


```

SSSSSSSS HH HH EEEEEEEEE LL LL
SSSSSSSS HH HH EEEEEEEEE LL LL
SS HH HH EE LL LL
SS HH HH EE LL LL
SS HH HH EE LL LL
SSSSSS HH HH EEEEEEE LL LL
SSSSSS HH HH EEEEEEE LL LL
SS HH HH EE LL LL
SS HH HH EE LL LL
SS HH HH EE LL LL
SSSSSSSS HH HH EEEEEEEEE LLLLLLLLLL LLLLLLLLLL
SSSSSSSS HH HH EEEEEEEEE LLLLLLLLLL LLLLLLLLLL

```

```

LL LL IIIIII SSSSSSSS
LL LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```

(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

```

0000 1      .IF      NDF,LIBSWITCH
0000 2      .TITLE   SHELL PROCESS DEFINITION
0000 3      .IFF
0000 4      .TITLE   SYSSUSRVECTOR - USER ACCESSIBLE P1 CELLS
0000 5      .ENDC
0000 6      .IDENT  'V04-000'
0000 7
0000 8      *****
0000 9      *
0000 10     *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 11     *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 12     *  ALL RIGHTS RESERVED.
0000 13     *
0000 14     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 15     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 16     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 17     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 18     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 19     *  TRANSFERRED.
0000 20     *
0000 21     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 22     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 23     *  CORPORATION.
0000 24     *
0000 25     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 26     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 27     *
0000 28     *
0000 29     *****
0000 30
0000 31     ++
0000 32     FACILITY:    EXECUTIVE, CREATE PROCESS/SWAPPER DATA BASE
0000 33
0000 34     ABSTRACT:    SHELL DEFINES THE SWAP IMAGE FOR THE INITIAL PROCESS
0000 35     STARTED BY THE CREATE PROCESS SYSTEM SERVICE.
0000 36
0000 37     ENVIRONMENT:
0000 38
0000 39     AUTHOR: RICHARD I. HUSTVEDT, CREATION DATE: 30-NOV-76
0000 40
0000 41     MODIFIED BY:
0000 42
0000 43     V03-039 RAS0319      Ron Schaefer      29-Jun-1984
0000 44     Add queue header cells for logical name table name
0000 45     translation cache.
0000 46
0000 47     V03-038 LJK0287      Lawrence J. Kenah    25-Jun-1984
0000 48     Add three longword array to contain the counts of termination
0000 49     handlers declared in each of the three outer access modes.
0000 50     Add sequence number cell for process private logical name
0000 51     directory.
0000 52
0000 53     V03-037 MSH0054      Michael S. Harvey    30-May-1984
0000 54     Remove some obsolete symbols.
0000 55
0000 56     V03-036 LJK0282      Lawrence J. Kenah    9-May-1984
0000 57     Add demand zero page after PIO page to accommodate long

```

```

0000 58 : directory names that straddle page boundary.
0000 59 :
0000 60 : V03-035 TMK0001 Todd M. Katz 27-Apr-1984
0000 61 : Remove CTL$GL_LOGTBL which was required only under the old
0000 62 : logical name design.
0000 63 :
0000 64 : V03-034 LJK0272 Lawrence J. Kenah 10-Apr-1984
0000 65 : Add image activator context cells privileged vectors
0000 66 : and shareable image initialization.
0000 67 :
0000 68 : V03-033 MHB0133 Mark Bramhall 9-Apr-1984
0000 69 : Remove preset values for CTL$T_USERNAME and CTL$T_ACCOUNT.
0000 70 : Validate CLI information ordering.
0000 71 :
0000 72 : V03-032 RAS0281 Ron Schaefer 28-Mar-1984
0000 73 : Add PIO$GB_DFNBC process-specific RMS parameter for
0000 74 : network block count transfer size.
0000 75 : Re-use the cell once used from RMS Compatibility.
0000 76 : Add CTL$GT_SPAWNCLI and CTL$GT_SPAWNTABLE as well.
0000 77 :
0000 78 : V03-031 WMC0006 Wayne Cardoza 21-Mar-1984
0000 79 : Go back to 3 pages of kernel stack.
0000 80 : Add cells for default image I/O area.
0000 81 :
0000 82 : V03-030 MSH0006 Michael S. Harvey 3-Feb-1984
0000 83 : Protect image header buffer page against USER WRITE.
0000 84 :
0000 85 : Protect image activator scratch page from user by changing
0000 86 : its owner to EXEC.
0000 87 :
0000 88 : Replace obsolete image rundown control flags with an image
0000 89 : list pointer for use by the debugger. This fixed cell pointer
0000 90 : allows the debugger to avoid linking against the executive.
0000 91 :
0000 92 : V03-029 RSH0091 R. Scott Hanna 31-Jan-1984
0000 93 : Add dedicated P1 demand zero pages for the security auditing
0000 94 : Impure Data Table (IDT). Remove the old vector page pointer
0000 95 : to the IDT.
0000 96 :
0000 97 : V03-028 LJK0258 Lawrence J. Kenah 18-Jan-1984
0000 98 : Correct problems introduced by LJK0257. In SHELINIT, only
0000 99 : reorder a number of pages equal to the pages in this module
0000 100 : that are NOT kernel stack pages. Convert a PIO page from
0000 101 : a page file page to demand zero.
0000 102 :
0000 103 : V03-027 BLS0262 Benn Schreiber 16-Jan-1984
0000 104 : Correct .ADDRESS to be .LONG in P1 lookaside list links
0000 105 :
0000 106 : V03-026 LJK0257 Lawrence J. Kenah 4-Jan-1984
0000 107 : A variety of changes.
0000 108 :
0000 109 : Add CTL$GQ_TERMCHAR and CREPRC_FLAGS to P1 vector page.
0000 110 :
0000 111 : Add UAF_FLAGS to CLI data page.
0000 112 :
0000 113 : Change PQB$C_MAXDIRLEN references to PQB$S_DDSTRING.
0000 114 :

```

