Add CTL$GT_SPAWNCLI and CTL$GT_SPAWNTABLE as well.
0000 ;
0000 ;  V03-031 WMC0006         Wayne Cardoza           21-Mar-1984
0000 ;          Go back to 3 pages of kernel stack.
0000 ;          Add cells for default image I/O area.
0000 ;
0000 ;  V03-030 MSH0006         Michael S. Harvey        3-Feb-1984
0000 ;          Protect image header buffer page against USER WRITE.
0000 ;
0000 ;          Protect image activator scratch page from user by changing
0000 ;          its owner to EXEC.
0000 ;
0000 ;          Replace obsolete image rundown control flags with an image
0000 ;          list pointer for use by the debugger. This fixed cell pointer
0000 ;          allows the debugger to avoid linking against the executive.
0000 ;
0000 ;  V03-029 RSH0091         R. Scott Hanna          31-Jan-1984
0000 ;          Add dedicated P1 demand zero pages for the security auditing
0000 ;          Impure Data Table (IDT). Remove the old vector page pointer
0000 ;          to the IDT.
0000 ;
0000 ;  V03-028 LJK0258         Lawrence J. Kenah       18-Jan-1984
0000 ;          Correct problems introduced by LJK0257. In SHELINIT, only
0000 ;          reorder a number of pagesequal to the  pages in this module
0000 ;          that are NOT kernel stack pages. Convert a PIO page from
0000 ;          a page file page to demand zero.
0000 ;
0000 ;  V03-027 BLS0262         Benn Schreiber          16-Jan-1984
0000 ;          Correct .ADDRESS to be .LONG in P1 lookaside list links
0000 ;
0000 ;  V03-026 LJK0257         Lawrence J. Kenah        4-Jan-1984
0000 ;          A variety of changes.
0000 ;
0000 ;              Add CTL$GQ_TERMCHAR and CREPRC_FLAGS to P1 vector page.
0000 ;
0000 ;              Add UAF_FLAGS to CLI data page.
0000 ;
0000 ;              Change PQB$C_MAXDIRLEN references to PQB$S_DDSTRING.
0000 ;
```

```
0000  ;                        Add second page for compatibility mode context.
0000  ;
0000  ;                        Reorder SHELL pages to achieve demand zero compaction.
0000  ;
0000  ;                        Add listhead and demand zero pages for P1 pool lookaside
0000  ;                        list.
0000  ;
0000  ;                        Move page file index and characteristics from PQB to PCB.
0000  ;
0000  ;        V03-025 WMC0005          Wayne Cardoza          02-Dec-1983
0000  ;                PHD$W_BAK, PHD$W_WSLX are now longwords.
0000  ;
0000  ;        V03-024 WMC0004          Wayne Cardoza          28-Nov-1983
0000  ;                Move the kstack expansion WSL slots.
0000  ;
0000  ;        V03-023 WMC0003          Wayne Cardoza          13-Oct-1983
0000  ;                Reserve space for 4 extra kstack pages.
0000  ;
0000  ;        V03-022 LJK0251          Lawrence J. Kenah      7-Sep-1983
0000  ;                Add CLI name counted string to CLI data page.
0000  ;
0000  ;        V03-021 LJK0246          Lawrence J. Kenah      24-Aug-1983
0000  ;                Add cell to P1 pointer page that records the size of the
0000  ;                user stack.
0000  ;
0000  ;        V03-020 WMC0002          Wayne Cardoza          22-Aug-1983
0000  ;                Temporarily increase kernel stack to 4 pages.
0000  ;
0000  ;        V03-019 LJK0239          Lawrence J. Kenah      1-Aug-1983
0000  ;                Increase size of exec stack to 16 pages.
0000  ;
0000  ;        V03-018 RAS0159          Ron Schaefer           22-Jul-1983
0000  ;                Add symbol PIO$S_EODSTR to specify the length of the
0000  ;                $EOD string for SYS$INPUT.
0000  ;
0000  ;        V03-017 LJK0208          Lawrence J. Kenah      26-May-1983
0000  ;                Add listheads for image control block lists to image
0000  ;                activator context page. Change name of image activator
0000  ;                work area.
0000  ;
0000  ;        V03-016 RSH0017          R. Scott Hanna         21-May-1983
0000  ;                Change CTL$GL_AUDITCHAN to CTL$GL_NSA_IDTPTR. This cell
0000  ;                contains a pointer to the security auditing impure data
0000  ;                table. The audit channel number is now in the IDT.
0000  ;
0000  ;        V03-015 PCA1016          Paul C. Anagnostopoulos 28-Apr-1983
0000  ;                Add CTL$AG_CLITABLE, two longwords which specify the
0000  ;                virtual address range into which the CLI table is mapped.
0000  ;
0000  ;        V03-014 GAS0122          Gerry Smith            14-Apr-1983
0000  ;                Add CTL$GQ_HELPFLAGS, two longwords of information on
0000  ;                current help settings.  Also double the size of the
0000  ;                supervisor stack, to 32 pages.
0000  ;
0000  ;        V03-013 RSH0012          R. Scott Hanna         13-Mar-1983
0000  ;                Add the cell CTL$GL_AUDITCHAN to store the security
0000  ;                audit trail journal channel number.
```

```
0000     ;
0000     ;       V03-012 RSH0002         R. Scott Hanna          9-Feb-1983
0000     ;               Add CTL$GL_RDIPTR to support rights database system
0000     ;               services.
0000     ;
0000     ;       V03-011 KBT0454         Keith B. Thompson       7-Jan-1983
0000     ;               Make directory cache 2 pages due to long directory names.
0000     ;
0000     ;       V03-010 JWH0138         Jeffrey W. Horn         29-Nov-1982
0000     ;               Fix problems related with movement within the
0000     ;               vector page introduced by JWH0118 and DMW4008.
0000     ;
0000     ;       V03-009 JWH0118         Jeffrey W. Horn         19-Nov-1982
0000     ;               Re-organize RMS Impure Areas.  Add cells needed to
0000     ;               support P1 allocation routines.  Eliminate the
0000     ;               RMS Process IO segment and the Process  Allocation
0000     ;               Region from SHELL, now created in PROCSTRT.
0000     ;
0000     ;       V03-008 DMW4008         DMWalp                  12-Nov-1982
0000     ;               Add definations for new logical name structures.
0000     ;
0000     ;       V03-007 WMC0001         Wayne Cardoza           19-Oct-1982
0000     ;               Add support for specifying desired page file.
0000     ;
0000     ;       V03-006 JWH0001         Jeffrey W. Horn         27-Aug-1982
0000     ;               Add cells needed to support RMS Recovery Units.
0000     ;               PIO$GL_RULOCK and PIO$GL_NXTIRBSEQ.
0000     ;
0000     ;       V03-005 CDS0001         C Saether               13-Aug-1982
0000     ;               Define CTL$GL_F11BXQP cell to dispatch to XQP.
0000     ;
0000     ;       V03-004 LJK47269        Lawrence J. Kenah       10-Aug-1982
0000     ;               Change protection on image activator scratch pages to
0000     ;               UREW. Remove $PRDEF.
0000     ;
0000     ;       V03-004 KDM46395        Kathleen D. Morse       28-Jun-1982
0000     ;               Increment the PTWSLELCK byte array for the P1 system
0000     ;               service vector pages, which are WINDOW PTEs.
0000     ;
0000     ;       V03-002 KDM0002         Kathleen D. Morse       28-Jun-1982
0000     ;               Added $PRDEF.
0000     ;
0000     ;
```

```
                0000           .SBTTL  DECLARATIONS
                0000
                0000    ;
                0000    ; INCLUDE FILES:
                0000    ;
                0000           $DYNDEF                        ;DYNAMIC DATA STRUCTURE TYPE DEFINITIONS
                0000           $FWADEF                        ;RMS FILE WORK AREA CONSTANTS
                0000           $IAFDEF                        ;IMAGE ACTIVATOR FIXUP VECTOR OFFSETS
                0000           $IMPDEF                        ;RMS INTERNAL STRUCTURE DEFINITION
                0000           $IPLDEF                        ;DEFINE INTERRUPT LEVELS
                0000           $JIBDEF                        ;JOB INFORMATION BLOCK DEFINITIONS
                0000           $NSAIDTDEF                     ;SECURITY AUDITING IMPURE DATA TABLE OFFSETS
                0000           $PCBDEF                        ;PROCESS CONTROL BLOCK DEFINITIONS
                0000           $PFLDEF                        ;PAGE FILE DEFINITIONS
                0000           $PFNDEF                        ;PFN DATA BASE DEFINITIONS
                0000           $PHDDEF                        ;PROCESS HEADER DEFINITIONS
                0000           $PQBDEF                        ;PROCESS QUOTA BLOCK DEFINITIONS
                0000           $PRTDEF                        ;DEFINE PROTECTION CODES
                0000           $PSLDEF                        ;DEFINE PSL FIELDS
                0000           $PTEDEF                        ;PAGE TABLE ENTRY DEFINTIONS
                0000           $SECDEF                        ;SECTION DEFINITIONS (GSTE/PSTE)
                0000           $SGNDEF                        ;SYSGEN VALUE DEFINITIONS
                0000           $VADEF                         ;DEFINE VIRTUAL ADDRESS FIELDS
                0000           $WSLDEF                        ;WORKING SET LIST DEFINITIONS
                0000
                0000    ;
                0000    ; EQUATES:
                0000    ;
                0000
00000000        0000    PFILPGCNT=0                           ; INITIALIZE COUNT OF PAGE FILE PAGES
00000003        0000    KSTACK=3                              ; THREE PAGES OF KERNEL STACK
00000004        0000    KSTACK_EX=4                           ; KERNEL STACK EXPANSION PAGES
00000010        0000    ESTACK=16                             ; SIXTEEN PAGES OF EXEC STACK
00000020        0000    SSTACK=32                             ; THIRTY TWO PAGES OF SUPER STACK
00000008        0000    IMGACTBUF=8                           ; EIGHT PAGES FOR IMAGE ACTIVATE BUFFER
                0000                                          ; FOUR FOR TOP LEVEL CALL, FOUR FOR
                0000                                          ; ONE RECURSIVE CALL.
00000004        0000    KRP_COUNT=4                           ; SIZE IN PAGES OF P1 LOOKASIDE LIST
                0000
00000001        0000    DBGPTCNT=1                            ; NUMBER OF DEBUG AREA PAGE TABLES
                0000
                0000           .IF     NDF,LIBSWITCH          ; IF EXECUTIVE, DEFINE THESE GLOBALLY
00000001        0000    SWP$C_DBGPTCNT==DBGPTCNT
00000003        0000    SWP$C_KSTACK==KSTACK
00000004        0000    SWP$C_KSTACK_EX==KSTACK_EX
00000004        0000    CTL$C_KRP_COUNT==KRP_COUNT
00000200        0000    CTL$C_KRP_SIZE==512                   ; EACH KRP PACKET IS 512 BYTES LONG
                0000           .ENDC
                0000
00000004        0000    COMMON=4                              ; FOUR PAGES FOR USER COMMON
                0000                                          ; FOUR PAGES FOR DEC COMMON
                0000
                0000    ;
                0000    ; MACROS:
                0000    ;
                0000
                0000           .MACRO  WSL     SYM,TYPE       ; MAKE VALID WS LIST ENTRY
```

```
0000          .LONG   SYM!WSL$M_VALID!WSL$C_'TYPE      ; SET VALID BIT
0000          .ENDM   WSL
0000
0000          .MACRO  CTL      SYM
0000          .IF     NDF,LIBSWITCH
0000  CTL$'SYM==.-CTLVECPAG+VECTORS
0000          .IFF
0000  CTL$'SYM=.-CTLVECPAG+VECTORS
0000          .ENDC
0000          .ENDM   CTL
0000
0000          .MACRO  PHD      SYM
0000          .=PHD...+PHD$'SYM
0000          .ENDM   PHD
0000
0000
0000          .MACRO  PIO      SYM                  ; DEFINE PROCESS I/O SEGMENT SYMBOL
0000  PIO$'SYM==PIOBASE+<.-PIO>                     ;
0000          .ENDM   PIO
0000
0000          .MACRO  PIOL     SYM                  ; DEFINE PROCESS I/O LOCAL SYMBOL
0000  PIOL'SYM=PIOBASE+<.-PIO>                      ;
0000          .ENDM   PIOL                          ;
0000
0000          .MACRO  PIO2     SYM                  ; DEFINE PROCESS I/O LOCAL SYMBOL
0000  PIO2'SYM=PIOTBL+<.-TBL>
0000          .ENDM   PIO2
0000
0000  ;
0000  ; P1PTE - GENERATE PTES FOR CONTROL REGION PAGE TABLE
0000  ;
0000
0000          .MACRO  P1PTE    NUM,ACCESS=NA,OWNER=K,STATE,SYM
0000          .IF     NB,SYM
0000          SYM=VA...
0000          .ENDC
0000
0000          .REPT   NUM
0000          .=.-4
0000          .IF     IDN,<STATE>,<VALID>
0000          TMP...=<1@31>
0000          .IFF
0000          TMP...=0
0000          .ENDC
0000
0000          .IF     IDN,<STATE>,<DZRO>
0000          PFILPGCNT=PFILPGCNT+1
0000          .ENDC
0000
0000          .IF     IDN,<STATE>,<PFIL>
0000          TMP...=PTE$M_TYP1
0000          PFILPGCNT=PFILPGCNT+1
0000          .ENDC
0000          .LONG   PTE$C_'ACCESS!PTE$C_'OWNER'OWN!TMP...
0000
0000          .=.-4
0000          VA...=VA...-^X200
```

```
0000   330        .ENDR
0000   331        .ENDM    P1PTE
0000   332
```

```
                0000
                0000            .IF     DF,LIBSWITCH            ; FOR USRVECTOR MODULE,
                0000            .PSECT  $ABS$,ABS               ; DO NOT GENERATE ANY STORAGE
                0000            .=0
                0000            .IFF                            ; ELSE,
                0000
            00000000            .PSECT  AEXENONPAGED
                0000    SWP$GL_SHELIO::                         ; I/O PAGE COUNT FOR SHELL
   00000007'    0000            .LONG   SWP$C_SHELLSIZ          ;
                0004    ;
                0004    ;       WARNING: SWP$C_SHLP1PT MUST BE MANUALLY COMPUTED SINCE IT IS USED
                0004    ;                TO ALLOCATE THE SPACE FOR P1 PAGE TABLE ENTRIES.  IF UPDATES
                0004    ;                ARE MADE THAT INTRODUCE MORE P1 PAGE TABLE ENTRIES, THEN
                0004    ;                VERIFY THAT THEY STILL FIT IN THE NUMBER OF PAGES PROVIDED
                0004    ;
   00000002     0004    SWP$C_SHLP1PT==2                        ; COUNT OF BASIC P1 PAGE TABLES FOR SHELL
            00000000            .PSECT  YYSHELLPAGED,PAGE       ; SHELL AT END OF PAGEABLE EXEC
                0000    SHELL:                                  ; BASE OF SHELL SWAP IMAGE
                0000    SWP$GL_SHELLBAS::                       ;
```

```
                   0000          .SBTTL  SHELL PROCESS HEADER
                   0000  ;
                   0000  ;       PROCESS HEADER FOR SHELL PROCESS
                   0000  ;
                   0000
          00000000 0000  PHD...=.                                      ; BASE OF PROCESS HEADER
          0000017C 0000          .BLKB   PHD$C_LENGTH                  ; RESERVE SPACE FOR HEADER
                   017C
                   017C  ;
                   017C  ;       WORKING SET LIST
                   017C  ;
                   017C  ; PAGES DESCRIBED IN THE WORKING SET LIST MUST OCCUR IN THE SAME ORDER
                   017C  ; WITHIN THE PROCESS BODY.
                   017C  ;
                   017C          .ALIGN  LONG                          ; AT LONGWORD RESOLUTION
          0000005F 017C  SWP$C_KSTACK_EX_WSL==<.-PHD...>@-2
          0000018C 017C          .BLKL   KSTACK_EX                     ; EXTRA SLOTS FOR KERNEL STACK EXPANSION
          0000018C 018C  WSL...=.                                      ; BASE OF WORKING SET LIST
          00000063 018C  SWP$C_KSTACK_WSL==<.-PHD...>@-2
                   018C          WSL     <<KSPINI-<1*512>>!WSL$M_WSLOCK>,PROCESS; KERNEL STACK PAGE 1
                   0190          WSL     <<KSPINI-<2*512>>!WSL$M_WSLOCK>,PROCESS; KERNEL STACK PAGE 2
                   0194          WSL     <<KSPINI-<3*512>>!WSL$M_WSLOCK>,PROCESS; KERNEL STACK PAGE 3
          00000198 0198  NOT_KSTACK_WSL=.                              ; REST OF WSL IS NOT KERNEL STACK
                   0198          WSL     <VECTORS!WSL$M_WSLOCK>,PROCESS  ; VECTOR PAGE
          0000019C 019C  DYNWSL=.                                      ; START OF DYNAMIC WORKING SET
                   019C          WSL     <PIOBASE!WSL$M_MODIFY>,PROCESS  ; PROCESS I/O SEGMENT BASE PAGE 1
                   01A0
          000001A0 01A0  FWSL...=.                                     ; FREE WORKING SET LIST
          00000001 01A0  NDYN=<FWSL...-DYNWSL>@-2                      ; NUMBER OF DYNAMIC ENTRIES IN SHELL
          00000002 01A0  SHUFFLE_COUNT=<FWSL...-NOT_KSTACK_WSL>@-2 ; AMOUNT OF SHUFFLING BY SHELINIT
          00000001 01A0  SWP$C_NDYN==NDYN                              ; GLOBAL NAME FOR NUMBER OF DYNAMIC PAGES
                   01A0          .ALIGN  PAGE                          ; GET TO END OF PAGE
          00000200 0200  SAV...=.                                      ;
                   0200
          00000067 0200  TMP...=<DYNWSL-PHD...>@-2                     ; BASE OF FLUID WORKING SET LIST
                   0200
                   0200          PHD     W_WSLOCK                      ; POINTER TO START OF LOCKED PAGES
              0067 000C          .WORD   TMP...                        ;
                   000E
                   000E          PHD     W_WSDYN                       ; POINTER TO START OF DYNAMIC PAGES
              0067 000E          .WORD   TMP...                        ;
                   0010
                   0010          PHD     W_WSLIST                      ;
              0063 0008          .WORD   <WSL...-PHD...>@-2            ; START OF WORKING SET LIST
                   000A          PHD     W_WSNEXT                      ; NEXT WORKING SET ENTRY
              0067 0010          .WORD   <TMP...+NDYN-1>               ;
                   0012
                   0012          PHD     L_FREP1VA                     ; VA OF FIRST FREE PAGE IN P1 SPACE
         7FFE0000' 0030          .LONG   VA...-512                     ;
                   0034
                   0034          PHD     Q_PRIVMSK                     ; ENABLE ALL PRIVILEGES
 FFFFFFFF FFFFFFFF 0000          .LONG   -1,-1                         ;
                   0008
                   0008          PHD     W_QUANT                       ; QUANTUM OF ONE SECOND
              FF9C 003C          .WORD   -100                          ;
                   003E
                   003E          PHD     L_KSP                         ; STACK POINTERS
```

```
7FFE7E00' 0078    .LONG   KSPINI                    ; KERNEL STACK POINTER
7FFE9E00' 007C    .LONG   ESPINI                    ; EXECUTIVE STACK POINTER
7FFEDE00' 0080    .LONG   SSPINI                    ; SUPER STACK POINTER
          0084                                      ; NO INITIAL USER STACK
          0084
          0084    PHD     L_PC                      ; INITIAL PROGRAM COUNTER
00000000' 00C0    .LONG   EXE$PROCSTRT              ; START PROCESS
          00C4
          00C4    PHD     L_PSL                     ; PROGRAM STATUS LONGWORD
00020000  00C4    .LONG   IPL$_ASTDEL@PSL$V_IPL     ; MUST RUN AT ASTDEL
          00C8    PHD     L_P0BR                    ; P0 BASE REGISTER
00000000  00C8    .LONG   0                         ; RELATIVE OFFSET
          00CC
          00CC    PHD     L_P0LRASTL                ; NO AST PENDING, P0LR = 0
04000000  00CC    .LONG   <4@PHD$V_ASTLVL>          ;
          00D0
          00D0    PHD     L_P1BR                    ; SVA OF P1 PAGE TABLE BASE
FF800000  00D0    .LONG   -<1@23>                   ;
          00D4
          00D4    PHD     L_P1LR                    ; P1 LENGTH REGISTER
001FFF01' 00D4    .LONG   <1@21>-P1PTLEN            ;
          00D8
          00D8    PHD     L_FREPTECNT               ; FREE PTE'S BETWEEN P0 AND P1 PT
FFFFFF00' 002C    .LONG   -<1+<<1@31-VA...>@-9>>    ; (NUMBER OF AVAILABLE LONGWORDS
          0030
          0030    PHD     W_PRCLM                   ; SUB PROCESS LIMIT
    0004  003E    .WORD   4                         ; ALLOW FOUR
          0040
          0040    PHD     W_WSFLUID                 ; GUARANTEED FLUID PAGE COUNT
    000A  0074    .WORD   10                        ; GUESS AT TEN FOR NOW (WAG)
          0076
          0076    PHD     B_DFPFC                   ; DEFAULT PAGE FAULT CLUSTER
      10  0034    .BYTE   16                        ;
          0035
          0035    PHD     B_CPUMODE                 ; ACCESS MODE FIELD FOR CPU TIME LIMIT
      03  0060    .BYTE   PSL$C_USER                ;  EXPIRATION AST STARTS WITH USER MODE
          0061
00000200  0061    .=SAV...                          ; RESTORE LOCATION COUNTER
```

```
          0200         .SBTTL  SHELL PROCESS PAGE TABLES
          0200  ;
          0200  ;
          0200  ;      SHELL PROCESS PAGE TABLES
          0200  ;
          0200  ;
          0200
00000200  0200  P1PTBAS=.                                   ;
          0200
00000400  0200         TMP=<SWP$C_SHLP1PT*512>              ; ALLOCATE SPACE FOR PAGE TABLES
00000600  0200         .BLKB   TMP                          ; ALLOCATE SPACE FOR PAGE TABLES
00000600  0600  SAV...=.                                    ; REMEMBER BASE OF PAGE TABLES
          0600
          0600         .IFTF                                ; ASSEMBLE FOLLOWING FOR BOTH
          0600                                              ; SHELL AND USRVECTOR MODULES:
          0600
          0600  ;
          0600  ;      INITIALIZE VIRTUAL ADDRESS BASE
          0600  ;
80000000  0600         VA...=1@31                           ;
          0600
          0600  ;      P1PTE   128*DBGPTCNT                 ; DEBUGGER CONTEXT AREA
7FFF0000  0600         VA...=VA...-<DBGPTCNT*128*512>       ; ACCOUNT FOR IT IN VA...
7FFF0000  0600  DBGAREA=VA...                               ; ADDRESS OF DEBUG AREA
          0600         P1PTE   1,URKW,K,PFIL                ; VECTOR PAGE
7FFEFE00  05FC  VECTORS=VA...                               ;
          05FC
          05FC         .IFF                                 ; ASSEMBLE REST ONLY FOR SHELL MODULE:
          05FC
7FFEFE00  05FC  CTL$GL_VECTORS==VECTORS                     ; BASE OF VECTOR PAGE
          05FC         P1PTE   16                           ; PAGES FOR SYSTEM SERVICE VECTORS
7FFEDE00  05BC  P1SYSVECTORS==VA...
          05BC         P1PTE   SSTACK,URSW,S,DZRO,SSPINI         ; SUPERVISOR STACK
          053C         P1PTE   ESTACK,SREW,E,DZRO,ESPINI         ; EXECUTIVE STACK
          04FC         P1PTE   KSTACK,SRKW,K,PFIL,KSPINI         ; KERNEL STACK
7FFE7800  04F0  CTL$GL_KSTKBAS==VA...                       ; MAXIMUM TOP OF KERNEL STACK
          04F0         P1PTE   KSTACK_EX,NA,K,PFIL               ; EXPANSION FOR  KERNEL STACK
7FFE7000  04E0  CTL$GL_KSTKBASEXP==VA...                    ; MAXIMUM TOP OF EXPANSION KERNEL STACK
7FFE7E00  04E0  CTL$GL_KSPINI==KSPINI                       ; INITIAL KERNEL STACK
          04E0         P1PTE   1                            ; NULL STOPPER PAGE
          04DC         P1PTE   KRP_COUNT,URKW,K,DZRO        ; P1 LOOKASIDE LIST FOR KERNEL MODE
7FFE6600  04CC  CTL$GL_KRP==VA...                           ; ADDRESS USED BY PROCSTRT TO FIND IT
          04CC         P1PTE   1,URSW,E,DZRO                ; IMAGE HEADER BUFFER
7FFE6400  04C8  MMG$IMGHDRBUF==VA...                        ; ADDRESS OF IMAGE HEADER BUFFER
          04C8         P1PTE   2,UREW,K,DZRO                ; VECTORS FOR USER SYS SRV & ERR MSGS
7FFE6000  04C0  CTL$A_DISPVEC==VA...                        ; ADR OF VECTOR PAGES FOR SYS SRV/ERRMSG
          04C0         P1PTE   4,UW,U,DZRO                  ; DEBUGGER CONTEXT
          04B0         P1PTE   IMGACTBUF,UREW,E,DZRO        ; IMAGE ACTIVATOR SCRATCH PAGES (8)
7FFE4800  0490  IAC$AL_IMGACTBUF==VA...                     ; ADR OF IMAGE ACTIVATOR SCRATCH PAGES
7FFE4800  0490  CLIDATAEND=VA...
          0490         P1PTE   12,URSW,S,DZRO               ; COMMMAND INTERPRETER DATA PAGES
7FFE3000  0460  CLIDATAPAG=VA...                            ; BASE ADDRESS OF CLI DATA PAGE
          0460         P1PTE   1,UREW,E,DZRO                ; IMAGE ACTIVATOR CONTEXT
7FFE2E00  045C  IMGACTCTX=VA...
          045C         P1PTE   NSA$K_IDT_PAGES,KW,K,DZRO ; SECURITY AUDITING IMPURE DATA TABLE
7FFE2800  0450  NSA$T_IDT==VA...
          0450         P1PTE   2                            ; UNUSED PAGES
```

```
          0448  ;
          0448  ;          THE FOLLOWING PAGE IS A USER-MODE WRITABLE CONTEXT PAGE
          0448  ;          FOR MISC. USER-MODE PACKAGES THAT ARE MAINTAINED BY VMS.
          0448  ;          (SEE DETAILED ALLOCATION OF PAGE BELOW)
          0448  ;
          0448             P1PTE   1,UW,K,DZRO                ; GLOBAL USER-MODE OWN STORAGE PAGE
7FFE2200  0444  UWVECPAG=VA...
          0444  ;
          0444  ;          THE FOLLOWING PAGE IS A WRITABLE CONTEXT PAGE FOR
          0444  ;          THE USE OF THE COMPATIBILITY MODE EMULATOR AND EXCEPTION
          0444  ;
          0444             P1PTE   2,UW,K,DZRO                ; WRITABLE PAGES FOR COMPATIBILITY MODE
7FFE1E00  043C  CTL$AG_CMEDATA==VA...                         ;
          043C  ;
          043C  ;          THE FOLLOWING PAGES ARE FOR USE BY BASIC/BASIC-PLUS TO PROVIDE
          043C  ;          THE ''CORE COMMON'' REQUIRED TO PASS DATA WHEN CHAINING FROM IMAGE
          043C  ;          TO IMAGE.  AN IDENTICAL AREA, NEGATIVELY DISPLACED HAS BEEN ALLOCATED
          043C  ;          FOR USERS AND CSS.
          043C  ;
          043C             P1PTE   COMMON,UW,K,DZRO           ; DEC ''CORE COMMON'' PAGES
7FFE1600  042C  CTL$A_COMMON==VA...                           ; BASE ADDRESS
          042C             P1PTE   COMMON,UW,K,DZRO           ; USER ''CORE COMMON'' PAGES
          041C             P1PTE   1,UREW,E,DZRO              ; PROCESS IFB/IRB TABLES
7FFE0C00  0418  PIOTBL=VA...
          0418             P1PTE   2,UREW,E,DZRO              ; RMS DIRECTORY CACHE PAGES
7FFE0800  0410  PIO$A_DIRCACHE==VA...
          0410             P1PTE   1,UREW,E,DZRO              ; RMS TRACEPOINT PAGE
7FFE0600  040C  PIO$A_TRACE==VA...
          040C             P1PTE   1,UREW,E,DZRO              ; EXTENSION TO RMS POINTER PAGE
          0408             P1PTE   1,UREW,E,PFIL              ; RMS POINTER PAGE
7FFE0200  0404  PIOBASE=VA...                                 ;
7FFE0200  0404  USPINI=VA...                                  ; USER STACK AREA
          0404
00000404  0404  END...=.                                      ; END OF CONTROL REGION
          0404
000000FF  0404  P1PTLEN=<1@31-VA...>@-9                       ; ENTRIES IN P1PT
          0404
00000081  0404  SWP$C_SHLFPTE==<<.-P1PTBAS>@-2> ; FREE P1PT IN BASIC P1 PAGE TABLES
          0404  ;
          0404  ; RESTORE LOCATION POINTER
          0404  ;
00000600  0404             .=SAV...                           ; FINISHED WITH PAGE TABLES
00000068  0600             PFILPGCNT=PFILPGCNT-KSTACK-1       ; NO PAGE FILE FOR KERNEL STACK OR VECTORS
00000068  0600             SWP$C_SHELLPFIL==PFILPGCNT         ; DEFINE GLOBAL VALUE FOR SHELL PAGE FILE
          0600                                                ; REQUIREMENT
```