```
0000 115 : Add second page for compatibility mode context.
0000 116 :
0000 117 : Reorder SHELL pages to achieve demand zero compaction.
0000 118 :
0000 119 : Add listhead and demand zero pages for P1 pool lookaside
0000 120 : list.
0000 121 :
0000 122 : Move page file index and characteristics from PQB to PCB.
0000 123 :
0000 124 : V03-025 WMC0005 Wayne Cardoza 02-Dec-1983
0000 125 : PHDSW_BAK, PHDSW_WSLX are now longwords.
0000 126 :
0000 127 : V03-024 WMC0004 Wayne Cardoza 28-Nov-1983
0000 128 : Move the kstack expansion WSL slots.
0000 129 :
0000 130 : V03-023 WMC0003 Wayne Cardoza 13-Oct-1983
0000 131 : Reserve space for 4 extra kstack pages.
0000 132 :
0000 133 : V03-022 LJK0251 Lawrence J. Kenah 7-Sep-1983
0000 134 : Add CLI name counted string to CLI data page.
0000 135 :
0000 136 : V03-021 LJK0246 Lawrence J. Kenah 24-Aug-1983
0000 137 : Add cell to P1 pointer page that records the size of the
0000 138 : user stack.
0000 139 :
0000 140 : V03-020 WMC0002 Wayne Cardoza 22-Aug-1983
0000 141 : Temporarily increase kernel stack to 4 pages.
0000 142 :
0000 143 : V03-019 LJK0239 Lawrence J. Kenah 1-Aug-1983
0000 144 : Increase size of exec stack to 16 pages.
0000 145 :
0000 146 : V03-018 RAS0159 Ron Schaefer 22-Jul-1983
0000 147 : Add symbol PIOUS_EODSTR to specify the length of the
0000 148 : $EOD string for SYSS$INPUT.
0000 149 :
0000 150 : V03-017 LJK0208 Lawrence J. Kenah 26-May-1983
0000 151 : Add listheads for image control block lists to image
0000 152 : activator context page. Change name of image activator
0000 153 : work area.
0000 154 :
0000 155 : V03-016 RSH0017 R. Scott Hanna 21-May-1983
0000 156 : Change CTL$GL_AUDITCHAN to CTL$GL_NSA_IDIPTR. This cell
0000 157 : contains a pointer to the security auditing impure data
0000 158 : table. The audit channel number is now in the IDT.
0000 159 :
0000 160 : V03-015 PCA1016 Paul C. Anagnostopoulos 28-Apr-1983
0000 161 : Add CTL$AG_CLITABLE, two longwords which specify the
0000 162 : virtual address range into which the CLI table is mapped.
0000 163 :
0000 164 : V03-014 GAS0122 Gerry Smith 14-Apr-1983
0000 165 : Add CTL$GQ_HELPFLAGS, two longwords of information on
0000 166 : current help settings. Also double the size of the
0000 167 : supervisor stack, to 32 pages.
0000 168 :
0000 169 : V03-013 RSH0012 R. Scott Hanna 13-Mar-1983
0000 170 : Add the cell CTL$GL_AUDITCHAN to store the security
0000 171 : audit trail journal channel number.
```

0000	172	:			
0000	173	:	V03-012	RSH0002	R. Scott Hanna 9-Feb-1983
0000	174	:		Add CTL\$GL_RDIPT	to support rights database system
0000	175	:		services.	
0000	176	:			
0000	177	:	V03-011	KBT0454	Keith B. Thompson 7-Jan-1983
0000	178	:		Make directory cache	2 pages due to long directory names.
0000	179	:			
0000	180	:	V03-010	JWH0138	Jeffrey W. Horn 29-Nov-1982
0000	181	:		Fix problems related	with movement within the
0000	182	:		vector page introduced	by JWH0118 and DMW4008.
0000	183	:			
0000	184	:	V03-009	JWH0118	Jeffrey W. Horn 19-Nov-1982
0000	185	:		Re-organize RMS	Impure Areas. Add cells needed to
0000	186	:		support P1 allocation	routines. Eliminate the
0000	187	:		RMS Process IO	segment and the Process Allocation
0000	188	:		Region from SHELL,	now created in PROCSTRT.
0000	189	:			
0000	190	:	V03-008	DMW4008	DMWalp 12-Nov-1982
0000	191	:		Add definations	for new logical name structures.
0000	192	:			
0000	193	:	V03-007	WMC0001	Wayne Cardoza 19-Oct-1982
0000	194	:		Add support for	specifying desired page file.
0000	195	:			
0000	196	:	V03-006	JWH0001	Jeffrey W. Horn 27-Aug-1982
0000	197	:		Add cells needed	to support RMS Recovery Units.
0000	198	:		PIO\$GL_RULOCK	and PIO\$GL_NXTIRBSEQ.
0000	199	:			
0000	200	:	V03-005	CDS0001	C Saether 13-Aug-1982
0000	201	:		Define CTL\$GL_F11	BXQP cell to dispatch to XQP.
0000	202	:			
0000	203	:	V03-004	LJK47269	Lawrence J. Kenah 10-Aug-1982
0000	204	:		Change protection	on image activator scratch pages to
0000	205	:		UREW. Remove	\$PRDEF.
0000	206	:			
0000	207	:	V03-004	KDM46395	Kathleen D. Morse 28-Jun-1982
0000	208	:		Increment the	PTWSLELCK byte array for the P1 system
0000	209	:		service vector	pages, which are WINDOW PTEs.
0000	210	:			
0000	211	:	V03-002	KDM0002	Kathleen D. Morse 28-Jun-1982
0000	212	:		Added \$PRDEF.	
0000	213	:			
0000	214	:			

```

0000 216          .SBTTL  DECLARATIONS
0000 217
0000 218          :
0000 219          : INCLUDE FILES:
0000 220          :
0000 221          $DYNDEF          ;DYNAMIC DATA STRUCTURE TYPE DEFINITIONS
0000 222          $FWADEF          ;RMS FILE WORK AREA CONSTANTS
0000 223          $IAFDEF          ;IMAGE ACTIVATOR FIXUP VECTOR OFFSETS
0000 224          $IMPDEF          ;RMS INTERNAL STRUCTURE DEFINITION
0000 225          $IPLDEF          ;DEFINE INTERRUPT LEVELS
0000 226          $JIBDEF          ;JOB INFORMATION BLOCK DEFINITIONS
0000 227          $NSAIDTDEF       ;SECURITY AUDITING IMPURE DATA TABLE OFFSETS
0000 228          $PCBDEF          ;PROCESS CONTROL BLOCK DEFINITIONS
0000 229          $PFLDEF          ;PAGE FILE DEFINITIONS
0000 230          $PFNDEF          ;PFN DATA BASE DEFINITIONS
0000 231          $PHDDEF          ;PROCESS HEADER DEFINITIONS
0000 232          $PQBDEF          ;PROCESS QUOTA BLOCK DEFINITIONS
0000 233          $PRTDEF          ;DEFINE PROTECTION CODES
0000 234          $PSLDEF          ;DEFINE PSL FIELDS
0000 235          $PTEDEF          ;PAGE TABLE ENTRY DEFINITIONS
0000 236          $SECDEF          ;SECTION DEFINITIONS (GSTE/PSTE)
0000 237          $SGNDEF          ;SYSGEN VALUE DEFINITIONS
0000 238          $VADEF          ;DEFINE VIRTUAL ADDRESS FIELDS
0000 239          $WSLDEF          ;WORKING SET LIST DEFINITIONS
0000 240
0000 241          :
0000 242          : EQUATES:
0000 243          :
0000 244
00000000 0000 245 PFILPGCNT=0          ; INITIALIZE COUNT OF PAGE FILE PAGES
00000003 0000 246 KSTACK=3          ; THREE PAGES OF KERNEL STACK
00000004 0000 247 KSTACK_EX=4       ; KERNEL STACK EXPANSION PAGES
00000010 0000 248 ESTACK=16        ; SIXTEEN PAGES OF EXEC STACK
00000020 0000 249 SSTACK=32       ; THIRTY TWO PAGES OF SUPER STACK
00000008 0000 250 IMGACTBUF=8      ; EIGHT PAGES FOR IMAGE ACTIVATE BUFFER
0000 251          ; FOUR FOR TOP LEVEL CALL, FOUR FOR
0000 252          ; ONE RECURSIVE CALL.
00000004 0000 253 KRP_COUNT=4      ; SIZE IN PAGES OF P1 LOOKASIDE LIST
0000 254
00000001 0000 255 DBGPTCNT=1       ; NUMBER OF DEBUG AREA PAGE TABLES
0000 256
0000 257          .IF      NDF,LIBSWITCH ; IF EXECUTIVE, DEFINE THESE GLOBALLY
00000001 0000 258 SWP$C_DBGPTCNT==DBGPTCNT
00000003 0000 259 SWP$C_KSTACK==KSTACK
00000004 0000 260 SWP$C_KSTACK_EX==KSTACK_EX
00000004 0000 261 CTL$C_KRP_COUNT==KRP_COUNT
00000200 0000 262 CTL$C_KRP_SIZE==512          ; EACH KRP PACKET IS 512 BYTES LONG
0000 263          .ENDC
0000 264
00000004 0000 265 COMMON=4          ; FOUR PAGES FOR USER COMMON
0000 266          ; FOUR PAGES FOR DEC COMMON
0000 267
0000 268          :
0000 269          : MACROS:
0000 270          :
0000 271
0000 272          .MACRO  WSL      SYM,TYPE          ; MAKE VALID WS LIST ENTRY

```