```
                0600         .SBTTL  BODY OF SHELL PROCESS
                0600   ;
                0600   ;       BODY OF SHELL PROCESS
                0600   ;
    00000600    0600           .=SAV...                         ; POSITION TO END OF HEADER
                0600   ;--------------------------------------------------------------------------
                0600   ;
                0600   ;       VECTOR PAGE
                0600   ;               ***** NOTE: The cells in this page must not move.  There are
                0600   ;               *****       facilities which locate cells via offsets from
                0600   ;               *****       the top of this page.  If you delete a cell
                0600   ;               *****       subsitute a .LONG 0 and comment it as SPARE.  If
                0600   ;               *****       you add a cell either replace a SPARE cell or
                0600   ;               *****       add it to the end.
                0600   ;
                0600   ;--------------------------------------------------------------------------
    00000600    0600   SWP$AL_PTRPAG==.-SHELL                   ; ADDRESS IN SWAPPER MAP
                0600
                0600           .IFT                             ; IF USRVECTOR MODULE,
                0600           .=VECTORS                        ; SHOW ACTUAL ADDRESSES IN LISTING
                0600
                0600           .IFTF                            ; ASSEMBLE FOLLOWING FOR BOTH
                0600                                            ; SHELL AND USRVECTOR MODULES:
                0600
    00000600    0600   CTLVECPAG=.                              ; BASE OF VECTOR PAGE
                0600           CTL     GW_NMIOCH                ; NUMBER OF CHANNELS
        0000    0600           .WORD   0                        ; FILLED IN IN PROCSTRT
                0602
                0602           CTL     GW_CHINDX                ; MAXIMUM CHANNEL INDEX
        0000    0602           .WORD   0                        ; FILLED IN IN PROCSTRT
                0604
                0604           CTL     GL_LNMHASH               ; process Logical NaMe HASH table
    00000000    0604           .LONG   0                        ;  pointer
                0608           CTL     GL_LNMDIRECT             ; process Logical NaMe DIRECTory
    00000000    0608           .LONG   0                        ;  pointer
                060C
                060C   ;
                060C   ;       ARRAY OF STACK POINTERS
                060C   ;
                060C   ;       THE -1 ENTRY OF CTL$AL_STACK IS USED TO DETERMINE THE MAXIMUM
                060C   ;       EXTENT OF THE KERNEL STACK.
                060C   ;
    7FFE7800    060C           .LONG   CTL$GL_KSTKBAS           ; MAXIMUM TOP OF KERNEL STACK
                0610           CTL     AL_STACK                 ; STACK POINTER RESTART ARRAY
    7FFE7E00    0610           .LONG   KSPINI                   ; KERNEL STACK POINTER
    7FFE9E00    0614           .LONG   ESPINI                   ; EXEC STACK POINTER
    7FFEDE00    0618           .LONG   SSPINI                   ; SUPERVISOR STACK POINTER
    7FFE0200    061C           .LONG   USPINI                   ; USER STACK POINTER
                0620
                0620   ;
                0620   ; LOGICAL NAME TABLE TRANSLATION CACHE QUEUE HEADER
                0620   ;
                0620
                0620           CTL     GQ_LNMTBLCACHE           ; QUEUE HEADER FOR LOGICAL NAME TABLE
    7FFEFE20    0620           .LONG   CTL$GQ_LNMTBLCACHE       ;  TRANSLATION CACHE (FORWARD LINK)
    7FFEFE20    0624           .LONG   CTL$GQ_LNMTBLCACHE       ;                    (BACK LINK)
                0628
```

```
          0628    ;
          0628    ; THE FOLLOWING THREE VECTORS MUST BE ADJACENT
          0628    ;
          0628
          0628            CTL     GL_CMSUPR                       ; SUPERVISOR CHANGE MODE DISPATCHER
00000000  0628            .LONG   0                               ; NULL
          062C
          062C            CTL     GL_CMUSER                       ; USER CHANGE MODE DISPATCHER
00000000  062C            .LONG   0                               ; NULL
          0630
          0630            CTL     GL_CMHANDLR                     ; COMPATIBILITY MODE HANDLER
00000000  0630            .LONG   0
          0634
          0634            CTL     AQ_EXCVEC                       ; PRIMARY/SECONDARY EXCEPTION VECTORS
00000000  0634            .LONG   0                               ; KERNEL MODE PRIMARY
00000000  0638            .LONG   0                               ; KERNEL MODE SECONDARY
00000000  063C            .LONG   0                               ; EXEC MODE PRIMARY
00000000  0640            .LONG   0                               ; EXEC MODE SECONDARY
00000000  0644            .LONG   0                               ; SUPERVISOR MODE PRIMARY
00000000  0648            .LONG   0                               ; SUPERVISOR MODE SECONDARY
00000000  064C            .LONG   0                               ; USER MODE PRIMARY
00000000  0650            .LONG   0                               ; USER MODE SECONDARY
          0654
          0654            CTL     GL_THEXEC                       ; EXECUTIVE TERMINATION HANDLER
00000000  0654            .LONG   0                               ; NULL
          0658
          0658            CTL     GL_THSUPR                       ; SUPERVISOR TERMINATION HANDLER
00000000  0658            .LONG   0                               ; NULL
          065C
          065C    ;       CTL     GL_THUSER                       ; USER TERMINATION HANDLER
00000000  065C            .LONG   0                               ; NULL
          0660
          0660    ;       FORMER LOCATION OF COMPATIBILITY MODE CONTEXT
          0660            CTL     GQ_COMMON                       ; CORE COMMON DESCRIPTOR
00000800  0660            .LONG   512*COMMON                      ; SIZE IN BYTES
7FFE1600  0664            .LONG   CTL$A_COMMON                    ; AND ADDRESS
          0668
          0668            CTL     GL_GETMSG                       ; PER-PROCESS VECTOR TO USER MESSAGE
00000000  0668            .LONG   0                               ; DISPATCHER
          066C
          066C            CTL     AL_STACKLIM                     ; STACK LIMIT ARRAY (INDEXED BY MODE)
7FFE7800  066C            .LONG   CTL$GL_KSTKBAS                  ; KERNEL STACK LO (TOP) LIMIT
7FFE7E00  0670            .LONG   KSPINI                          ; EXEC   STACK LO (TOP) LIMIT
7FFE9E00  0674            .LONG   ESPINI                          ; SUPER  STACK LO (TOP) LIMIT
00000000  0678            .LONG   0                               ; USER STACK (NOT CHECKED)
          067C
          067C            CTL     GL_CTLBASVA                     ; BASE CONTROL REGION ADDRESS
00000000  067C            .LONG   0                               ; FILLED IN BY PROCSTRT
          0680
          0680            CTL     GL_IMGHDRBF                     ; ADDRESS OF IMAGE ACTIVATOR'S IMAGE
          0680                                                    ; HEADER BUFFER, IF IMAGE IS ACTIVE;
00000000  0680            .LONG   0                               ; 0 IF NO IMAGE ACTIVE
          0684
          0684            CTL     GL_IMGLSTPTR                    ; ADDRESS OF ICB LIST (FOR DEBUGGER)
7FFE2E7C' 0684            .LONG   IAC$GL_IMAGE_LIST
          0688
          0688            CTL     GL_PHD                          ; ADDRESS OF PHD WINDOW
```

```
         00000000  0688        .LONG   0                           ; INIT BY INSWAP
                   068C
                   068C        CTL     GQ_ALLOCREG                 ; HEAD OF PROCESS ALLOCATION REGION POOL
00000000 00000000  068C        .LONG   0,0                         ;       (Filled in by PROCSTRT)
                   0694
                   0694        CTL     GQ_MOUNTLST                 ; MOUNTED DEVICE LIST
         7FFEFE94  0694        .LONG   CTL$GQ_MOUNTLST             ; FORWARD LINK
         7FFEFE94  0698        .LONG   CTL$GQ_MOUNTLST             ; BACKWARD LINK ( LIST EMPTY )
                   069C
                   069C ;***************************************************************************
                   069C ; NOTE: The order of the accounting data between CTL$T_USERNAME and
                   069C ;       CTL$T_REMOTEID should be preserved!
                   069C ;***************************************************************************
                   069C
                   069C        CTL     T_USERNAME                  ; USERNAME
         000006A8  069C        .BLKB   JIB$S_USERNAME              ;  Filled in by PROCSTRT
                   06A8
                   06A8        CTL     T_ACCOUNT                   ; ACCOUNT NAME
         000006B0  06A8        .BLKB   JIB$S_ACCOUNT               ;  Filled in by PROCSTRT
                   06B0
                   06B0 ASSUME <CTL$T_ACCOUNT - CTL$T_USERNAME> EQ <JIB$T_ACCOUNT - JIB$T_USERNAME>
                   06B0
                   06B0        CTL     GQ_LOGIN                    ; SYSTEM TIME AT PROCESS CREATION
00000000 00000000  06B0        .LONG   0,0                         ;
                   06B8
                   06B8        CTL     GL_FINALSTS                 ; FINAL EXIT STATUS FOR PROCESS
         00000000  06B8        .LONG   0                           ;
                   06BC
                   06BC        CTL     GL_WSPEAK                   ; PEAK WORKING SET SIZE
         00000000  06BC        .LONG   0                           ;
                   06C0
                   06C0        CTL     GL_VIRTPEAK                 ; PEAK VIRTUAL SIZE
         00000000  06C0        .LONG   0                           ;
                   06C4
                   06C4        CTL     GL_VOLUMES                  ; COUNT OF VOLUMES MOUNTED
         00000000  06C4        .LONG   0                           ;
                   06C8
                   06C8        CTL     GQ_ISTART                   ; IMAGE ACTIVATION TIME
00000000 00000000  06C8        .LONG   0,0
                   06D0
                   06D0        CTL     GL_ICPUTIM                  ; INITIAL IMAGE CPU TIME
         00000000  06D0        .LONG   0
                   06D4
                   06D4        CTL     GL_IFAULTS                  ; INITIAL IMAGE FAULT COUNT
         00000000  06D4        .LONG   0
                   06D8
                   06D8        CTL     GL_IFAULTIO                 ; INITIAL IMAGE FAULT I/O COUNT
         00000000  06D8        .LONG   0
                   06DC
                   06DC        CTL     GL_IWSPEAK                  ; IMAGE WORKING SET PEAK
         00000000  06DC        .LONG   0
                   06E0
                   06E0        CTL     GL_IPAGEFL                  ; IMAGE PAGE FILE PEAK USAGE
         00000000  06E0        .LONG   0
                   06E4
                   06E4        CTL     GL_IDIOCNT                  ; INITIAL IMAGE DIRECT I/O COUNT
         00000000  06E4        .LONG   0
```

```
                            06E8
                            06E8       CTL     GL_IBIOCNT                 ; INITIAL IMAGE BUFFERED I/O COUNT
                   00000000 06E8       .LONG   0
                            06EC
                            06EC       CTL     GL_IVOLUMES                ; INITIAL IMAGE VOLUME MOUNT COUNT
                   00000000 06EC       .LONG   0
                            06F0
                            06F0       CTL     T_NODEADDR                 ; REMOTE NODE ADDRESS (BINARY)
                         00 06F0       .BYTE   0                                  ; BYTE COUNT BYTE
                   000006F7 06F1       .BLKB   6                                  ; 6 BYTES MAX
                            06F7
                            06F7       CTL     T_NODENAME                 ; REMOTE NODE NAME (ASCII)
                         00 06F7       .BYTE   0                                  ; BYTE COUNT BYTE
                   000006FE 06F8       .BLKB   6                                  ; 6 CHARACTERS MAX
                            06FE
                            06FE       CTL     T_REMOTEID                 ; REMOTE ID
                         00 06FE       .BYTE   0                                  ; BYTE COUNT BYTE
                   0000070F 06FF       .BLKB   16                                 ; 16 CHARACTERS MAX
                            070F
                   00000710 070F       .BLKB   1                          ; LONGWORD ALIGN SECTION
                            0710
                            0710  ;*********************************
                            0710  ; End of adjacent accounting data
                            0710  ;*********************************
                            0710
                            0710       CTL     GQ_PROCPRIV                ; PROCESS PRIVILEGE MASK
          FFFFFFFF FFFFFFFF 0710       .LONG   -1,-1                      ;  ALL PRIVILEGES PERMITTED
                            0718
                            0718       CTL     GL_USRCHMK                 ; PER-PROCESS VECTOR TO USER CHANGE MODE
                   00000000 0718       .LONG   0                          ; TO KERNEL HANDLER
                            071C
                            071C       CTL     GL_USRCHME                 ; PER-PROCESS VECTOR TO USER CHANGE MODE
                   00000000 071C       .LONG   0                          ; TO EXECUTIVE HANDLER
                            0720
                            0720       CTL     GL_POWERAST                ; POWER FAIL AST ADDRESS
                   00000000 0720       .LONG   0                          ;
                            0724       CTL     GB_PWRMODE                 ; ACCESS MODE FOR POWER FAIL AST
                         00 0724       .BYTE   0                          ;
                            0725
                            0725       CTL     GB_SSFILTER                ; SYS SERV INHIBIT FILTER MASK
                         00 0725       .BYTE   0
                   00000728 0726       .BLKB   2                          ; SPARE
                            0728
```

```
                    0728
                    0728   ;
                    0728   ;       ARRAY OF FINAL EXCEPTION VECTORS - ONE PER MODE
                    0728   ;
                    0728   ;       THESE VECTORS ARE USED TO LOCATE AN EXCEPTION HANDLER WHEN ALL
                    0728   ;       OTHER METHODS FAIL.  PERMITS DEBUGGERS TO RECEIVE CONTROL EVEN
                    0728   ;       WHEN THE STACK IS CLOBBERED.
                    0728   ;
                    0728           CTL     AL_FINALEXC                     ; FINAL EXCEPTION HANDLER ARRAY
          00000000' 0728           .LONG   EXE$EXCPTN                      ; KERNEL MODE
          00000000' 072C           .LONG   EXE$EXCPTNE                     ; EXECUTIVE MODE
          00000000  0730           .LONG   0                               ; SUPERVISOR MODE
          00000000  0734           .LONG   0                               ; USER MODE
                    0738   ;
                    0738   ;       POINTER TO BASE OF CHANNELS
                    0738   ;
                    0738           CTL     GL_CCBBASE                      ; BASE OF I/O CHANNELS
          00000000  0738           .LONG   0                               ; FILLED IN IN PROCSTRT
                    073C           CTL     GQ_DBGAREA                      ; DEBUG AREA
          00010000  073C           .LONG   64*1024                         ; SIZE
          7FFF0000  0740           .LONG   DBGAREA                         ; ADDRESS
                    0744   ;
                    0744   ; POINTER TO RMS IMAGE
                    0744   ;
                    0744           CTL     GL_RMSBASE
          00000000  0744           .LONG   0                               ; POINT TO RMS IN SYSTEM SPACE
                    0748   ;
                    0748   ;       PROCESS PERMANENT MESSAGE SECTION POINTER AND DEFAULT DISPLAY FLAGS
                    0748   ;
                    0748           CTL     GL_PPMSG                        ; ADDRESS OF PROCESS PERM. MSG SECTION
 00000000 00000000  0748           .QUAD   0                               ; STARTING/ENDING ADDRESS OF SECTION
                    0750
                    0750           CTL     GB_MSGMASK                      ; DEFAULT MESSAGE DISPLAY FLAGS
                0F  0750           .BYTE   15                              ; DEFAULT: FAC,SEV,IDENT AND TEXT (ALL)
                    0751
                    0751           CTL     GB_DEFLANG                      ; DEFAULT MESSAGE LANGUAGE
                00  0751           .BYTE   0                               ; (CURRENTLY UNUSED)
                    0752
                    0752           CTL     GW_PPMSGCHN                     ; CHANNEL TO PROCESS PERM. MESSAGE
              0000  0752           .WORD   0                               ; SECTION (MAPPED IN CTL$GL_PPMSG)
                    0754
                    0754           CTL     GL_USRUNDWN                     ; PER-PROCESS VECTOR TO USER RUNDOWN
          00000000  0754           .LONG   0                               ;  SERVICE
                    0758
                    0758           CTL     GL_PCB                          ; ADDRESS OF PROCESS CONTROL BLOCK
          00000000  0758           .LONG   0                               ; INIT BY PROCSTRT
                    075C
                    075C           CTL     GL_RUF                          ; POINTER TO RECOVERY UNIT BLOCKS
          00000000  075C           .LONG   0
                    0760
                    0760           CTL     GL_SITESPEC                     ; SITE-SPECIFIC PER-PROCESS CELL
          00000000  0760           .LONG   0
                    0764
                    0764           CTL     GL_KNOWNFIL                     ; PROCESS KNOWN FILE LIST POINTER
          00000000  0764           .LONG   0
                    0768
                    0768           CTL     AL_IPASTVEC                     ; VECTOR OF IPAST ADDRESSES
```

```
00000000 00000000 00000000 00000000  0768        .LONG   0,0,0,0,0,0,0,0
00000000 00000000 00000000 00000000  0778
                                     0788
                                     0788        CTL     GL_CMCNTX                  ; ADDRESS OF AME CONTEXT PAGE
                           7FFE1E00' 0788        .LONG   CTL$AL_CMCNTX
                                     078C
                                     078C        CTL     GL_IAFLNKPTR               ; ADDRESS OF IAF LIST (FOR DEBUGGER)
                           7FFE2E00' 078C        .LONG   CTL$GL_IAFLINK
                                     0790
                                     0790        CTL     GL_F11BXQP                 ; ADDRESS OF F11B XQP QUEUE AND
                            00000000 0790        .LONG   0                          ; DISPATCH VECTORS
                                     0794
                                     0794        CTL     GQ_P0ALLOC                 ; HEADER OF P0 EXTENTION TO PROCESS
                   00000000 00000000 0794        .LONG   0,0                        ; ALLOCATION REGION
                                     079C
                                     079C        CTL     GL_PRCALLCNT               ; COUNT OF BYTES OF PROCESS ALLOCATION
                            00000000 079C        .LONG   0                          ; REGION USABLE BY IMAGE REQUESTS.
                                     07A0
                                     07A0        CTL     GL_RDIPTR                  ; POINTER TO RIGHTS DATABASE IDENTIFIER (RDI
                            00000000 07A0        .LONG   0
                                     07A4
                                     07A4        CTL     GL_LNMDIRSEQ               ; SEQUENCE NUMBER FOR CACHE OF LOGICAL
                            00000000 07A4        .LONG   0                          ;  NAME TABLE TRANSLATIONS
                                     07A8
                                     07A8        CTL     GQ_HELPFLAGS               ; HELP FLAGS, ONE LONGWORD FOR USE
                   00000000 00000000 07A8        .LONG   0,0                        ; NOW, ONE FOR LATER
                                     07B0
                                     07B0        CTL     GQ_TERMCHAR                ; HOME FOR TERMINAL CHARACTERISTICS
                   00000000 00000000 07B0        .LONG   0 , 0                      ; NEED A QUADWORD
                                     07B8
                                     07B8        CTL             GL_KRPFL           ; P1 POOL LOOKASIDE LIST FORWARD LINK
                            7FFEFFB8 07B8        .LONG           CTL$GL_KRPFL
                                     07BC        CTL             GL_KRPBL           ; P1 POOL LOOKASIDE LIST BACKWARD LINK
                            7FFEFFB8 07BC        .LONG           CTL$GL_KRPFL
                                     07C0
                                     07C0        CTL             GL_CREPRC_FLAGS ; $CREPRC FLAGS
                            00000000 07C0        .LONG   0                          ;
                                     07C4
                                     07C4        CTL     GL_THCOUNT                 ; COUNT OF TERMINATION HANDLERS FOR
                            00000000 07C4        .LONG   0                          ;  EXEC MODE,
                            00000000 07C8        .LONG   0                          ;  SUPERVISOR MODE, AND
                            00000000 07CC        .LONG   0                          ;  USER MODE
                                     07D0
                                     07D0  ;***************************************************************************
                                     07D0  ;
                                     07D0  ;       END OF VECTOR PAGE
                                     07D0  ;
                                     07D0  ;***************************************************************************
                            000007D0 07D0  CTLVECEND=.
                            FFFFFFD0 07D0        .IF GREATER     <CTLVECEND-CTLVECPAG>-512
                                     07D0        .ERROR          ; *** VECTOR PAGE NOW LARGER THAN A PAGE ***
                                     07D0        .ENDC
                                     07D0
                                     07D0        .IFF                               ; ASSEMBLE REST ONLY FOR SHELL MODULE:
                                     07D0
```

```
                     07D0        .SBTTL  PROCESS I/O SEGMENT
                     07D0  ;---------------------------------------------------------------------
                     07D0  ;
                     07D0  ;     PROCESS I/O SEGMENT INITIAL CONTENT
                     07D0  ;
                     07D0  ;---------------------------------------------------------------------
                     07D0
                     07D0        .ALIGN  PAGE                        ; PAGE BOUNDARY
                     0800  PIO:                                      ; BASE OF PROCESS I/O SEGMENT
                     0800
                     0800        PIO     GL_FMLH                     ; FREE MEMORY LIST HEAD
7FFE0200 7FFE0200    0800        .LONG   PIO$GL_FMLH,PIO$GL_FMLH ;        (Empty List)
                     0808
                     0808        PIO     GL_IIOFSPLH                 ; FREE LIST HEADER FOR IMAGE I/O SEG
         7FFE0208    0808        .LONG   PIO$GL_IIOFSPLH             ;
         7FFE0208    080C        .LONG   PIO$GL_IIOFSPLH             ;
                     0810
                     0810        PIO     GW_STATUS                   ; RMS OVERALL STATUS
             0000    0810        .WORD   0                           ;
                     0812
                     0812
         00000010    0812  PIO$S_EODSTR==16                          ; SIZE OF STRING + COUNT BYTE
                     0812        PIO     GT_ENDSTR                   ; END OF DATA STRING
         00000822    0812        .BLKB   PIO$S_EODSTR                ; (COUNTED STRING)
                     0822
                     0822        PIO     GW_DFPROT                   ; DEFAULT FILE PROTECTION
             FA00    0822        .WORD   ^XFA00                      ; SYS:RWED, OWN:RWED, GROUP: RE, WORLD:N
                     0824
                     0824        PIO     GB_DFMBC                    ; DEFAULT MULTI-BLOCK COUNT
         00000825    0824        .BLKB   1
                     0825
                     0825        PIO     GB_DFMBFSDK                 ; DEFAULT MULTI-BUFFER COUNT SEQ. DISK
         00000826    0825        .BLKB   1
                     0826
                     0826        PIO     GB_DFMBFSMT                 ; DEFAULT MULTI-BUFFER COUNT MAGTAPE
         00000827    0826        .BLKB   1
                     0827
                     0827        PIO     GB_DFMBFSUR                 ; DEFAULT MULTI-BUFFER COUNT UNIT REC.
         00000828    0827        .BLKB   1
                     0828
                     0828        PIO     GB_DFMBFREL                 ; DEFAULT MULTI-BUFFER COUNT RELATIVE
         00000829    0828        .BLKB   1
                     0829
                     0829        PIO     GB_DFMBFIDX                 ; DEFAULT MULTI-BUFFER COUNT INDEXED
         0000082A    0829        .BLKB   1
                     082A
                     082A        PIO     GB_DFMBFHSH                 ; DEFAULT MULTI-BUFFER COUNT HASHED
         0000082B    082A        .BLKB   1
                     082B
                     082B        PIO     GB_DFNBC                    ; Network block count transfer size
         0000082C    082B        .BLKB   1
                     082C
                     082C        PIO     GB_RMSPROLOG                ; Structure level for RMS files
         0000082D    082C        .BLKB   1
                     082D
                     082D        PIO     GW_RMSEXTEND                ; Extend quantity for RMS files
         0000082F    082D        .BLKW   1
```

```
          082F
          082F      .ALIGN  LONG
          0830
          0830      PIO     GL_DIRCACHE           ; DIRECTORY CACHE LIST HEAD
7FFE0230  0830      .LONG   PIO$GL_DIRCACHE       ; EMPTY LIST
7FFE0230  0834      .LONG   PIO$GL_DIRCACHE
          0838
          0838      PIO     GL_DIRCFRLH           ; FREE LIST FOR DIRECTORY CACHE NODES
00000000  0838      .LONG   0                     ;    (NOTE: SINGLY LINKED)
          083C
          083C      PIO     GL_RULOCK             ; List of locks held for Recovery Units
00000000  083C      .LONG   0                     ;    (Note: Singly Linked)
          0840
          0840      PIO     GL_NXTIRBSEQ          ; Next sequence number for IRB$L_IDENT
00000000  0840      .LONG   0
          0844
```

```
                           0844
                           0844         .ALIGN  QUAD
                           0848
                           0848    ;
                           0848    ;    PROCESS I/O SEGMENT CONTEXT AREA
                           0848    ;
                           0848         PIO     GW_PIOIMPA              ;
                     0000  0848         .WORD   0                       ; FLAGS
                     000D  084A         .WORD   PRT$C_UREW              ; I/O BUFFER PROTECTION
       00000000 00000000  084C         .LONG   0,0                     ; PIO SEGMENT, SET UP BY PROCSTRT
                           0854
                           0854         PIOL    A1                      ; FREE PAGE LIST HEAD
       7FFE0254 7FFE0254  0854         .LONG   PIOLA1,PIOLA1           ; FREE LIST HEADER
                00000860  085C         .BLKL   1                       ; SP SAVED LONGWORD
               7FFE0C00'  0860         .LONG   PIO2A10                 ; IFAB TABLE ADDRESS
               7FFE0D00'  0864         .LONG   PIO2A20                 ; IRAB TABLE ADDRESS
                0000003F  0868         .LONG   IMP$C_NPIOFILES         ; # OF SLOTS PER TABLE
                           086C
```

```
                           086C  ;
                           086C  ;       IMAGE I/O SEGMENT CONTEXT AREA
                           086C  ;
                           086C          .ALIGN  QUAD
                           0870
                           0870          PIO     GW_IIOIMPA                   ; IMAGE I/O IMPURE AREA
                     0001  0870          .WORD   1                            ; FLAGS WITH IIO SEGMENT SET
                     000D  0872          .WORD   PRT$C_UREW                   ; PROTECTION STO SET ON PAGES
                           0874
                 00000000  0874          .LONG   0                            ; MUST BE INITIALIZED BY THE IMAGE
                           0878                                               ; ACTIVATOR TO THE ADDRESS OF THE FIRST
                           0878                                               ; PAGE IN THE IMAGE I/O SEGMENT.
                           0878                                               ; (NOTE: THIS LONGWORD IS REFERENCED BY:
                           0878                                               ; PIO$GW_IIOIMPA+IMP$L_IOSEGADDR)
                 00000000  0878          .LONG   0                            ; LENGTH OF IMAGE I/O SEGMENT IN BYTES
                           087C                                               ; MUST BE INITIALIZED BY THE IMAGE ACTIVATOR
                           087C                                               ; (NOTE: THIS LONGWORD IS REFERENCED BY:
                           087C                                               ; PIO$GW_IIOMPA+IMP$L_IOSEGLEN)
                           087C
                           087C          PIOL    B1                           ;
        7FFE027C 7FFE027C  087C          .LONG   PIOLB1,PIOLB1                ; FREEPAGE LIST HEAD
                           0884
                           0884
                           0884          PIOL    B2                           ;
                 00000888  0884          .BLKL   1                            ; SP SAVE LONGWORD
                           0888
                7FFE0294'  0888          .LONG   PIOLB10                      ; IFAB TABLE ADDRESS
                7FFE02D4'  088C          .LONG   PIOLB20                      ; IRAB TABLE ADDRESS
                 0000000F  0890          .LONG   IMP$C_ENTPERSEG              ; # OF SLOTS PER TABLE
                           0894
                           0894          PIOL    B10                          ;
                 00000000  0894          .LONG   0                            ; LINK TO NEXT TABLE SEGMENT
                 000008D4  0898          .BLKL   IMP$C_ENTPERSEG              ; IFAB TABLE SLOTS
                           08D4
                           08D4          PIOL    B20                          ;
                 00000000  08D4          .LONG   0                            ; LINK TO NEXT TABLE SEGMENT
                 00000914  08D8          .BLKL   IMP$C_ENTPERSEG              ; IRAB TABLE SLOTS
                           0914
                           0914          PIO     AL_RMSEXH                    ; EXIT HANDLER CONTROL BLOCK
                 0000091C  0914          .BLKL   2                            ;
                 00000001  091C          .LONG   1                            ; ARGUMENT COUNT
                 7FFE031C  0920          .LONG   PIO$AL_RMSEXH+8              ; STORE EXIT CODE OVER ARGUMENT COUNT
                           0924
                           0924          PIO     GQ_IIODEFAULT                ; DEFAULT IMAGE I/O AREA
        00000000 00000000  0924          .LONG   0,0
                           092C
                           092C          .ALIGN  LONG
                           092C
                           092C  ;
                           092C  ;       DEFAULT DIRECTORY INFORMATION
                           092C  ;
                           092C          PIO     GT_DDSTRING                  ; DEFAULT DIRECTORY STRING
                           092C
                 7FFE032C  092C  FIL$GT_DDSTRING==PIO$GT_DDSTRING             ; FILE READ DEFAULT DIRECTORY STRING
                           092C
                           092C  ; THE DEFAULT DIRECTORY STRING IN THE PROCESS QUOTA BLOCK MOVED FROM A
                           092C  ; CREATOR PROCESS TO A NEWLY CREATED PROCESS MUST BE AT LEAST AS LARGE
```

```
                              092C  ; AS THE LARGEST DIRECTORY STRING ALLOWED BY RMS.
                              092C
                              092C          ASSUME  PQB$S_DDSTRING GE FWA$C_MAXDIRLEN
                              092C
                    0000092C  092C  TEMP$...=.
5D 45 58 45 53 59 53 5B 00'  092C          .ASCIC  \[SYSEXE]\                ; DEFAULT DIRECTORY STRING
                          08  092C
                              0935
                              0935  ; ******************************** NOTE WELL ********************************
                              0935  ;
                              0935  ; THE DEFAULT DIRECTORY STRING MUST BE THE LAST ELEMENT TO APPEAR IN THE
                              0935  ; RMS CONTEXT AREA. IN FACT, IT HANGS OFF THE END OF THE PAGE. IF STORAGE
                              0935  ; WERE ALLOCATED, THE SHELL WOULD GROW IN SIZE BY ONE PAGE, AN UNNECESSARY
                              0935  ; WASTE OF SPACE. THE .BLKB DIRECTIVE HERE IS SYMBOLIC, TO EXPRESS THE SIZE
                              0935  ; OF THE DIRECTORY STRING IF SPACE WERE REALLY ALLOCATED.
                              0935  ;
                              0935  ;        .BLKB   PQB$S_DDSTRING-<.-TEMP$...>      ; FILL TO MAXIMUM POSSIBLE SIZE
                              0935  ;
                              0935  ; ******************************* END OF NOTE *******************************
```