```
0000 273      .LONG  SYM!WLSM_VALID!WLSLSC_'TYPE      ; SET VALID BIT
0000 274      .ENDM  WSL
0000 275
0000 276      .MACRO  CTL      SYM
0000 277      .IF    NDF,LIBSWITCH
0000 278  CTL$'SYM==.-CTLVECPAG+VECTORS
0000 279      .IFF
0000 280  CTL$'SYM==.-CTLVECPAG+VECTORS
0000 281      .ENDC
0000 282      .ENDM  CTL
0000 283
0000 284      .MACRO  PHD      SYM
0000 285      .=PHD...+PHD$'SYM
0000 286      .ENDM  PHD
0000 287
0000 288
0000 289      .MACRO  PIO      SYM      ; DEFINE PROCESS I/O SEGMENT SYMBOL
0000 290  PIO$'SYM==PIOBASE+<.-PIO>      ;
0000 291      .ENDM  PIO
0000 292
0000 293      .MACRO  PIOL     SYM      ; DEFINE PROCESS I/O LOCAL SYMBOL
0000 294  PIOL'SYM=PIOBASE+<.-PIO>      ;
0000 295      .ENDM  PIOL      ;
0000 296
0000 297      .MACRO  PIO2     SYM      ; DEFINE PROCESS I/O LOCAL SYMBOL
0000 298  PIO2'SYM=PIOTBL+<.-TBL>      ;
0000 299      .ENDM  PIO2
0000 300
0000 301      ;
0000 302      ; P1PTE - GENERATE PTES FOR CONTROL REGION PAGE TABLE
0000 303      ;
0000 304
0000 305      .MACRO  P1PTE   NUM,ACCESS=NA,OWNER=K,STATE,SYM
0000 306      .IF    NB,SYM
0000 307      SYM=VA...
0000 308      .ENDC
0000 309
0000 310      .REPT  NUM
0000 311      .=.-4
0000 312      .IF    IDN,<STATE>,<VALID>
0000 313      TMP...=<1@31>
0000 314      .IFF
0000 315      TMP...=0
0000 316      .ENDC
0000 317
0000 318      .IF    IDN,<STATE>,<DZRO>
0000 319      PFILPGCNT=PFILPGCNT+1
0000 320      .ENDC
0000 321
0000 322      .IF    IDN,<STATE>,<PFIL>
0000 323      TMP...=PTESM TYP1
0000 324      PFILPGCNT=PFILPGCNT+1
0000 325      .ENDC
0000 326      .LONG  PTESC_'ACCESS!PTESC_'OWNER'OWN!TMP...
0000 327
0000 328      .=.-4
0000 329      VA...=VA...-^X200
```

SHELL
V04-000

PROCESS DEFINITION
DECLARATIONS

J 8

16-SEP-1984 01:11:32 VAX/VMS Macro V04-00
5-SEP-1984 03:47:44 [SYS.SRC]SHELL.MAR;1

Page 7
(1)

SH
VO

0000 330
0000 331
0000 332

.ENDR
.ENDM P1PTE

```
0000 334
0000 335 .IF DF,LIBSWITCH ; FOR USRVECTOR MODULE,
0000 336 .PSECT $AB$$,ABS ; DO NOT GENERATE ANY STORAGE
0000 337 .=0
0000 338 .IFF ; ELSE,
0000 339
00000000 340 .PSECT AEXENONPAGED
00000007' 0000 341 SWP$GL_SHELLIO:: ; I/O PAGE COUNT FOR SHELL
0004 342 .LONG SWP$C_SHELLSIZ ;
0004 343 :
0004 344 : WARNING: SWP$C_SHLP1PT MUST BE MANUALLY COMPUTED SINCE IT IS USED
0004 345 : TO ALLOCATE THE SPACE FOR P1 PAGE TABLE ENTRIES. IF UPDATES
0004 346 : ARE MADE THAT INTRODUCE MORE P1 PAGE TABLE ENTRIES, THEN
0004 347 : VERIFY THAT THEY STILL FIT IN THE NUMBER OF PAGES PROVIDED
0004 348 :
00000002 0004 349 SWP$C_SHLP1PT==2 ; COUNT OF BASIC P1 PAGE TABLES FOR SHELL
00000000 350 .PSECT YYSHELLPAGED,PAGE ; SHELL AT END OF PAGEABLE EXEC
0000 351 SHELL: ; BASE OF SHELL SWAP IMAGE
0000 352 SWP$GL_SHELLBAS:: ;
```

```

0000 354 .SBTTL SHELL PROCESS HEADER
0000 355 :
0000 356 : PROCESS HEADER FOR SHELL PROCESS
0000 357 :
0000 358 :
00000000 0000 359 PHD...=. : BASE OF PROCESS HEADER
0000017C 0000 360 .BLKB PHD$C_LENGTH : RESERVE SPACE FOR HEADER
017C 361 :
017C 362 : WORKING SET LIST
017C 363 :
017C 364 :
017C 365 : PAGES DESCRIBED IN THE WORKING SET LIST MUST OCCUR IN THE SAME ORDER
017C 366 : WITHIN THE PROCESS BODY.
017C 367 :
0000005F 017C 368 .ALIGN LONG : AT LONGWORD RESOLUTION
0000018C 017C 369 SWP$C_KSTACK_EX_WSL==<.-PHD...>a-2
0000018C 017C 370 .BLKB KSTACK_EX : EXTRA SLOTS FOR KERNEL STACK EXPANSION
00000063 018C 371 WSL...=. : BASE OF WORKING SET LIST
00000063 018C 372 SWP$C_KSTACK_WSL==<.-PHD...>a-2
018C 373 WSL <<KSPINI-<1*512>>!WSL$M_WSLOCK>,PROCESS; KERNEL STACK PAGE 1
0190 374 WSL <<KSPINI-<2*512>>!WSL$M_WSLOCK>,PROCESS; KERNEL STACK PAGE 2
0194 375 WSL <<KSPINI-<3*512>>!WSL$M_WSLOCK>,PROCESS; KERNEL STACK PAGE 3
00000198 0198 376 NOT_KSTACK_WSL=. : REST OF WSL IS NOT KERNEL STACK
0198 377 WSC <VECTORS!WSL$M_WSLOCK>,PROCESS ; VECTOR PAGE
0000019C 019C 378 DYNWSL=. : START OF DYNAMIC WORKING SET
019C 379 WSL <PIOBASE!WSL$M_MODIFY>,PROCESS ; PROCESS I/O SEGMENT BASE PAGE 1
01A0 380 :
000001A0 01A0 381 FWSL...=. : FREE WORKING SET LIST
00000001 01A0 382 NDYN=<FWSL...-DYNWSL>a-2 : NUMBER OF DYNAMIC ENTRIES IN SHELL
00000002 01A0 383 SHUFFLE_COUNT=<FWSL...-NOT_KSTACK_WSL>a-2 : AMOUNT OF SHUFFLING BY SHELINIT
00000001 01A0 384 SWP$C_NDYN=NDYN : GLOBAL NAME FOR NUMBER OF DYNAMIC PAGES
01A0 385 .ALIGN PAGE : GET TO END OF PAGE
00000200 0200 386 SAV...=. :
0200 387 :
00000067 0200 388 TMP...=<DYNWSL-PHD...>a-2 : BASE OF FLUID WORKING SET LIST
0200 389 :
0067 000C 390 PHD W WSLOCK : POINTER TO START OF LOCKED PAGES
000E 391 .WORD TMP... :
000E 392 :
0067 000E 393 PHD W WSDYN : POINTER TO START OF DYNAMIC PAGES
000E 394 .WORD TMP... :
0010 395 :
0063 0010 396 PHD W WSLIST :
0008 397 .WORD <WSL...-PHD...>a-2 : START OF WORKING SET LIST
000A 398 PHD W WSNEXT : NEXT WORKING SET ENTRY
0067 0010 399 .WORD <TMP...+NDYN-1> :
0012 400 :
7FFE0000' 0012 401 PHD L FREP1VA : VA OF FIRST FREE PAGE IN P1 SPACE
0030 402 .LONG VA...-512 :
0034 403 :
0034 404 PHD Q PRIVMSK : ENABLE ALL PRIVILEGES
FFFFFFFF FFFFFFFF 0000 405 .LONG -T,-1 :
0008 406 :
0008 407 PHD W QUANT : QUANTUM OF ONE SECOND
FF9C 003C 408 .WORD -T00 :
003E 409 :
003E 410 PHD L_KSP : STACK POINTERS

```

7FFE7E00'	0078	411	.LONG	KSPINI	:	KERNEL STACK POINTER
7FFE9E00'	007C	412	.LONG	ESPINI	:	EXECUTIVE STACK POINTER
7FFEDE00'	0080	413	.LONG	SSPINI	:	SUPER STACK POINTER
	0084	414			:	NO INITIAL USER STACK
	0084	415			:	
	0084	416	PHD	L_PC	:	INITIAL PROGRAM COUNTER
00000000'	00C0	417	.LONG	EXES\$PROCSTRT	:	START PROCESS
	00C4	418			:	
	00C4	419	PHD	L_PSL	:	PROGRAM STATUS LONGWORD
00020000	00C4	420	.LONG	IPL\$ ASTDEL@PSL\$V_IPL	:	MUST RUN AT ASTDEL
	00C8	421	PHD	L_POBR	:	P0 BASE REGISTER
00000000	00C8	422	.LONG	0	:	RELATIVE OFFSET
	00CC	423			:	
	00CC	424	PHD	L_POLRASTL	:	NO AST PENDING, POLR = 0
04000000	00CC	425	.LONG	<4@PHD\$V_ASTLVL>	:	
	00D0	426			:	
	00D0	427	PHD	L_P1BR	:	SVA OF P1 PAGE TABLE BASE
FF800000	00D0	428	.LONG	-21@23>	:	
	00D4	429			:	
	00D4	430	PHD	L_P1LR	:	P1 LENGTH REGISTER
001FFF01'	00D4	431	.LONG	<T@21>-P1PTLEN	:	
	00D8	432			:	
	00D8	433	PHD	L_FREPTECNT	:	FREE PTE'S BETWEEN P0 AND P1 PT
FFFFFF00'	002C	434	.LONG	-21+<<1@31-VA...>@-9>>	:	(NUMBER OF AVAILABLE LONGWORDS)
	0030	435			:	
	0030	436	PHD	W_PRCLM	:	SUB PROCESS LIMIT
0004	003E	437	.WORD	4	:	ALLOW FOUR
	0040	438			:	
	0040	439	PHD	W_WSFLUID	:	GUARANTEED FLUID PAGE COUNT
000A	0074	440	.WORD	10	:	GUESS AT TEN FOR NOW (WAG)
	0076	441			:	
	0076	442	PHD	B_DFPFC	:	DEFAULT PAGE FAULT CLUSTER
10	0034	443	.BYTE	16	:	
	0035	444			:	
	0035	445	PHD	B_CPUMODE	:	ACCESS MODE FIELD FOR CPU TIME LIMIT
03	0060	446	.BYTE	PSL\$C_USER	:	EXPIRATION AST STARTS WITH USER MODE
	0061	447			:	
00000200	0061	448	.=SAV...		:	RESTORE LOCATION COUNTER

```

0200 450 .SBTTL SHELL PROCESS PAGE TABLES
0200 451 :
0200 452 :
0200 453 : SHELL PROCESS PAGE TABLES
0200 454 :
0200 455 :
0200 456 :
00000200 0200 457 P1PTBAS=. ;
0200 458 :
00000400 0200 459 TMP=<SWP$C_SHLP1PT*512> ; ALLOCATE SPACE FOR PAGE TABLES
00000600 0200 460 .BLKB TMP ; ALLOCATE SPACE FOR PAGE TABLES
00000600 0600 461 SAV...=. ; REMEMBER BASE OF PAGE TABLES
0600 462 :
0600 463 .IFTF ; ASSEMBLE FOLLOWING FOR BOTH
0600 464 : SHELL AND USRVECTOR MODULES:
0600 465 :
0600 466 :
0600 467 : INITIALIZE VIRTUAL ADDRESS BASE
0600 468 :
80000000 0600 469 VA...=1031 ;
0600 470 :
0600 471 ; P1PTE 128*DBGPTCNT ; DEBUGGER CONTEXT AREA
7FFF0000 0600 472 VA...=VA...-<DBGPTCNT*128*512> ; ACCOUNT FOR IT IN VA...
7FFF0000 0600 473 DBGAREA=VA... ; ADDRESS OF DEBUG AREA
0600 474 P1PTE 1,URKW,K,PFIL ; VECTOR PAGE
7FFEF000 05FC 475 VECTORS=VA... ;
05FC 476 :
05FC 477 .IFF ; ASSEMBLE REST ONLY FOR SHELL MODULE:
05FC 478 :
7FFEF000 05FC 479 CTL$GL_VECTORS==VECTORS ; BASE OF VECTOR PAGE
05FC 480 P1PTE 16 ; PAGES FOR SYSTEM SERVICE VECTORS
7FFEDE00 05BC 481 P1SYSVECTORS==VA... ;
05BC 482 P1PTE SSTACK,URSW,S,DZRO,SSPINI ; SUPERVISOR STACK
053C 483 P1PTE ESTACK,SREW,E,DZRO,ESPINI ; EXECUTIVE STACK
04FC 484 P1PTE KSTACK,SRKW,K,PFIL,KSPINI ; KERNEL STACK
7FFE7800 04F0 485 CTL$GL_KSTKBAS==VA... ; MAXIMUM TOP OF KERNEL STACK
04F0 486 P1PTE KSTACK_EX,NA,K,PFIL ; EXPANSION FOR KERNEL STACK
7FFE7000 04E0 487 CTL$GL_KSTKBASEXP==VA... ; MAXIMUM TOP OF EXPANSION KERNEL STACK
7FFE7E00 04E0 488 CTL$GL_KSPINI==KSPINI ; INITIAL KERNEL STACK
04E0 489 P1PTE 1 ; NULL STOPPER PAGE
04DC 490 P1PTE KRP_COUNT,URKW,K,DZRO ; P1 LOOKASIDE LIST FOR KERNEL MODE
7FFE6600 04CC 491 CTL$GL_KRP==VA... ; ADDRESS USED BY PROCSTRT TO FIND IT
04CC 492 P1PTE 1,URSW,E,DZRO ; IMAGE HEADER BUFFER
7FFE6400 04C8 493 MM$IMGHDRBUF==VA... ; ADDRESS OF IMAGE HEADER BUFFER
04C8 494 P1PTE 2,UREW,K,DZRO ; VECTORS FOR USER SYS SRV & ERR MSGS
7FFE6000 04C0 495 CTL$A_DISPVEC==VA... ; ADR OF VECTOR PAGES FOR SYS SRV/ERRMSG
04C0 496 P1PTE 4,UW,U,DZRO ; DEBUGGER CONTEXT
04B0 497 P1PTE IMGACTBUF,UREW,E,DZRO ; IMAGE ACTIVATOR SCRATCH PAGES (8)
7FFE4800 0490 498 IACSAL_IMGACTBUF==VA... ; ADR OF IMAGE ACTIVATOR SCRATCH PAGES
7FFE4800 0490 499 CLIDATAEND=VA... ;
0490 500 P1PTE 12,URSW,S,DZRO ; COMMAND INTERPRETER DATA PAGES
7FFE3000 0460 501 CLIDATAPAG=VA... ; BASE ADDRESS OF CLI DATA PAGE
0460 502 P1PTE 1,UREW,E,DZRO ; IMAGE ACTIVATOR CONTEXT
7FFE2E00 045C 503 IMGACTCTX=VA... ;
045C 504 P1PTE NSASK_IDT_PAGES,KW,K,DZRO ; SECURITY AUDITING IMPURE DATA TABLE
7FFE2800 0450 505 NSAST_IDT=VA... ;
0450 506 P1PTE 2 ; UNUSED PAGES

```