```
                                0935          .SBTTL  INITIALZE SHELL WITH SYSGEN PARAMETERS
                                0935    ;++
                                0935    ; FUNCTIONAL DESCRIPTION:
                                0935    ;       SWP$SHELINIT IS CALLED BY THE SWAPPER FOLLOWING THE INSWAP OF A SHELL
                                0935    ;       PROCESS TO APPLY THE SYSGEN PARAMETERS AND CONFIGURE THE PROCESS
                                0935    ;       FOR THE PROPER WORKING SET SIZE AND VIRTUAL ADDRESS SPACE.
                                0935    ;       THIS CODE IS CONTAINED IN PAGES THAT ARE TO BECOME THE KERNEL
                                0935    ;       STACK FOR THE NEW PROCESS.
                                0935    ;
                                0935    ; CALLING SEQUENCE:
                                0935    ;       JSB     @#SWP$SHELINIT
                                0935    ;
                                0935    ; INPUT PARAMETERS:
                                0935    ;       R4 - PCB ADDRESS
                                0935    ;       R9 - SWAPPER MAP POINTER
                                0935    ;       R10 - PTC$M_VALID!PTE$C_ERKW!PTE$M_MODIFY
                                0935    ;       R11 - SWAPPER END MAP POINTER
                                0935    ;
                                0935    ; OUTPUT PARAMETERS:
                                0935    ;       PHD AND CONTROL REGION FOR THE NEW PROCESS
                                0935    ;
                                0935    ;--
                                0935
                                0935            .ALIGN  PAGE                    ; START OUT ON PAGE BOUNDARY
                                0A00
               00000A00         0A00    SWP$SHELINIT==.-SHELL                   ;
            55   6C A4   D0     0A00            MOVL    PCB$L_PHD(R4),R5        ; GET PROCESS HEADER BASE ADDRESS
     50   00000000'9F    D0     0A04            MOVL    @#SWP$GL_SHELLSIZ,R0    ; GET PAGES ALLOCATED FOR SHELL
            5B    6940   DE     0A0B            MOVAL   (R9)[R0],R11            ; COMPUTE END ADDRESS IN MAP
50   00000000'9F    02   78     0A0F            ASHL    #2,@#SWP$GL_SHELIO,R0   ; GET I/O SIZE OF SHELL
        51    5B    59   C3     0A17            SUBL3   R9,R11,R1               ; COMPUTE SIZE OF EXTENSION PAGES
              51    50   C2     0A1B            SUBL    R0,R1                   ; LESS I/O SIZE
        51    51    03   9C     0A1E            ROTL    #3,R1,R1                ; CONVERT TO DOUBLE QUAD COUNT
50   00000000'9F    09   9C     0A22            ROTL    #9,@#SWP$GL_SHELIO,R0   ; COMPUTE ADDRESS AT END OF I/O TRANSFER
                    80   7C     0A2A    10$:    CLRQ    (R0)+                   ; CLEAR PAGES
                    80   7C     0A2C            CLRQ    (R0)+                   ; NOT READ FROM SHELL
              F9    51   F5     0A2E            SOBGTR  R1,10$
              52    55   D0     0A31            MOVL    R5,R2                   ; VA OF PHD
          00000000'9F    16     0A34            JSB     @#MMG$SVAPTECHK         ; GET SVA OF FIRST PHD PTE
              83    89   D0     0A3A            MOVL    (R9)+,(R3)+             ; MAP PROCESS HEADER FIXED PAGE
     51   00000000'9F    3C     0A3D            MOVZWL  @#SWP$GW_WSLPTE,R1      ; GET COUNT OF ADDED PAGES FOR WSL+PST
                    08   13     0A44            BEQL    30$                     ; BR IF NONE
                    51   D7     0A46            DECL    R1                      ; LESS FIXED HEADER PAGE
              83    7B   D0     0A48    20$:    MOVL    -(R11),(R3)+            ; MAP A WSL PAGE
              FA    51   F5     0A4B            SOBGTR  R1,20$                  ; DO THEM ALL
     50   00000000'9F    3C     0A4E    30$:    MOVZWL  @#SWP$GW_EMPTPTE,R0     ; GET COUNT OF EMPTY PAGES
            53    6340   DE     0A55            MOVAL   (R3)[R0],R3             ; UPDATE MAP POINTER
     50   00000000'9F    3C     0A59            MOVZWL  @#SWP$GW_BAKPTE,R0      ; GET COUNT OF BAK/WSL/VAL/LCK PTE
              83    7B   D0     0A60    40$:    MOVL    -(R11),(R3)+            ; MAP BAK/WSL/VAL/LCK PAGES
              FA    50   F5     0A63            SOBGTR  R0,40$
                                0A66            INVALID                         ; INVALIDATE TRANSLATION BUFFER
                                0A69    ;
                                0A69    ;       ALL OF THE HEADER PAGES LESS PAGE TABLES HAVE NOW BEEN MAPPED
                                0A69    ;
     58   00000000'9F    3C     0A69            MOVZWL  @#SWP$GW_IBALSETX,R8    ; GET BALANCE SLOT INDEX
         42 A5    58     B0     0A70            MOVW    R8,PHD$W_PHVINDEX(R5)   ; SET INTO PROCESS HEADER
  34 A5   00000000'9F    33     0A74            CVTWB   @#SGN$GW_DFPFC,PHD$B_DFPFC(R5)  ; SET DEFAULT PAGE FAULT CLUSTER
```

```
                 50  D4 0A7C          CLRL    R0                       ; COUNT MAXIMUM FREE SPACE TILL NOW
   53  00000000'9F  3C 0A7E          MOVZWL  @#MMG$GW_MINPFIDX,R3     ; START SCAN AT FIRST PAGE FILE
              51 53  D0 0A85          MOVL    R3,R1                    ; SET DEFAULT PAGING FILE TO CHOOSE
   56  00000000'9F  D0 0A88          MOVL    @#MMG$GL_PAGSWPVC,R6     ; GET ADDRESS OF PAGE FILES VECTOR
                        0A8F     ;
                        0A8F     ;       CHECK FOR SPECIFICATION OF A PAGE FILE
                        0A8F     ;       THIS IS TAKEN AS ADVICE ONLY, NORMAL ALGORITHM USED IF SELECTION IMPOSSIBLE
                        0A8F     ;
        52    5A A4  3C 0A8F          MOVZWL  PCB$B_PGFLINDEX(R4),R2   ; GET THE REQUESTED PAGE FILE INDEX
                 1C  13 0A93          BEQL    44$                      ; NONE SPECIFIED
              51 52  D1 0A95          CMPL    R2,R1                    ; CHECK FOR LEGAL INDEX
                 17  1F 0A98          BLSSU   44$                      ; A SWAP FILE WAS SPECIFIED - GIVE UP
    00000000'9F 52  D1 0A9A          CMPL    R2,@#MMG$GL_MAXPFIDX
                 0E  1A 0AA1          BGTRU   44$                      ; TOO HIGH - GIVE UP
         58   6642  D0 0AA3          MOVL    (R6)[R2],R8              ; GET PFL BLOCK ADDRESS
      05 23 A8   00  E1 0AA7          BBC     #PFL$V_INITED,PFL$B_FLAGS(R8),44$ ; BRANCH IF NOT USABLE
              51 52  D0 0AAC          MOVL    R2,R1
                 29  11 0AAF          BRB     47$                      ; WE HAVE A GOOD SELECTION
         52   6643  D0 0AB1  44$:    MOVL    (R6)[R3],R2              ; GET ADDRESS OF NEXT PAGE FILE BLOCK
          58 A4  01  88 0AB5          BISB    #PFL$M_INITED,PCB$W_PGFLCHAR(R4); ADD INITED TO THE REQUESTED FLAGS
   58   23 A2 58 A4  89 0AB9          BISB3   PCB$W_PGFLCHAR(R4),PFL$B_FLAGS(R2),R8 ; OR THE FLAGS TOGETHER
          23 A2  58  91 0ABF          CMPB    R8,PF[$B_FLAGS(R2)       ; DID THEY CHANGE
                 0D  12 0AC3          BNEQ    46$                      ; YES - NO MATCH - TRY AGAIN
        50    18 A2  D1 0AC5          CMPL    PFL$L_FREPAGCNT(R2),R0   ; CHOOSE PAGING FILE WITH MOST FREE PAGES
                 07  15 0AC9          BLEQ    46$                      ; BRANCH IF WE HAVE BETTER CANDIDATE
        50    18 A2  D0 0ACB          MOVL    PFL$L_FREPAGCNT(R2),R0   ; SAVE NEW FREE PAGE COUNT
              51 53  D0 0ACF          MOVL    R3,R1                    ; SAVE NEW INDEX
 D7 53  00000000'9F  F3 0AD2  46$:    AOBLEQ  @#MMG$GL_MAXPFIDX,R3,44$; LOOP THROUGH ALL PAGE FILES
          1F A5  51  90 0ADA  47$:    MOVB    R1,PHD$B_PAGFIL(R5)      ; SET PAGING FILE INDEX
 35 A5  00000000'9F  90 0ADE          MOVB    @#SGN$GB_PGTBPFC,PHD$B_PGTBPFC(R5); SET SYSTEM DEFAULT PT CLUSTER
   08 A5    0063 8F  B0 0AE6          MOVW    #SWP$C_KSTACK_WSL,PHD$Q_WSLIST(R5); INIT POINTER TO LIST BASE
   57   00000000'9F  D0 0AEC          MOVL    @#SGN$GL_PHDPAGCT,R7     ; GET TOTAL COUNT OF HEADER PAGES
   56   00000000'9F  3C 0AF3          MOVZWL  @#SWP$GW_WSLPTE,R6       ; GET COUNT OF WSL HEADER PAGES
   20 A5    56   09  78 0AFA          ASHL    #9,R6,PHD$L_PSTBASOFF(R5)       ; SET END AS BASE FOR PST
   56   00000000'9F  A0 0AFF          ADDW    @#SWP$GW_EMPTPTE,R6      ; ASSUMES NO OVERFLOW POSSIBLE
        56    56 07  78 0B06          ASHL    #7,R6,R6                 ; CONVERT TO LONGWORD COUNT
          48 A5  56  D0 0B0A          MOVL    R6,PHD$L_WSLX(R5)        ; SET BASE OFFSET TO WSLX AREA
   57   00000000'9F  C0 0B0E          ADDL    @#SGN$GL_PTPAGCNT,R7     ; ADD PAGE TABLES TO COUNT
        50    57 01  C1 0B15          ADDL3   #1,R7,R0                 ; ROUND TO LONGWORD SIZE
              50 02  C6 0B19          DIVL    #2,R0                    ;
              56 50  C0 0B1C          ADDL    R0,R6                    ; ALLOCATE SPACE IN WHOLE LONGWORDS
          44 A5  56  D0 0B1F          MOVL    R6,PHD$L_BAK(R5)         ; SET BASE OF BACKING STORE VECTOR
              56 57  C0 0B23          ADDL    R7,R6                    ; ALLOCATE SPACE FOR BACKING STORE VECTOR
              56 04  C4 0B26          MULL    #4,R6                    ; CONVERT TO BYTE OFFSET
          64 A5  56  D0 0B29          MOVL    R6,PHD$L_PTWSLELCK(R5)   ; SET BASE OF LOCKED COUNT VECTOR
   57   00000000'9F  D0 0B2D          MOVL    @#SGN$GL_PTPAGCNT,R7     ; GET COUNT OF PAGE TABLES
     68 A5    6647  9E 0B34          MOVAB   (R6)[R7],PHD$L_PTWSLEVAL(R5)    ; ALLOCATE AND SET BASE OF VAL CNT
   50     68 B545   9E 0B39          MOVAB   @PHD$L_PTWSLEVAL(R5)[R5],R0     ; SET BASE FOR VALID COUNT VEC
              52 01  CE 0B3E          MNEGL   #1,R2                           ; MINUS ONE FOR BACKGROUND
   51     64 B545   9E 0B41          MOVAB   @PHD$L_PTWSLELCK(R5)[R5],R1     ; AND BASE FOR LOCKED COUNT
              80 52  90 0B46  50$:    MOVB    R2,(R0)+                 ; INIT BOTH COUNT VECTORS TO MINUS ONE
              81 52  90 0B49          MOVB    R2,(R1)+
              F7 57  F5 0B4C          SOBGTR  R7,50$                   ; FOR ALL PAGE TABLE SLOTS
          FE A1  04  80 0B4F          ADDB    #KSTACK+1,-2(R1)         ; COUNT OF LOCK PAGES FOR SHELL
          FE A0  06  80 0B53          ADDB    #KSTACK+3,-2(R0)         ; COUNT OF VALID PAGES
   56   00000000'9F  D0 0B57          MOVL    @#SGN$GL_PHDAPCNT,R6     ; GET ACTUAL HEADER SIZE
   50   00000000'9F  9A 0B5E          MOVZBL  @#SWP$GB_SHLP1PT,R0      ; NUMBER OF PERM PAGE TABLES
```

```
       6C A5   50   B0  0B65       MOVW    R0,PHD$W_PTCNTLCK(R5)     ; COUNT OF PT CONTAINING LOCKED PAGES
       6E A5   50   B0  0B69       MOVW    R0,PHD$W_PTCNTVAL(R5)     ; COUNT OF PT CONTAINING VALID PAGES
       70 A5   50   B0  0B6D       MOVW    R0,PHD$W_PTCNTACT(R5)     ; COUNT OF ACTIVE PAGE TABLES
       72 A5   50   B0  0B71       MOVW    R0,PHD$W_PTCNTMAX(R5)     ; COUNT OF PT WITH NON-ZERO ENTRIES
          56   50   C0  0B75       ADDL    R0,R6                     ; MAKE ROOM IN WSLIST FOR HDR+PGTBLS
    57    0E A5     3C  0B78       MOVZWL  PHD$W_WSDYN(R5),R7        ; GET INDEX TO CURRENT DYNAMIC ENTRIES
               57   DD  0B7C       PUSHL   R7                        ; SAVE POINTER FOR LATER USE
       0C A5   56   A0  0B7E       ADDW    R6,PHD$W_WSLOCK(R5)       ; CORRECT LOCK WSL INDEX
       0E A5   56   A0  0B82       ADDW    R6,PHD$W_WSDYN(R5)        ; AND DYNAMIC INDEX
       10 A5   56   A0  0B86       ADDW    R6,PHD$W_WSNEXT(R5)       ; AND NEXT POINTER
    51    57   56   C1  0B8A       ADDL3   R6,R7,R1                  ; SET DESTINATION INDEX
          50   01   9A  0B8E       MOVZBL  #SWP$C_NDYN,R0            ; SET COUNT OF DYNAMIC PAGES
     6541    6547   D0  0B91  60$: MOVL    (R5)[R7],(R5)[R1]         ; MOVE A DYNAMIC WORKING SET PAGE
               51   D6  0B96       INCL    R1                        ; NEXT DESTINATION
               57   D6  0B98       INCL    R7                        ; NEXT WSL ENTRY
          F4   50   F5  0B9A       SOBGTR  R0,60$                    ; MOVE ALL DYNAMIC PAGES
    58   80000029 8F   D0  0B9D    MOVL    #<WSL$M_VALID-            ; WSL ENTRIES FOR HEADER ARE VALID
                        0BA4               !WSL$M_WSLOCK-            ; LOCKED,
                        0BA4               !WSL$C_PPGTBL-            ; PROCESS PAGE TABLE,
                        0BA4               !VA$M_SYSTEM>,R8          ; AND SYSTEM SPACE
          52   55   D0  0BA4       MOVL    R5,R2                     ; SET PHD ADDRESS TO GET SVAPTE
   00000000'9F      16  0BA7       JSB     @#MMG$SVAPTECHK           ; SVAPTE TO R3
          50   58   D0  0BAD       MOVL    R8,R0                     ; MAKE WORKING COPY
 51   00000000'9F   3C  0BB0       MOVZWL  @#SWP$GW_WSLPTE,R1        ; COUNT OF WSLPTE
          52   8E   D0  0BB7       MOVL    (SP)+,R2                  ; RESTORE WORKING SET INDEX TO LOCKED
             0147   30  0BBA       BSBW    FILLPHD                   ; SET UP FOR VALID PHD PAGES
 51   00000000'9F   3C  0BBD       MOVZWL  @#SWP$GW_EMPTPTE,R1       ; GET COUNT OF EMPTY PTE AGAIN
       53    6341   DE  0BC4       MOVAL   (R3)[R1],R3               ; ADVANCE SPT ADDRESS
    51    51   09   78  0BC8       ASHL    #9,R1,R1                  ; CONVERT TO BYTE COUNT
          50   51   C0  0BCC       ADDL    R1,R0                     ; AND UPDATE CURRENT VA
 51   00000000'9F   3C  0BCF       MOVZWL  @#SWP$GW_BAKPTE,R1        ; GET COUNT OF REMAINING HEADER PAGES
             012B   30  0BD6       BSBW    FILLPHD                   ; SET UP FOR VALID PHD PAGES
 51   00000000'9F   D0  0BD9       MOVL    @#SGN$GL_PTPAGCNT,R1      ; GET COUNT OF PAGE TABLES
FC A341  30000000 8F   D0  0BE0  65$: MOVL #PTE$C_ERKW,-4(R3)[R1]    ; SET A DEMAND ZERO PAGE FOR PAGE TABLES
          F4   51   F5  0BE9       SOBGTR  R1,65$                    ; INIT ENTIRE PAGE TABLE AREA
 51   00000000'9F   9A  0BEC       MOVZBL  @#SWP$GB_SHLP1PT,R1       ; GET COUNT OF P1 PAGE TABLES
          56   51   CE  0BF3       MNEGL   R1,R6                     ; NEGATE COUNT
               56   D7  0BF6       DECL    R6                        ; LESS ONE MORE FOR DEBUGGER PAGE
 56   00000000'9F   C0  0BF8       ADDL    @#SGN$GL_PTPAGCNT,R6      ; PAGE OFFSET TO PAGE TABLE
       53    6346   DE  0BFF       MOVAL   (R3)[R6],R3               ; COMPUTE SVAPTE
    56    56   09   78  0C03       ASHL    #9,R6,R6                  ; CONVERT TO BYTE OFFSET
          50   56   C0  0C07       ADDL    R6,R0                     ; ADD TO CURRENT BASE WSL
               0A   BB  0C0A       PUSHR   #^M<R1,R3>                ; SAVE KEY REGS
          51   02   C2  0C0C       SUBL    S^#SWP$C_SHLP1PT,R1       ; LESS STANDARD P1 PAGE TABLES
               06   15  0C0F       BLEQ    80$                       ; BR IF NO EXTENSIONS
          83   7B   D0  0C11  70$: MOVL    -(R11),(R3)+              ; STORE A PTE FOR EXTENSION
          FA   51   F5  0C14       SOBGTR  R1,70$                    ; DO ALL EXTENSION PAGE TABLES
          51   02   D0  0C17  80$: MOVL    S^#SWP$C_SHLP1PT,R1       ; SET COUNT OF STANDARD
          83   89   D0  0C1A  90$: MOVL    (R9)+,(R3)+               ; STORE PTE FOR STANDARD
          FA   51   F5  0C1D       SOBGTR  R1,90$                    ; DO ALL STANDARD P1PT
               0A   BA  0C20       POPR    #^M<R1,R3>                ; RESTORE KEY REGISTERS
             00DF   30  0C22       BSBW    FILLPHD                   ; SET UP FOR VALID PHD PAGES
 50   00000000'9F   09   9C  0C25  ROTL    #9,@#SGN$GL_PHDPAGCT,R0   ; OFFSET FOR P0BR
       00C8 C5   50   C0  0C2D     ADDL    R0,PHD$L_P0BR(R5)         ; FORM P0BR BASE VALUE
 51   00000000'9F   09   9C  0C32  ROTL    #9,@#SGN$GL_PTPAGCNT,R1   ; OFFSET TO P1BR
          51   50   C0  0C3A       ADDL    R0,R1                     ;
       00D0 C5   51   C0  0C3D     ADDL    R1,PHD$L_P1BR(R5)         ; FORM P1BR BASE VALUE
```

```
          30 A5     50   C2  0C42           SUBL    R0,PHD$L_FREP1VA(R5)      ; COMPUTE CORRECT FREP1VA
             51     50   C2  0C46           SUBL    R0,R1                     ; EXTRACT PT SIZE TOTAL
    50    50    F7  8F   78  0C49           ASHL    #-9,R0,R0                 ; BACK TO PAGE COUNT
        00D4 C5     50   C2  0C4E           SUBL    R0,PHD$L_P1LR(R5)         ; NOW CORRECT P1 LENGTH REGISTER
    51    51    FE  8F   78  0C53           ASHL    #-2,R1,R1                 ; COUNT OF AVAILABLE PTES
             51     50   C2  0C58           SUBL    R0,R1                     ; DIMINISH BY PHD WINDOW SIZE
          2C A5     51   C0  0C5B           ADDL    R1,PHD$L_FREPTECNT(R5)    ; AND FORM COUNT OF FREE PTES
                             0C5F
                             0C5F   ; THE FOLLOWING SHUFFLE OF THE WORKING SET LIST REARRANGES ALL WORKING SET
                             0C5F   ; LIST ENTRIES THAT ARE NOT KERNEL STACK PAGES. THE SWAPPER MAP IS BEING
                             0C5F   ; REORDERED TO AGREE WITH THE WORKING SET LIST DEFINED IN THE SHELL.
                             0C5F
                             0C5F           .REPEAT SHUFFLE_COUNT             ; COUNT FROM WSL TEMPLATE
                             0C5F           MOVL    (R9)+,(R11)+              ; REARRANGE SWAP PAGE LIST
             8B     89   D0  0C5F           .ENDR
                             0C65
                    8B   D4  0C65           CLRL    (R11)+                    ; SET STOPPER IN MAP
  30 A5  00000200 8F     C1  0C67           ADDL3   #512,PHD$L_FREP1VA(R5),-       ; SET ADDRESS OF WINDOW
         00000688 9F         0C6F                   @#<SWP$AL_PTRPAG+-        ; INTO POINTER PAGE
                             0C74                   <CTL$GL_PHD-CTL$GL_VECTORS>>     ; THROUGH SWAPPER MAP
     00 24 A4       12   E2  0C74           BBSS    #PCB$V_PHDRES,PCB$L_STS(R4),100$; MARK PHD RESIDENT
                             0C79   100$:                                     ;
       50    08 A5  3C       0C79           MOVZWL  PHD$W_WSLIST(R5),R0       ; GET INDEX TO WS BASE
                    50   D7  0C7D           DECL    R0                        ;
 51    00000000'9F  50   A1  0C7F           ADDW3   R0,@#SGN$GL_MAXWSCNT,R1                   ; CALC MAX AUTHORIZED
             0A A5  51   B0  0C87           MOVW    R1,PHD$W_WSAUTH(R5)                       ; SET MAX AUTHORIZED
             18 A5  51   B0  0C8B           MOVW    R1,PHD$W_WSQUOTA(R5)                      ; AND QUOTA
             16 A5  51   B0  0C8F           MOVW    R1,PHD$W_WSEXTENT(R5)                     ; AND EXTENT
             14 A5  51   B0  0C93           MOVW    R1,PHD$W_WSAUTHEXT(R5)                    ; AND AUTHORIZED EXTENT
  74 A5  00000000'9F     B0  0C97           MOVW    @#SGN$GW_MINWSCNT,PHD$W_WSFLUID(R5)       ; SET FLUID REQUIREMENT
  51     00000000'9F     B0  0C9F           MOVW    @#SGN$GW_DFWSCNT,R1                       ; GET DEFUALT WS SIZE
                50  51   A0  0CA6           ADDW    R1,R0                                     ; CALC LAST
             12 A5  50   B0  0CA9           MOVW    R0,PHD$W_WSLAST(R5)                       ; SET LAST
             1A A5  50   B0  0CAD           MOVW    R0,PHD$W_DFWSCNT(R5)                      ; AND DEFAULT COUNT
             50 A5  51   B0  0CB1           MOVW    R1,PHD$W_WSSIZE(R5)                       ; SET WS SIZE
           00C8 C5  55   C0  0CB5           ADDL    R5,PHD$L_P0BR(R5)         ; BIAS P0 BASE REGISTER
           00D0 C5  55   C0  0CBA           ADDL    R5,PHD$L_P1BR(R5)         ; AND P1 BASE REGISTER ALSO
                             0CBF   ;
                             0CBF   ;       REMAP SYSTEM SERVICE VECTORS FOR THIS PROCESS
                             0CBF   ;
    52   7FFEDE00 8F     D0  0CBF           MOVL    #P1SYSVECTORS,R2          ; PUT THEM HERE
         00000000'9F     16  0CC6           JSB     @#MMG$SVAPTECHK           ; GET ADDRESS OF PTE FOR VECTORS
    50   00000000'9F     D0  0CCC           MOVL    @#MMG$GL_SPTBASE,R0       ; SYSTEM PAGE TABLE
             51     05   9A  0CD3           MOVZBL  #SGN$C_SYSVECPGS,R1       ; NUMBER OF PAGES
 83   80 00200000 8F     C9  0CD6   110$:   BISL3   #PTE$M_WINDOW,(R0)+,(R3)+
                 F5 51   F5  0CDE           SOBGTR  R1,110$
                             0CE1   ;
                             0CE1   ;       THE PAGE TABLE ARRAY FOR LOCKED WSLE'S MUST BE INCREMENTED ONE
                             0CE1   ;       FOR EACH WINDOW PTE.  THIS BALANCES THE LOGIC IN $CREPAG/$DELPAG
                             0CE1   ;       THAT PLACES THE PAGE TABLE PAGE IN THE LOCKED PORTION OF THE
                             0CE1   ;       WORKING SET FOR USER PFNMAPPED PAGES.
                             0CE1   ;
                             0CE1   ;       THIS LOGIC ASSUMES THAT THE VECTOR PAGES ALL LIVE IN THE SAME
                             0CE1   ;       PAGE TABLE PAGE.
                             0CE1   ;
    50    53   00C8 C5   C3  0CE1           SUBL3   PHD$L_P0BR(R5),R3,R0      ;BYTE OFFSET OF PTE
       50    50    F7 8F 78  0CE7           ASHL    #-9,R0,R0                 ;BYTE INDEX OF CONTAINING PAGE TABLE
```

```
        50   64 A5   C0  0CEC        ADDL    PHD$L_PTWSLELCK(R5),R0 ;COMPUTE THE ADDRESS OF THE LOCKED
           50   55   C0  0CF0        ADDL    R5,R0                  ; WORKING SET LIST PAGE TABLE BYTE
           60   05   80  0CF3        ADDB    #SGN$C_SYSVECPGS,(R0)  ; ARRAY AND INCR BY NUMBER OF VEC PAGES
                         0CF6
                         0CF6 ;
                         0CF6 ;      FINISH UP INITIALIZATION, SET DATA NEEDED BY SWAPPER
                         0CF6 ;
     20 A4   20 A4   CE  0CF6        MNEGL   PCB$L_WSSWP(R4),PCB$L_WSSWP(R4)
                         0CFB                                   ; INDICATE SHELL FINISHED TO SWAPPER
     52 A5   5C A4   F7  0CFB        CVTLW   PCB$L_SWAPSIZE(R4),PHD$W_SWAPSIZE(R5) ; SET INITIAL SWAP ALLOCATION
                         0D00
                         0D00 ; CLEAR THE PCB FIELDS THAT WERE USED TO CARRY PAGE FILE AND SWAP FILE
                         0D00 ; INFORMATION TO THE NEW PROCESS. THESE FIELDS WILL BECOME THE COMMON EVENT
                         0D00 ; FLAG CLUSTER POINTERS.
                         0D00
                         0D00        ASSUME PCB$L_EFC3P EQ <PCB$L_EFC2P + 4>
                         0D00        ASSUME PCB$W_PGFLCHAR EQ PCB$L_EFC2P
                         0D00        ASSUME PCB$B_PGFLINDEX EQ <PCB$L_EFC2P + 2>
                         0D00        ASSUME PCB$L_SWAPSIZE EQ <PCB$L_EFC2P + 4>
                         0D00
             58 A4   7C  0D00        CLRQ    PCB$L_EFC2P(R4)        ; START WITH NO COMMON EF CLUSTERS
                         0D03
                     05  0D03        RSB                            ; RETURN
                         0D04
```

```
                         0D04        .SBTTL  FILLPHD - SETUP A VALID PHD PTE
                         0D04  ;
                         0D04  ;       R0 - WORKING SET LIST ENTRY, ADDRESS AND FLAGS (UPDATED)
                         0D04  ;       R1 - COUNT OF PAGES TO FILL (UPDATED)
                         0D04  ;       R2 - WORKING SET LIST INDEX (UPDATED)
                         0D04  ;       R3 - SVAPTE FOR PHD PAGE (UPDATED)
                         0D04  ;       R4 - PCB ADDRESS
                         0D04  ;       R5 - PHD ADDRESS
                         0D04  ;       R7 - SCRATCH
                         0D04  ;       R8 - SCRATCH
                         0D04  ;       R10 - PFN$C_ERKW!PFN$M_MODIFY!PFN$M_VALID
                         0D04  ;
                         0D04  FILLPHD:                                    ; SETUP VALID PHD PTES
      6542    50    D0   0D04  10$:    MOVL    R0,(R5)[R2]                 ; STORE WORKING SET LIST ENTRY
   57   83    5A    CB   0D08          BICL3   R10,(R3)+,R7                ; GET PFN FROM SPT
   00000000'9F      DD   0D0C          PUSHL   @#PFN$AB_STATE              ; STACK ADDRESS BASE FOR PFN STATE
   9E47    87 8F    90   0D12          MOVB    #<PFN$C_ACTIVE!PFN$M_MODIFY>,@(SP)+[R7] ; MARK PAGE ACTIVE
   00000000'9F      DD   0D17          PUSHL   @#PFN$AB_TYPE               ; STACK ADDRESS BASE FOR PFN TYPE
      9E47    04    90   0D1D          MOVB    #PFN$C_PPGTBL,@(SP)+[R7]; SET PAGE TYPE TO PAGE TABLE
   00000000'9F      DD   0D21          PUSHL   @#PFN$AL_BAK                ; STACK ADDRESS BASE FOR PFN BAK
      58    1F A5   9A   0D27          MOVZBL  PHD$B_PAGFIL(R5),R8         ; GET PAGE FILE NUMBER
 9E47   58    18    9C   0D2B          ROTL    #PFN$V_PGFLX,R8,@(SP)+[R7]        ; SET BACKING STORE ADDRESS
   00000000'9F      DD   0D30          PUSHL   @#PFN$AL_PTE                ; STACK ADDRESS OF PFN PTE BACK POINTER
 9E47      FC A3    DE   0D36          MOVAL   -4(R3),@(SP)+[R7]           ; SET PTE BACK POINTER FOR PAGE
                         0D3B  ;
                         0D3B  ; The following use of the PFN_REFERENCE macro must force absolute addressing
                         0D3B  ; because of the peculiar method in which this code executes. In addition,
                         0D3B  ; this code is not a part of the nonpaged executive and cannot have its opcode
                         0D3B  ; automatically fixed up by INIT.
                         0D3B  ;
   00000000'9F      DD   0D3B          PUSHL   @#PFN$Ax_WSLX               ; STACK BASE ADDRESS OF WSLX VECTOR
                         0D41                  PFN_REFERENCE   -
                         0D41          MOVW    <R2,@(SP)+[R7]>,-           ; SET WSLX FOR PAGE
                         0D41                  LONG_OPCODE=MOVZWL,-
                         0D41                  IMAGE=SHELL,-
                         0D41                  MODE=@#                     ; FORCE POSITION INDEPENDENCE
            52      D6   0D53          INCL    R2                          ; NEXT WORKING SET LIST ENTRY
   50     0200 C0   DE   0D55          MOVAL   512(R0),R0                  ; NEXT VA
         A7 51      F5   0D5A          SOBGTR  R1,10$                      ; FILL ALL REQUESTED PAGES
            05           0D5D          RSB                                 ; AND RETURN
                         0D5E
                         0D5E          .ALIGN  PAGE                        ; PAGE ALIGN
          00000007       0E00  SWP$C_SHELLSIZ=<.-SHELL>@-9                 ; SIZE OF SHELL PROCESS IN PAGES
          FFFFFFFF       0E00          .IF GT  SWP$C_SHELLSIZ-8
                         0E00          .ERROR  SWP$C_SHELLSIZ; Shell size changed, update swapper and FREELIM
                         0E00          .ENDC
                         0E00
```

```
          0E00
          0E00   ;
          0E00   ;          Process IFI/ISI Tables
          0E00   ;
          0E00
          0E00   ; The rest of this module merely defines global symbols and offsets into
          0E00   ; various P1 pages. No more storage is added to the SHELL module.
          0E00
          0E00              .PSECT  $ABS$,ABS
          0000
7FFE0C00  0000              .=PIOTBL
          0C00   TBL:
          0C00              PIO2    A10                       ; LINK TO NEXT TABLE SEGMENT
7FFE0C04  0C00              .BLKL   1                         ; INITIALLY ZERO
7FFE0D00  0C04              .BLKL   IMP$C_NPIOFILES           ; IFAB TABLE SLOTS
          0D00
          0D00              PIO2    A20                       ; LINK TO NEXT TABLE SEGMENT
7FFE0D04  0D00              .BLKL   1                         ; INITIALLY ZERO
7FFE0E00  0D04              .BLKL   IMP$C_NPIOFILES           ; IRAB TABLE SLOTS
          0E00
```