```

0448 507 :
0448 508 : THE FOLLOWING PAGE IS A USER-MODE WRITABLE CONTEXT PAGE
0448 509 : FOR MISC. USER-MODE PACKAGES THAT ARE MAINTAINED BY VMS.
0448 510 : (SEE DETAILED ALLOCATION OF PAGE BELOW)
0448 511 :
7FFE2200 0448 512 P1PTE 1,UW,K,DZRO ; GLOBAL USER-MODE OWN STORAGE PAGE
0444 513 UWVECPAG=VA...
0444 514 :
0444 515 : THE FOLLOWING PAGE IS A WRITABLE CONTEXT PAGE FOR
0444 516 : THE USE OF THE COMPATIBILITY MODE EMULATOR AND EXCEPTION
0444 517 :
7FFE1E00 0444 518 P1PTE 2,UW,K,DZRO ; WRITABLE PAGES FOR COMPATIBILITY MODE
043C 519 CTLSAG_CMEDATA==VA...
043C 520 :
043C 521 : THE FOLLOWING PAGES ARE FOR USE BY BASIC/BASIC-PLUS TO PROVIDE
043C 522 : THE "CORE COMMON" REQUIRED TO PASS DATA WHEN CHAINING FROM IMAGE
043C 523 : TO IMAGE. AN IDENTICAL AREA, NEGATIVELY DISPLACED HAS BEEN ALLOCATED
043C 524 : FOR USERS AND CSS.
043C 525 :
7FFE1600 043C 526 P1PTE COMMON,UW,K,DZRO ; DEC "CORE COMMON" PAGES
042C 527 CTLSA_COMMON==VA... ; BASE ADDRESS
042C 528 P1PTE COMMON,UW,K,DZRO ; USER "CORE COMMON" PAGES
041C 529 P1PTE 1,UREW,E,DZRO ; PROCESS IFB/IRB TABLES
7FFE0C00 0418 530 PIOTBL=VA...
0418 531 P1PTE 2,UPEW,E,DZRO ; RMS DIRECTORY CACHE PAGES
7FFE0800 0410 532 PIOSA_DIRCACHE==VA...
0410 533 P1PTE 1,UREW,E,DZRO ; RMS TRACEPOINT PAGE
7FFE0600 040C 534 PIOSA_TRACE==VA...
040C 535 P1PTE 1,UREW,E,DZRO ; EXTENSION TO RMS POINTER PAGE
0408 536 P1PTE 1,UREW,E,PFIL ; RMS POINTER PAGE
7FFE0200 0404 537 PIOBASE=VA...
7FFE0200 0404 538 USPINI=VA... ; USER STACK AREA
0404 539 :
00000404 0404 540 END...=. ; END OF CONTROL REGION
0404 541 :
000000FF 0404 542 P1PTLEN=<1@31-VA...>@-9 ; ENTRIES IN P1PT
0404 543 :
00000081 0404 544 SWP$C_S4LFPTE==<<.-P1PTBAS>@-2> ; FREE P1PT IN BASIC P1 PAGE TABLES
0404 545 :
0404 546 : RESTORE LOCATION POINTER
0404 547 :
00000600 0404 548 .=SAV... ; FINISHED WITH PAGE TABLES
00000068 0600 549 PFILPGCNT=PFILPGCNT-KSTACK-1 ; NO PAGE FILE FOR KERNEL STACK OR VECTORS
00000068 0600 550 SWP$C_SHELLPFIL==PFILPGCNT ; DEFINE GLOBAL VALUE FOR SHELL PAGE FILE
0600 551 : REQUIREMENT

```

```

0600 553 .SBTTL BODY OF SHELL PROCESS
0600 554 :
0600 555 : BODY OF SHELL PROCESS
0600 556 :
00000600 0600 557 : .=SAV... ; POSITION TO END OF HEADER
0600 558 :-----
0600 559 :
0600 560 : VECTOR PAGE
0600 561 : ***** NOTE: The cells in this page must not move. There are
0600 562 : ***** facilities which locate cells via offsets from
0600 563 : ***** the top of this page. If you delete a cell
0600 564 : ***** substitute a .LONG 0 and comment it as SPARE. If
0600 565 : ***** you add a cell either replace a SPARE cell or
0600 566 : ***** add it to the end.
0600 567 :-----
00000600 0600 568 :
00000600 0600 569 SWP$AL_PTRPAG==.-SHELL ; ADDRESS IN SWAPPER MAP
0600 570 :
0600 571 .IFT ; IF USRVECTOR MODULE
0600 572 .=VECTORS ; SHOW ACTUAL ADDRESSES IN LISTING
0600 573 :
0600 574 .IFTF ; ASSEMBLE FOLLOWING FOR BOTH
0600 575 : SHELL AND USRVECTOR MODULES:
0600 576 :
00000600 0600 577 CTLVECPAG=. ; BASE OF VECTOR PAGE
0600 578 CTL GW_NMIOCH ; NUMBER OF CHANNELS
0000 0600 579 .WORD 0 ; FILLED IN IN PROCSTRT
0602 580 :
0602 581 CTL GW_CHINDX ; MAXIMUM CHANNEL INDEX
0000 0602 582 .WORD 0 ; FILLED IN IN PROCSTRT
0604 583 :
0604 584 CTL GL_LNMHASH ; process Logical Name HASH table
00000000 0604 585 .LONG 0 ; pointer
0608 586 CTL GL_LNMDIRECT ; process Logical Name DIRECTORY
00000000 0608 587 .LONG 0 ; pointer
060C 588 :
060C 589 :
060C 590 : ARRAY OF STACK POINTERS
060C 591 :
060C 592 : THE -1 ENTRY OF CTL$AL_STACK IS USED TO DETERMINE THE MAXIMUM
060C 593 : EXTENT OF THE KERNEL STACK.
060C 594 :
060C 595 .LONG CTL$GL_KSTKBAS ; MAXIMUM TOP OF KERNEL STACK
0610 596 CTL AL_STACK ; STACK POINTER RESTART ARRAY
0610 597 .LONG KSPINI ; KERNEL STACK POINTER
0614 598 .LONG ESPINI ; EXEC STACK POINTER
0618 599 .LONG SSPINI ; SUPERVISOR STACK POINTER
061C 600 .LONG USPINI ; USER STACK POINTER
0620 601 :
0620 602 :
0620 603 : LOGICAL NAME TABLE TRANSLATION CACHE QUEUE HEADER
0620 604 :
0620 605 :
0620 606 CTL GQ_LNMTBLCACHE ; QUEUE HEADER FOR LOGICAL NAME TABLE
0620 607 .LONG CTL$GQ_LNMTBLCACHE ; TRANSLATION CACHE (FORWARD LINK)
0624 608 .LONG CTL$GQ_LNMTBLCACHE ; (BACK LINK)
0628 609 :

```