```
          0E00     .SBTTL  COMMAND LANGUAGE INTERPRETER DATA AREA
          0E00 ;-----------------------------------------------------------------------------
          0E00 ;                  GENERIC COMMAND LANGUAGE INTERPRETER DATA
          0E00 ;-----------------------------------------------------------------------------
          0E00
7FFE3000  0E00     .=CLIDATAPAG                              ; SET BASE VALUE FOR REGION
          3000 CTL$AL_CLICALBK::                             ; CALL BACK VECTOR FOR CLI
7FFE3004  3000     .BLKL   1
7FFE3008  3004     .BLKL   1                                 ; ALLOW FOR EXPANSION
          3008 CTL$AG_CLIMAGE::
7FFE3010  3008     .BLKL   2                                 ; VA RANGE INTO WHICH CLI IS MAPPED
          3010 CTL$AG_CLITABLE::                             ; VA RANGE INTO WHICH CLI TABLE IS
7FFE3018  3010     .BLKL   2                                 ; MAPPED.
          3018
          3018 CTL$GL_UAF_FLAGS::
7FFE301C  3018     .BLKL   1                                 ; FLAGS FROM AUTHORIZATION RECORD
          301C
          301C CTL$GT_CLINAME::                              ; CLI NAME (FILE NAME ONLY)
7FFE303C  301C     .BLKB   PQB$S_CLI_NAME                    ; STORED AS COUNTED ASCII
          303C CTL$GT_TABLENAME::                            ; CLI TABLE NAME (FULL FILE SPEC)
7FFE313C  303C     .BLKB   PQB$S_CLI_TABLE                   ; STORED AS COUNTED ASCII
          313C CTL$GT_SPAWNCLI::                             ; SPAWN CLI NAME (FILE NAME ONLY)
7FFE315C  313C     .BLKB   PQB$S_SPAWN_CLI                   ; STORED AS COUNTED ASCII
          315C CTL$GT_SPAWNTABLE::                           ; SPAWN CLI TABLE NAME (FULL FILE SPEC)
7FFE325C  315C     .BLKB   PQB$S_SPAWN_TABLE                 ; STORED AS COUNTED ASCII
          325C
          325C ASSUME <CTL$GT_TABLENAME -CTL$GT_CLINAME> EQ <PQB$T_CLI_TABLE  -PQB$T_CLI_NAME>
          325C ASSUME <CTL$GT_SPAWNCLI  -CTL$GT_CLINAME> EQ <PQB$T_SPAWN_CLI  -PQB$T_CLI_NAME>
          325C ASSUME <CTL$GT_SPAWNTABLE-CTL$GT_CLINAME> EQ <PQB$T_SPAWN_TABLE-PQB$T_CLI_NAME>
          325C
          325C CTL$AG_CLIDATA::                              ; START OF DATA AREA
000015A4  325C CTL$C_CLIDATASZ==CLIDATAEND-CTL$AG_CLIDATA        ; SIZE OF DATA AREA
          325C
```

```
                  325C  1405          .SBTTL  COMPATIBILITY MODE EMULATOR CONTEXT PAGE
                  325C  1406 ;-------------------------------------------------------------------------------
                  325C  1407 ;        COMPATIBILITY MODE EMULATOR CONTEXT PAGE
                  325C  1408 ;-------------------------------------------------------------------------------
        7FFE1E00  325C  1409          .=CTL$AG_CMEDATA                   ; POINT TO START OF AREA
                  1E00  1410 CTL$AL_CMCNTX::                             ; COMPATIBILITY MODE REGISTER CONTEXT
        7FFE1E28  1E00  1411          .BLKL   10                         ; SAVED BY EXCEPTION
                  1E28  1412
```

```
                1E28        .SBTTL  GLOBAL USER-MODE CONTEXT PAGE
                1E28  ;-----------------------------------------------------------------------
                1E28  ;     GLOBAL USER-MODE-ONLY CONTEXT PAGE
                1E28  ;
                1E28  ;     This page may ONLY be used by user-mode code, since the data
                1E28  ;     has absolutely no protection from errant user-mode programs.
                1E28  ;     Do NOT use this area for cells which cannot be wiped out by
                1E28  ;     any user-mode program at any time during image execution,
                1E28  ;     except when it only prevents that program from running.
                1E28  ;-----------------------------------------------------------------------
      7FFE2200  1E28        .=UWVECPAG                          ; POINT TO START OF AREA
                2200
                2200  CTL$GL_DCLPRSOWN::                        ; ADDRESS OF DCL WRK AREA CREATED
      7FFE2204  2200        .BLKL   1                           ; BY DCL OR CLI$DCL_PARSE
                2204  CTL$GL_CLINTOWN::                         ; ADDRESS OF CONTEXT BLOCK USED BY
      7FFE2208  2204        .BLKL   1                           ; CLI$INTERFACE PACKAGE IN DCL/MCR
                2208
                2208  ; THE REST OF THIS PAGE IS UNUSED
                2208
```

```
                  2208         .SBTTL  IMAGE ACTIVATOR CONTEXT PAGE
                  2208  ;---------------------------------------------------------------------------
                  2208  ;      IMAGE ACTIVATOR CONTEXT PAGE
                  2208  ;
                  2208  ;      The following page contains image activator context that must
                  2208  ;      remain behind after an image is activated. The IAFLIST survives
                  2208  ;      across successive activations. The IAFEXE and IAFMERGE lists are
                  2208  ;      cleared when a new image activation is initiated. The FIXUP link
                  2208  ;      is cleared as soon as a given fixup pass completes.
                  2208  ;
                  2208  ;---------------------------------------------------------------------------
      7FFE2E00    2208         .=IMGACTCTX                          ; Point to start of area
                  2E00
                  2E00  CTL$GL_IAFLINK::                            ; Listhead of linked list of fixup vectors
      7FFE2E04    2E00         .BLKL   1                            ;  (containing shareable image lists)
                  2E04
                  2E04  CTL$GL_IAFLAST::                            ; Address of last fixup vector in list
      7FFE2E08    2E04         .BLKL   1                            ;  (used in normal and merged image activati
                  2E08
                  2E08  CTL$GL_FIXUPLNK::                           ; Listhead of linked list of fixup
      7FFE2E0C    2E08         .BLKL   1                            ;  vectors used during fixup pass
                  2E0C
                  2E0C  CTL$GL_P1MERGE::                            ; Listhead of linked list of fixup vectors
      7FFE2E10    2E0C         .BLKL   1                            ;  for images merged into P1 space
                  2E10
                  2E10  ; The following data structure is an empty fixup vector that facilitates
                  2E10  ; the fixup vector list manipulation. It is plugged into the fixup vector
                  2E10  ; lists by PROCSTRT when a process is first created.
                  2E10
                  2E10  CTL$GL_IAFPERM::
      7FFE2E50    2E10         .BLKB   IAF$K_LENGTH
                  2E50
                  2E50  ; The following label locates the link field in the just allocated structure.
                  2E50  ; While an image is active, it locates the fixup vector for an executable image.
                  2E50  ; It is cleared as part of the image activation initialization code path.
                  2E50
      7FFE2E10    2E50  CTL$GL_IAFEXE == CTL$GL_IAFPERM + IAF$L_IAFLINK
                  2E50
                  2E50  IAC$GL_IMAGCTX::
      7FFE2E54    2E50         .BLKL   1                            ; Context that exists for life of image
                  2E54
                  2E54  IAC$GL_PROCCTX::
      7FFE2E58    2E54         .BLKL   1                            ; Context that exists beyond image exit
                  2E58
                  2E58  IAC$AL_VECADDR::
      7FFE2E68    2E58         .BLKL   4                            ; Array of altered opcode addresses
                  2E68
                  2E68  IAC$AL_VECOPCOD::
      7FFE2E6C    2E68         .BLKB   4                            ; Array of saved opcodes
                  2E6C
                  2E6C  IAC$AW_VECRESET::
      7FFE2E74    2E6C         .BLKW   4                            ; Array of offsets used to reset vectors
                  2E74
                  2E74  IAC$AW_VECSET::
      7FFE2E7C    2E74         .BLKW   4                            ; Array of offsets used to locate vectors
                  2E7C
                  2E7C  ; The following linked list contains image control blocks for all of the
```

```
          2E7C  ; images currently mapped into a process' address space.
          2E7C
          2E7C  IAC$GL_IMAGE_LIST::
7FFE2E84  2E7C          .BLKL   2                       ; Room for forward and backward links
          2E84
          2E84  ; The following linked list is used by the image activator to record work
          2E84  ; in progress. It is empty while an image is executing.
          2E84
          2E84  IAC$GL_WORK_LIST::
7FFE2E8C  2E84          .BLKL   2                       ; Room for forward and backward links
          2E8C
          2E8C  ; The following list is a potential source of unused image control blocks.
          2E8C  ; Although empty when a process is created, it grows to reflect the largest
          2E8C  ; number of images activated at the same time.
          2E8C
          2E8C  IAC$GL_ICBFL::
7FFE2E94  2E8C          .BLKL   2                       ; Room for forward and backward links
          2E94
          2E94  ; The following two cells locate the ICB for the main image and the ICB for
          2E94  ; the image most recently merged into the address space.
          2E94
          2E94  IAC$GL_MAIN_ICB::
7FFE2E98  2E94          .BLKL   1                       ; ICB of main image
          2E98
          2E98  IAC$GL_FIRST_ICB::
7FFE2E9C  2E98          .BLKL   1                       ; ICB of image just merged
          2E9C
          2E9C  IAC$GL_STACK_SIZE::
7FFE2EA0  2E9C          .BLKL   1                       ; Amount by which to expand user stack
```

```
2EA0
2EA0          .IFT                                  ; IF USRVECTOR MODULE:
2EA0
2EA0  ;
2EA0  ; These symbols (SYS$...) specify which cells in the P1 pointer page may
2EA0  ; be used by any program, regardless of system version, without having
2EA0  ; to link with SYS.STB (and thus, make it system dependant).
2EA0  ;
2EA0  ; These symbols will remain constant forever and will continue to be supported
2EA0  ; from release to release.
2EA0  ;
2EA0
2EA0          .MACRO  USRSYM  SYM,VALCHECK,CTLSYM
2EA0          .IF     B,CTLSYM
2EA0  SYS$'SYM == CTL$'SYM
2EA0          .IF     NB,VALCHECK
2EA0          .IF     NE,CTL$'SYM'-^X'VALCHECK'
2EA0          .ERROR  ; *** CTL$'SYM' MUST EQ ^X'VALCHECK' TO REMAIN COMPATIBLE WITH PREVI
2EA0          .ENDC
2EA0          .ENDC
2EA0          .IFF
2EA0  SYS$'SYM == CTL$'CTLSYM
2EA0          .IF     NB,VALCHECK
2EA0          .IF     NE,CTL$'CTLSYM'-^X'VALCHECK'
2EA0          .ERROR  ; *** CTL$'CTLSYM' MUST EQ ^X'VALCHECK' TO REMAIN COMPATIBLE WITH PR
2EA0          .ENDC
2EA0          .ENDC
2EA0          .ENDC
2EA0          .ENDM
2EA0
2EA0          .DISABLE TRACEBACK
2EA0
2EA0          USRSYM  GL_CMCNTX,7FFEFF88        ; Address of 2 pages of AME storage
2EA0          USRSYM  GL_IAFLINK,7FFEFF8C,-     ; Points to IMGACT fixup listhead
2EA0                          GL_IAFLNKPTR
2EA0          USRSYM  GL_IMGLSTPTR,7FFEFE84     ; Points to ICM list (for debugger)
2EA0
2EA0          .ENDC                             ; END OF SHELL / USRVECTOR CONDITIONAL
2EA0
2EA0          .END
```

```
CLIDATAEND                      = 7FFE4800
CLIDATAPAG                      = 7FFE3000
COMMON                          = 00000004
CTL$AG_CLIDATA                    7FFE325C   G
CTL$AG_CLIMAGE                    7FFE3008   G
CTL$AG_CLITABLE                   7FFE3010   G
CTL$AG_CMEDATA                  = 7FFE1E00   G
CTL$AL_CLICALBK                   7FFE3000   G
CTL$AL_CMCNTX                     7FFE1E00   G
CTL$AL_FINALEXC                 = 7FFEFF28   G
CTL$AL_IPASTVEC                 = 7FFEFF68   G
CTL$AL_STACK                    = 7FFEFE10   G
CTL$AL_STACKLIM                 = 7FFEFE6C   G
CTL$AQ_EXCVEC                   = 7FFEFE34   G
CTL$A_COMMON                    = 7FFE1600   G
CTL$A_DISPVEC                   = 7FFE6000   G
CTL$C_CLIDATASZ                 = 000015A4   G
CTL$C_KRP_COUNT                 = 00000004   G
CTL$C_KRP_SIZE                  = 00000200   G
CTL$GB_DEFLANG                  = 7FFEFF51   G
CTL$GB_MSGMASK                  = 7FFEFF50   G
CTL$GB_PWRMODE                  = 7FFEFF24   G
CTL$GB_SSFILTER                 = 7FFEFF25   G
CTL$GL_CCBBASE                  = 7FFEFF38   G
CTL$GL_CLINTOWN                   7FFE2204   G
CTL$GL_CMCNTX                   = 7FFEFF88   G
CTL$GL_CMHANDLR                 = 7FFEFE30   G
CTL$GL_CMSUPR                   = 7FFEFE28   G
CTL$GL_CMUSER                   = 7FFEFE2C   G
CTL$GL_CREPRC_FLAGS             = 7FFEFFC0   G
CTL$GL_CTLBASVA                 = 7FFEFE7C   G
CTL$GL_DCLPRSOWN                  7FFE2200   G
CTL$GL_F11BXQP                  = 7FFEFF90   G
CTL$GL_FINALSTS                 = 7FFEFEB8   G
CTL$GL_FIXUPLNK                   7FFE2E08   G
CTL$GL_GETMSG                   = 7FFEFE68   G
CTL$GL_IAFEXE                   = 7FFE2E10   G
CTL$GL_IAFLAST                    7FFE2E04   G
CTL$GL_IAFLINK                    7FFE2E00   G
CTL$GL_IAFLNKPTR                = 7FFEFF8C   G
CTL$GL_IAFPERM                    7FFE2E10   G
CTL$GL_IBIOCNT                  = 7FFEFEE8   G
CTL$GL_ICPUTIM                  = 7FFEFED0   G
CTL$GL_IDIOCNT                  = 7FFEFEE4   G
CTL$GL_IFAULTIO                 = 7FFEFED8   G
CTL$GL_IFAULTS                  = 7FFEFED4   G
CTL$GL_IMGHDRBF                 = 7FFEFE80   G
CTL$GL_IMGLSTPTR                = 7FFEFE84   G
CTL$GL_IPAGEFL                  = 7FFEFEE0   G
CTL$GL_IVOLUMES                 = 7FFEFEEC   G
CTL$GL_IWSPEAK                  = 7FFEFEDC   G
CTL$GL_KNOWNFIL                 = 7FFEFF64   G
CTL$GL_KRP                      = 7FFE6600   G
CTL$GL_KRPBL                    = 7FFEFFBC   G
CTL$GL_KRPFL                    = 7FFEFFB8   G
CTL$GL_KSPINI                   = 7FFE7E00   G
CTL$GL_KSTKBAS                  = 7FFE7800   G
CTL$GL_KSTKBASEXP               = 7FFE7000   G
CTL$GL_LNMDIRECT                = 7FFEFE08   G
CTL$GL_LNMDIRSEQ                = 7FFEFFA4   G
CTL$GL_LNMHASH                  = 7FFEFE04   G
CTL$GL_P1MERGE                    7FFE2E0C   G
CTL$GL_PCB                      = 7FFEFF58   G
CTL$GL_PHD                      = 7FFEFE88   G
CTL$GL_POWERAST                 = 7FFEFF20   G
CTL$GL_PPMSG                    = 7FFEFF48   G
CTL$GL_PRCALLCNT                = 7FFEFF9C   G
CTL$GL_RDIPTR                   = 7FFEFFA0   G
CTL$GL_RMSBASE                  = 7FFEFF44   G
CTL$GL_RUF                      = 7FFEFF5C   G
CTL$GL_SITESPEC                 = 7FFEFF60   G
CTL$GL_THCOUNT                  = 7FFEFFC4   G
CTL$GL_THEXEC                   = 7FFEFE54   G
CTL$GL_THSUPR                   = 7FFEFE58   G
CTL$GL_UAF_FLAGS                  7FFE3018   G
CTL$GL_USRCHME                  = 7FFEFF1C   G
CTL$GL_USRCHMK                  = 7FFEFF18   G
CTL$GL_USRUNDWN                 = 7FFEFF54   G
CTL$GL_VECTORS                  = 7FFEFE00   G
CTL$GL_VIRTPEAK                 = 7FFEFEC0   G
CTL$GL_VOLUMES                  = 7FFEFEC4   G
CTL$GL_WSPEAK                   = 7FFEFEBC   G
CTL$GQ_ALLOCREG                 = 7FFEFE8C   G
CTL$GQ_COMMON                   = 7FFEFE60   G
CTL$GQ_DBGAREA                  = 7FFEFF3C   G
CTL$GQ_HELPFLAGS                = 7FFEFFA8   G
CTL$GQ_ISTART                   = 7FFEFEC8   G
CTL$GQ_LNMTBLCACHE              = 7FFEFE20   G
CTL$GQ_LOGIN                    = 7FFEFEB0   G
CTL$GQ_MOUNTLST                 = 7FFEFE94   G
CTL$GQ_POALLOC                  = 7FFEFF94   G
CTL$GQ_PROCPRIV                 = 7FFEFF10   G
CTL$GQ_TERMCHAR                 = 7FFEFFB0   G
CTL$GT_CLINAME                    7FFE301C   G
CTL$GT_SPAWNCLI                   7FFE313C   G
CTL$GT_SPAWNTABLE                 7FFE315C   G
CTL$GT_TABLENAME                  7FFE303C   G
CTL$GW_CHINDX                   = 7FFEFE02   G
CTL$GW_NMIOCH                   = 7FFEFE00   G
CTL$GW_PPMSGCHN                 = 7FFEFF52   G
CTL$T_ACCOUNT                   = 7FFEFEA8   G
CTL$T_NODEADDR                  = 7FFEFEF0   G
CTL$T_NODENAME                  = 7FFEFEF7   G
CTL$T_REMOTEID                  = 7FFEFEFE   G
CTL$T_USERNAME                  = 7FFEFE9C   G
CTLVECEND                       = 000007D0 R        03
CTLVECPAG                       = 00000600 R        03
DBGAREA                         = 7FFF0000
DBGPTCNT                        = 00000001
DYNWSL                          = 0000019C R        03
END...                          = 00000404 R        03
ESPINI                          = 7FFE9E00
ESTACK                          = 00000010
EXE$EXCPTN                        ********     X    03
```