```

0628 610 :
0628 611 : THE FOLLOWING THREE VECTORS MUST BE ADJACENT
0628 612 :
0628 613 :
00000000 0628 614 CTL GL_CMSUPR ; SUPERVISOR CHANGE MODE DISPATCHER
          0628 615 .LONG 0 ; NULL
          062C 616
00000000 062C 617 CTL GL_CMUSER ; USER CHANGE MODE DISPATCHER
          062C 618 .LONG 0 ; NULL
          0630 619
00000000 0630 620 CTL GL_CMHANDLR ; COMPATIBILITY MODE HANDLER
          0630 621 .LONG 0
          0634 622
00000000 0634 623 CTL AQ_EXCVEC ; PRIMARY/SECONDARY EXCEPTION VECTORS
          0634 624 .LONG 0 ; KERNEL MODE PRIMARY
00000000 0638 625 .LONG 0 ; KERNEL MODE SECONDARY
00000000 063C 626 .LONG 0 ; EXEC MODE PRIMARY
00000000 0640 627 .LONG 0 ; EXEC MODE SECONDARY
00000000 0644 628 .LONG 0 ; SUPERVISOR MODE PRIMARY
00000000 0648 629 .LONG 0 ; SUPERVISOR MODE SECONDARY
00000000 064C 630 .LONG 0 ; USER MODE PRIMARY
00000000 0650 631 .LONG 0 ; USER MODE SECONDARY
          0654 632
00000000 0654 633 CTL GL_THEXEC ; EXECUTIVE TERMINATION HANDLER
          0654 634 .LONG 0 ; NULL
          0658 635
00000000 0658 636 CTL GL_THSUPR ; SUPERVISOR TERMINATION HANDLER
          0658 637 .LONG 0 ; NULL
          065C 638
00000000 065C 639 : CTL GL_THUSER ; USER TERMINATION HANDLER
          065C 640 .LONG 0 ; NULL
          0660 641
          0660 642 : FORMER LOCATION OF COMPATIBILITY MODE CONTEXT
00000800 0660 643 CTL GQ_COMMON ; CORE COMMON DESCRIPTOR
7FFE 600 0660 644 .LONG 512*COMMON ; SIZE IN BYTES
          0664 645 .LONG CTL$A_COMMON ; AND ADDRESS
          0668 646
00000000 0668 647 CTL GL_GETMSG ; PER-PROCESS VECTOR TO USER MESSAGE
          0668 648 .LONG 0 ; DISPATCHER
          066C 649
          066C 650 CTL AL_STACKLIM ; STACK LIMIT ARRAY (INDEXED BY MODE)
7FFE7800 066C 651 .LONG CTL$GL_KSTKBAS ; KERNEL STACK LO (TOP) LIMIT
7FFE7E00 0670 652 .LONG KSPINI ; EXEC STACK LO (TOP) LIMIT
7FFE9E00 0674 653 .LONG ESPINI ; SUPER STACK LO (TOP) LIMIT
00000000 0678 654 .LONG 0 ; USER STACK (NOT CHECKED)
          067C 655
00000000 067C 656 CTL GL_CTLBASVA ; BASE CONTROL REGION ADDRESS
          067C 657 .LONG 0 ; FILLED IN BY PROCSTRT
          0680 658
          0680 659 CTL GL_IMGHDRBF ; ADDRESS OF IMAGE ACTIVATOR'S IMAGE
00000000 0680 660 .LONG 0 ; HEADER BUFFER, IF IMAGE IS ACTIVE;
          0684 662 ; 0 IF NO IMAGE ACTIVE
          0684 663
7FFE2E7C' 0684 664 CTL GL_IMGLSTPTR ; ADDRESS OF ICB LIST (FOR DEBUGGER)
          0684 664 .LONG IAC$GL_IMAGE_LIST
          0688 665
          0688 666 CTL GL_PHD ; ADDRESS OF PHD WINDOW

```

```

00000000 0688 667 .LONG 0 ; INIT BY INSWAP
068C 668
00000000 00000000 068C 669 CTL GQ_ALLOCREG ; HEAD OF PROCESS ALLOCATION REGION POOL
068C 670 .LONG 0,0 ; (Filled in by PROCSTRT)
0694 671
0694 672 CTL GQ_MOUNTLST ; MOUNTED DEVICE LIST
7FFEFE94 0694 673 .LONG CTL$GQ_MOUNTLST ; FORWARD LINK
7FFEFE94 0698 674 .LONG CTL$GQ_MOUNTLST ; BACKWARD LINK ( LIST EMPTY )
069C 675
069C 676 ;*****
069C 677 ; NOTE: The order of the accounting data between CTLST_USERNAME and
069C 678 ; CTLST_REMOTEID should be preserved!
069C 679 ;*****
069C 680
000006A8 069C 681 CTL T_USERNAME ; USERNAME
069C 682 .BLKB JIBSS_USERNAME ; Filled in by PROCSTRT
06A8 683
000006B0 06A8 684 CTL T_ACCOUNT ; ACCOUNT NAME
06A8 685 .BLKB JIBSS_ACCOUNT ; Filled in by PROCSTRT
06B0 686
ASSUME <CTLST_ACCOUNT - CTLST_USERNAME> EQ <JIBST_ACCOUNT - JIBST_USERNAME>
06B0 687
06B0 688
00000000 00000000 06B0 689 CTL GQ_LOGIN ; SYSTEM TIME AT PROCESS CREATION
06B0 690 .LONG 0,0 ;
06B8 691
00000000 06B8 692 CTL GL_FINALSTS ; FINAL EXIT STATUS FOR PROCESS
06B8 693 .LONG 0 ;
06BC 694
00000000 06BC 695 CTL GL_WSPEAK ; PEAK WORKING SET SIZE
06BC 696 .LONG 0 ;
06C0 697
00000000 06C0 698 CTL GL_VIRTPEAK ; PEAK VIRTUAL SIZE
06C0 699 .LONG 0 ;
06C4 700
00000000 06C4 701 CTL GL_VOLUMES ; COUNT OF VOLUMES MOUNTED
06C4 702 .LONG 0 ;
06C8 703
00000000 00000000 06C8 704 CTL GQ_ISTART ; IMAGE ACTIVATION TIME
06C8 705 .LONG 0,0 ;
06D0 706
00000000 06D0 707 CTL GL_ICPUTIM ; INITIAL IMAGE CPU TIME
06D0 708 .LONG 0 ;
06D4 709
00000000 06D4 710 CTL GL_IFAULTS ; INITIAL IMAGE FAULT COUNT
06D4 711 .LONG 0 ;
06D8 712
00000000 06D8 713 CTL GL_IFAULTIO ; INITIAL IMAGE FAULT I/O COUNT
06D8 714 .LONG 0 ;
06DC 715
00000000 06DC 716 CTL GL_IWSPEAK ; IMAGE WORKING SET PEAK
06DC 717 .LONG 0 ;
06E0 718
00000000 06E0 719 CTL GL_IPAGEFL ; IMAGE PAGE FILE PEAK USAGE
06E0 720 .LONG 0 ;
06E4 721
00000000 06E4 722 CTL GL_IDIOCNT ; INITIAL IMAGE DIRECT I/O COUNT
06E4 723 .LONG 0 ;

```

PROCESS DEFINITION
BODY OF SHELL PROCESS