```
EXE$EXCPTNE                ********   X   03    PFILPGCNT                   = 00000068
EXE$PROCSTRT               ********   X   03    PFL$B_FLAGS                 = 00000023
FIL$GT_DDSTRING          = 7FFE032C  G          PFL$L_FREPAGCNT             = 00000018
FILLPHD                    00000D04 R     03    PFL$M_INITED                = 00000001
FWA$C_MAXDIRLEN          = 000000FF             PFL$V_INITED                = 00000000
FWSL...                  = 000001A0 R     03    PFN$AB_STATE                  ********   X   03
IAC$AL_IMGACTBUF         = 7FFE4800  G          PFN$AB_TYPE                   ********   X   03
IAC$AL_VECADDR             7FFE2E58  G          PFN$AL_BAK                    ********   X   03
IAC$AL_VECOPCOD            7FFE2E68  G          PFN$AL_PTE                    ********   X   03
IAC$AW_VECRESET            7FFE2E6C  G          PFN$AX_WSLX                   ********   X   03
IAC$AW_VECSET              7FFE2E74  G          PFN$C_ACTIVE                = 00000007
IAC$GL_FIRST_ICB           7FFE2E98  G          PFN$C_PPGTBL                = 00000004
IAC$GL_ICBFL               7FFE2E8C  G          PFN$M_MODIFY                = 00000080
IAC$GL_IMAGCTX             7FFE2E50  G          PFN$V_PGFLX                 = 00000018
IAC$GL_IMAGE_LIST          7FFE2E7C  G          PHD$B_CPUMODE               = 00000060
IAC$GL_MAIN_ICB            7FFE2E94  G          PHD$B_DFPFC                 = 00000034
IAC$GL_PROCCTX             7FFE2E54  G          PHD$B_PAGFIL                = 0000001F
IAC$GL_STACK_SIZE          7FFE2E9C  G          PHD$B_PGTBPFC               = 00000035
IAC$GL_WORK_LIST           7FFE2E84  G          PHD$C_LENGTH                = 0000017C
IAF$K_LENGTH             = 00000040             PHD$L_BAK                   = 00000044
IAF$L_IAFLINK            = 00000000             PHD$L_FREP1VA               = 00000030
IMGACTBUF                = 00000008             PHD$L_FREPTECNT             = 0000002C
IMGACTCTX                = 7FFE2E00             PHD$L_KSP                   = 00000078
IMP$C_ENTPERSEG          = 0000000F             PHD$L_P0BR                  = 000000C8
IMP$C_NPIOFILES          = 0000003F             PHD$L_P0LRASTL              = 000000CC
IPL$_ASTDEL              = 00000002             PHD$L_P1BR                  = 000000D0
JIB$S_ACCOUNT            = 00000008             PHD$L_P1LR                  = 000000D4
JIB$S_USERNAME           = 0000000C             PHD$L_PC                    = 000000C0
JIB$T_ACCOUNT            = 00000018             PHD$L_PSL                   = 000000C4
JIB$T_USERNAME           = 0000000C             PHD$L_PSTBASOFF             = 00000020
KRP_COUNT                = 00000004             PHD$L_PTWSLELCK             = 00000064
KSPINI                   = 7FFE7E00             PHD$L_PTWSLEVAL             = 00000068
KSTACK                   = 00000003             PHD$L_WSLX                  = 00000048
KSTACK_EX                = 00000004             PHD$Q_PRIVMSK               = 00000000
MMG$GL_MAXPFIDX            ********   X   03    PHD$V_ASTLVL                = 00000018
MMG$GL_PAGSWPVC            ********   X   03    PHD$W_DFWSCNT               = 0000001A
MMG$GL_SPTBASE             ********   X   03    PHD$W_PHVINDEX              = 00000042
MMG$GW_BIGPFN              ********   X   03    PHD$W_PRCLM                 = 0000003E
MMG$GW_MINPFIDX            ********   X   03    PHD$W_PTCNTACT              = 00000070
MMG$IMGHDRBUF            = 7FFE6400  G          PHD$W_PTCNTLCK              = 0000006C
MMG$SVAPTECHK              ********   X   03    PHD$W_PTCNTMAX              = 00000072
NDYN                     = 00000001             PHD$W_PTCNTVAL              = 0000006E
NOT_KSTACK_WSL           = 00000198 R     03    PHD$W_QUANT                 = 0000003C
NSA$K_IDT_PAGES          = 00000003             PHD$W_SWAPSIZE              = 00000052
NSA$T_IDT                = 7FFE2800  G          PHD$W_WSAUTH                = 0000000A
P1PTBAS                  = 00000200 R     03    PHD$W_WSAUTHEXT             = 00000014
P1PTLEN                  = 000000FF             PHD$W_WSDYN                 = 0000000E
P1SYSVECTORS             = 7FFEDE00  G          PHD$W_WSEXTENT              = 00000016
PCB$B_PGFLINDEX          = 0000005A             PHD$W_WSFLUID               = 00000074
PCB$L_EFC2P              = 00000058             PHD$W_WSLAST                = 00000012
PCB$L_EFC3P              = 0000005C             PHD$W_WSLIST                = 00000008
PCB$L_PHD                = 0000006C             PHD$W_WSLOCK                = 0000000C
PCB$L_STS                = 00000024             PHD$W_WSNEXT                = 00000010
PCB$L_SWAPSIZE           = 0000005C             PHD$W_WSQUOTA               = 00000018
PCB$L_WSSWP              = 00000020             PHD$W_WSSIZE                = 00000050
PCB$V_PHDRES             = 00000012             PHD...                      = 00000000 R     03
PCB$W_PGFLCHAR           = 00000058             PIO                           00000800 R     03
```

```
PIO$AL_RMSEXH             = 7FFE0314   G          PTE$C_UOWN                     = 01800000
PIO$A_DIRCACHE            = 7FFE0800   G          PTE$C_UREW                     = 68000000
PIO$A_TRACE               = 7FFE0600   G          PTE$C_URKW                     = 70000000
PIO$GB_DFMBC              = 7FFE0224   G          PTE$C_URSW                     = 60000000
PIO$GB_DFMBFHSH           = 7FFE022A   G          PTE$C_UW                       = 20000000
PIO$GB_DFMBFIDX           = 7FFE0229   G          PTE$M_TYP1                     = 04000000
PIO$GB_DFMBFREL           = 7FFE0228   G          PTE$M_WINDOW                   = 00200000
PIO$GB_DFMBFSDK           = 7FFE0225   G          SAV...                         = 00000600 R      03
PIO$GB_DFMBFSMT           = 7FFE0226   G          SGN$C_SYSVECPGS                = 00000005
PIO$GB_DFMBFSUR           = 7FFE0227   G          SGN$GB_PGTBPFC                   ********   X    03
PIO$GB_DFNBC              = 7FFE022B   G          SGN$GL_MAXWSCNT                  ********   X    03
PIO$GB_RMSPROLOG          = 7FFE022C   G          SGN$GL_PHDAPCNT                  ********   X    03
PIO$GL_DIRCACHE           = 7FFE0230   G          SGN$GL_PHDPAGCT                  ********   X    03
PIO$GL_DIRCFRLH           = 7FFE0238   G          SGN$GL_PTPAGCNT                  ********   X    03
PIO$GL_FMLH               = 7FFE0200   G          SGN$GW_DFPFC                     ********   X    03
PIO$GL_IIOFSPLH           = 7FFE0208   G          SGN$GW_DFWSCNT                   ********   X    03
PIO$GL_NXTIRBSEQ          = 7FFE0240   G          SGN$GW_MINWSCNT                  ********   X    03
PIO$GL_RULOCK             = 7FFE023C   G          SHELL                            00000000 R      03
PIO$GQ_IIODEFAULT         = 7FFE0324   G          SHUFFLE_COUNT                  = 00000002
PIO$GT_DDSTRING           = 7FFE032C   G          SSPINI                         = 7FFEDE00
PIO$GT_ENDSTR             = 7FFE0212   G          SSTACK                         = 00000020
PIO$GW_DFPROT             = 7FFE0222   G          SWP$AL_PTRPAG                  = 00000600  G
PIO$GW_IIOIMPA            = 7FFE0270   G          SWP$C_DBGPTCNT                 = 00000001  G
PIO$GW_PIOIMPA            = 7FFE0248   G          SWP$C_KSTACK                   = 00000003  G
PIO$GW_RMSEXTEND          = 7FFE022D   G          SWP$C_KSTACK_EX                = 00000004  G
PIO$GW_STATUS             = 7FFE0210   G          SWP$C_KSTACK_EX_WSL            = 0000005F  G
PIO$S_EODSTR              = 00000010   G          SWP$C_KSTACK_WSL               = 00000063  G
PIO2AT0                   = 7FFE0C00              SWP$C_NDYN                     = 00000001  G
PIO2A20                   = 7FFE0D00              SWP$C_SHELLPFIL                = 00000068  G
PIOBASE                   = 7FFE0200              SWP$C_SHELLSIZ                 = 00000007
PIOLA1                    = 7FFE0254              SWP$C_SHLFPTE                  = 00000081  G
PIOLB1                    = 7FFE027C              SWP$C_SHLP1PT                  = 00000002  G
PIOLB10                   = 7FFE0294              SWP$GB_SHLP1PT                   ********   X    03
PIOLB2                    = 7FFE0284              SWP$GL_SHELIO                    00000000 RG     02
PIOLB20                   = 7FFE02D4              SWP$GL_SHELLBAS                  00000000 RG     03
PIOTBL                    = 7FFE0C00              SWP$GL_SHELLSIZ                  ********   X    03
PQB$S_CLI_NAME            = 00000020              SWP$GW_BAKPTE                    ********   X    03
PQB$S_CLI_TABLE           = 00000100              SWP$GW_EMPTPTE                   ********   X    03
PQB$S_DDSTRING            = 00000100              SWP$GW_IBALSETX                  ********   X    03
PQB$S_SPAWN_CLI           = 00000020              SWP$GW_WSLPTE                    ********   X    03
PQB$S_SPAWN_TABLE         = 00000100              SWP$SHELINIT                   = 00000A00  G
PQB$T_CLI_NAME            = 00000088              TBL                              7FFE0C00
PQB$T_CLI_TABLE           = 000000A8              TEMP$...                       = 0000092C R      03
PQB$T_SPAWN_CLI           = 000001A8              TMP                            = 00000400
PQB$T_SPAWN_TABLE         = 000001C8              TMP...                         = 04000000
PR$_TBIA                    ********     X    03  USPINI                         = 7FFE0200
PRT$C_UREW                = 0000000D              UWVECPAG                       = 7FFE2200
PSL$C_USER                = 00000003              VA$M_SYSTEM                    = 80000000
PSL$V_IPL                 = 00000010              VA...                          = 7FFE0200
PTE$C_EOWN                = 00800000              VECTORS                        = 7FFEFE00
PTE$C_ERKW                = 30000000              WSL$C_PPGTBL                   = 00000008
PTE$C_KOWN                = 00000000              WSL$C_PROCESS                  = 00000000
PTE$C_KW                  = 10000000              WSL$M_MODIFY                   = 00000100
PTE$C_NA                  = 00000000              WSL$M_VALID                    = 00000001
PTE$C_SOWN                = 01000000              WSL$M_WSLOCK                   = 00000020
PTE$C_SREW                = 48000000              WSL...                         = 0000018C R      03
PTE$C_SRKW                = 50000000
```

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

PSECT name                     Allocation          PSECT No.  Attributes
----------                     ----------          ---------  ----------
.  ABS  .                      00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                          7FFE325C  (*****.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
AEXENONPAGED                   00000004  (     4.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
YYSHELLPAGED                   00000E00  (  3584.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC PAGE

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   33    00:00:00.09    00:00:00.66
Command processing              159    00:00:00.65    00:00:05.84
Pass 1                          463    00:00:20.28    00:01:04.80
Symbol table sort                 0    00:00:02.12    00:00:05.46
Pass 2                          288    00:00:05.18    00:00:17.56
Symbol table output              39    00:00:00.26    00:00:00.68
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            987    00:00:28.61    00:01:35.14

The working set limit was 1950 pages.
115790 bytes (227 pages) of virtual memory were used to buffer the intermediate code.
There were 80 pages of symbol table space allocated to hold 1413 non-local and 18 local symbols.
1560 source lines were read in Pass 1, producing 31 object records in Pass 2.
37 pages of virtual memory were used to define 35 macros.

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

Macro library name                         Macros defined
------------------                         --------------
_$255$DUA28:[SHRLIB]RMS.MLB;1                           1
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                         17
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       7
TOTALS (all libraries)                                 25

1360 GETS were required to define 25 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SHELL/OBJ=OBJ$:SHELL MSRC$:SHELL/UPDATE=(ENH$:SHELL)+EXECML$/LIB+SHRLIB$:RMS.MLB/LIB
```

0380 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SHELL
LIS

RSE
LIS

SCBVECTOR
LIS

SDAT
LIS

SHMGSDRTN
LIS

SCSVEC
LIS

SCHED
LIS

SECAUDIT
LIS

SPTSKEL
LIS

RUFSYSVEC
LIS