```

00000000 06E8 724          CTL  GL_IBIOCNT      ; INITIAL IMAGE BUFFERED I/O COUNT
          06E8 725          .LONG 0
          06E8 726
          06EC 727
00000000 06EC 728          CTL  GL_IVOLUMES    ; INITIAL IMAGE VOLUME MOUNT COUNT
          06EC 729          .LONG 0
          06F0 730
          00 06F0 731          CTL  T_NODEADDR     ; REMOTE NODE ADDRESS (BINARY)
          000006F7 06F0 732          .BYTE 0                ; BYTE COUNT BYTE
          06F1 733          .BLKB 6                ; 6 BYTES MAX
          06F7 734
          00 06F7 735          CTL  T_NODENAME     ; REMOTE NODE NAME (ASCII)
          000006FE 06F7 736          .BYTE 0                ; BYTE COUNT BYTE
          06F8 737          .BLKB 6                ; 6 CHARACTERS MAX
          06FE 738
          00 06FE 739          CTL  T_REMOTEID     ; REMOTE ID
          0000070F 06FE 740          .BYTE 0                ; BYTE COUNT BYTE
          06FF 741          .BLKB 16               ; 16 CHARACTERS MAX
          070F 742
00000710 070F 743          .BLKB 1
          0710 744
          0710 745          ;*****
          0710 746          ; End of adjacent accounting data
          0710 747          ;*****
          0710 748
          0710 749          CTL  GQ_PROCPRIV    ; PROCESS PRIVILEGE MASK
          FFFFFFFF FFFFFFFF 0710 750          .LONG -1,-1          ; ALL PRIVILEGES PERMITTED
          0718 751
          00000000 0718 752          CTL  GL_USRCHMK     ; PER-PROCESS VECTOR TO USER CHANGE MODE
          0718 753          .LONG 0                ; TO KERNEL HANDLER
          071C 754
          00000000 071C 755          CTL  GL_USRCHME     ; PER-PROCESS VECTOR TO USER CHANGE MODE
          071C 756          .LONG 0                ; TO EXECUTIVE HANDLER
          0720 757
          00000000 0720 758          CTL  GL_POWERAST    ; POWER FAIL AST ADDRESS
          0720 759          .LONG 0
          00 0724 760          CTL  GB_PWRMODE     ; ACCESS MODE FOR POWER FAIL AST
          0724 761          .BYTE 0
          0725 762
          00 0725 763          CTL  GB_SSFILTER    ; SYS SERV INHIBIT FILTER MASK
          00000728 0725 764          .BYTE 0
          0726 765          .BLKB 2
          0728 766          ; SPARE

```

```

0728 768
0728 769 :
0728 770 :
0728 771 :
0728 772 :
0728 773 :
0728 774 :
0728 775 :
0728 776 :
00000000' 0728 777 CTL AL FINALEXC ; FINAL EXCEPTION HANDLER ARRAY
00000000' 072C 778 .LONG EXE$EXCPTN ; KERNEL MODE
00000000 0730 779 .LONG EXE$EXCPTNE ; EXECUTIVE MODE
00000000 0734 780 .LONG 0 ; SUPERVISOR MODE
0738 781 .LONG 0 ; USER MODE
0738 782 :
0738 783 :
0738 784 :
00000000 0738 785 CTL GL_CCBASE ; BASE OF I/O CHANNELS
073C 786 .LONG 0 ; FILLED IN IN PROCSTRT
00010000 073C 787 CTL GQ_DBGAREA ; DEBUG AREA
7FFF0000 0740 788 .LONG 64*1024 ; SIZE
0744 789 .LONG DBGAREA ; ADDRESS
0744 790 :
0744 791 :
0744 792 :
00000000 0744 793 CTL GL_RMSBASE ; POINT TO RMS IN SYSTEM SPACE
0748 794 .LONG 0
0748 795 :
0748 796 :
00000000 0748 797 CTL GL_PPMSG ; ADDRESS OF PROCESS PERM. MSG SECTION
0750 798 .QUAD 0 ; STARTING/ENDING ADDRESS OF SECTION
0750 800 CTL GB_MSGMASK ; DEFAULT MESSAGE DISPLAY FLAGS
OF 0750 801 .BYTE 15 ; DEFAULT: FAC,SEV,IDENT AND TEXT (ALL)
0751 802 :
0751 803 CTL GB_DEFLANG ; DEFAULT MESSAGE LANGUAGE
00 0751 804 .BYTE 0 ; (CURRENTLY UNUSED)
0752 805 :
0000 0752 806 CTL GW_PPMSGCHN ; CHANNEL TO PROCESS PERM. MESSAGE
0754 807 .WORD ^ ; SECTION (MAPPED IN CTL$GL_PPMSG)
0754 808 :
00000000 0754 809 CTL GL_USRUNDWN ; PER-PROCESS VECTOR TO USER RUNDOWN
0758 810 .LONG 0 ; SERVICE
0758 811 :
00000000 0758 812 CTL GL_PCB ; ADDRESS OF PROCESS CONTROL BLOCK
0758 813 .LONG 0 ; INIT BY PROCSTRT
075C 814 :
00000000 075C 815 CTL GL_RUF ; POINTER TO RECOVERY UNIT BLOCKS
0760 816 .LONG 0
0760 817 :
00000000 0760 818 CTL GL_SITESPEC ; SITE-SPECIFIC PER-PROCESS CELL
0764 819 .LONG 0
0764 820 :
00000000 0764 821 CTL GL_KNOWNFIL ; PROCESS KNOWN FILE LIST POINTER
0768 822 .LONG 0
0768 823 :
0768 824 CTL AL_IPASTVEC ; VECTOR OF IPAST ADDRESSES

```

```

00000000 00000000 00000000 00000000 0768 825 .LONG 0,0,0,0,0,0,0,0
00000000 00000000 00000000 00000000 0778
0788 826
0788 827 CTL GL_CMCNTX ; ADDRESS OF AME CONTEXT PAGE
7FFE1E00' 0788 828 .LONG CT$AL_CMCNTX
078C 829
078C 830 CTL GL_IAFLNKPTR ; ADDRESS OF IAF LIST (FOR DEBUGGER)
7FFE2E00' 078C 831 .LONG CT$GL_IAFLINK
0790 832
0790 833 CTL GL_F11BXQP ; ADDRESS OF F11B XQP QUEUE AND
00000000 0790 834 .LONG 0 ; DISPATCH VECTORS
0794 835
0794 836 CTL GQ_POALLOC ; HEADER OF PO EXTENTION TO PROCESS
00000000 00000000 0794 837 .LONG 0,0 ; ALLOCATION REGION
079C 838
079C 839 CTL GL_PRCALLCNT ; COUNT OF BYTES OF PROCESS ALLOCATION
00000000 079C 840 .LONG 0 ; REGION USABLE BY IMAGE REQUESTS.
07A0 841
07A0 842 CTL GL_RDIPTR ; POINTER TO RIGHTS DATABASE IDENTIFIER (RDI)
00000000 07A0 843 .LONG 0
07A4 844
07A4 845 CTL GL_LNMDIRSEQ ; SEQUENCE NUMBER FOR CACHE OF LOGICAL
00000000 07A4 846 .LONG 0 ; NAME TABLE TRANSLATIONS
07A8 847
07A8 848 CTL GQ_HELPFLAGS ; HELP FLAGS, ONE LONGWORD FOR USE
00000000 00000000 07A8 849 .LONG 0,0 ; NOW, ONE FOR LATER
07B0 850
07B0 851 CTL GQ_TERMCHAR ; HOME FOR TERMINAL CHARACTERISTICS
00000000 00000000 07B0 852 .LONG 0,0 ; NEED A QUADWORD
07B8 853
07B8 854 CTL GL_KRPFL ; P1 POOL LOOKASIDE LIST FORWARD LINK
7FFEFFB8 07B8 855 .LONG CT$GL_KRPFL
07BC 856 CTL GL_KRPBL ; P1 POOL LOOKASIDE LIST BACKWARD LINK
7FFEFFB8 07BC 857 .LONG CT$GL_KRPFL
07C0 858
07C0 859 CTL GL_CREPRC_FLAGS ; $CREPRC FLAGS
00000000 07C0 860 .LONG 0
07C4 861
07C4 862 CTL GL_THCOUNT ; COUNT OF TERMINATION HANDLERS FOR
00000000 07C4 863 .LONG 0 ; EXEC MODE,
00000000 07C8 864 .LONG 0 ; SUPERVISOR MODE, AND
00000000 07CC 865 .LONG 0 ; USER MODE
07D0 866
07D0 867 :*****
07D0 868 :
07D0 869 : END OF VECTOR PAGE
07D0 870 :
07D0 871 :*****
000007D0 07D0 872 CTLVECEND=
FFFFFDD0 07D0 873 .IF GREATER <CTLVECEND-CTLVECPAG>-512
07D0 874 .ERROR ; *** VECTOR PAGE NOW LARGER THAN A PAGE ***
07D0 875 .ENDC
07D0 876
07D0 877 .IFF ; ASSEMBLE REST ONLY FOR SHELL MODULE:
07D0 878

```

```

07D0 880 .SBTTL PROCESS I/O SEGMENT
07D0 881 -----
07D0 882 :
07D0 883 : PROCESS I/O SEGMENT INITIAL CONTENT
07D0 884 :
07D0 885 : -----
07D0 886
07D0 887
0800 888 PIO: .ALIGN PAGE ; PAGE BOUNDARY
0800 889 ; BASE OF PROCESS I/O SEGMENT
0800 890 PIO GL FMLH ; FREE MEMORY LIST HEAD
7FFE0200 7FFE0200 0800 891 .LONG PIO$GL_FMLH,PIO$GL_FMLH ; (Empty List)
0808 892
0808 893 PIO GL IIOFSPLH ; FREE LIST HEADER FOR IMAGE I/O SEG
7FFE0208 0808 894 .LONG PIO$GL_IIOFSPLH ;
7FFE0208 080C 895 .LONG PIO$GL_IIOFSPLH ;
0810 896
0810 897 PIO GW_STATUS ; RMS OVERALL STATUS
0000 0810 898 .WORD 0 ;
0812 899
0812 900
00000010 0812 901 PIO$$_EODSTR==16 ; SIZE OF STRING + COUNT BYTE
0812 902 PIO GT_ENDSTR ; END OF DATA STRING
00000822 0812 903 .BLKB PIO$$_EODSTR ; (COUNTED STRING)
0822 904
0822 905 PIO GW_DFPROT ; DEFAULT FILE PROTECTION
FA00 0822 906 .WORD ^XFA00 ; SYS:RWED, OWN:RWED, GROUP: RE, WORLD:N
0824 907
0824 908 PIO GB_DFMBBC ; DEFAULT MULTI-BLOCK COUNT
00000825 0824 909 .BLKB 1 ;
0825 910
0825 911 PIO GB_DFMBFSDK ; DEFAULT MULTI-BUFFER COUNT SEQ. DISK
00000826 0825 912 .BLKB 1 ;
0826 913
0826 914 PIO GB_DFMBFSMT ; DEFAULT MULTI-BUFFER COUNT MAGTAPE
0J000827 0826 915 .BLKB 1 ;
0827 916
0827 917 PIO GB_DFMBFSUR ; DEFAULT MULTI-BUFFER COUNT UNIT REC.
00000828 0827 918 .BLKB 1 ;
0828 919
0828 920 PIO GB_DFMBFREL ; DEFAULT MULTI-BUFFER COUNT RELATIVE
00000829 0828 921 .BLKB 1 ;
0829 922
0829 923 PIO GB_DFMBFIDX ; DEFAULT MULTI-BUFFER COUNT INDEXED
0000082A 0829 924 .BLKB 1 ;
082A 925
082A 926 PIO GB_DFMBFHSB ; DEFAULT MULTI-BUFFER COUNT HASHED
0000082B 082A 927 .BLKB 1 ;
082B 928
082B 929 PIO GB_DFNBC ; Network block count transfer size
0000082C 082B 930 .BLKB 1 ;
082C 931
082C 932 PIO GB_RMSPROLOG ; Structure level for RMS files
0000082D 082C 933 .BLKB 1 ;
082D 934
082D 935 PIO GW_RMSEXTEND ; Extend quantity for RMS files
0000082F 082D 936 .BLKW 1 ;

```

	082F	937			
	082F	938	.ALIGN	LONG	
	0830	939			
	0830	940	PIO	GL_DIRCACHE	: DIRECTORY CACHE LIST HEAD
7FFE0230	0830	941	.LONG	PIO\$GL_DIRCACHE	: EMPTY LIST
7FFE0230	0834	942	.LONG	PIO\$GL_DIRCACHE	
	0838	943			
	0838	944	PIO	GL_DIRCFLH	: FREE LIST FOR DIRECTORY CACHE NODES
00000000	0838	945	.LONG	0	: (NOTE: SINGLY LINKED)
	083C	946			
	083C	947	PIO	GL_RULOCK	: List of locks held for Recovery Units
00000000	083C	948	.LONG	0	: (Note: Singly Linked)
	0840	949			
	0840	950	PIO	GL_NXTIRBSEQ	: Next sequence number for IRB\$\$_IDENT
00000000	0840	951	.LONG	0	
	0844	952			

		0844	954
		0844	955
		0848	956
		0848	957
		0848	958
		0848	959
		0848	960
	0000	0848	961
	000D	084A	962
00000000	00000000	084C	963
		0854	964
		0854	965
7FFE0254	7FFE0254	0854	966
	0000860	085C	967
	7FFE0C00	0860	968
	7FFE0D00	0864	969
	0000003F	0868	970
		086C	971

.ALIGN QUAD

PROCESS I/O SEGMENT CONTEXT AREA

PIO	GW_PIOIMPA	:
.WORD	0	: FLAGS
.WORD	PRTSC_UREW	: I/O BUFFER PROTECTION
.LONG	0,0	: PIO SEGMENT, SET UP BY PROCSTR
PIOL	A1	: FREE PAGE LIST HEAD
.LONG	PIOLA1,PIOLA1	: FREE LIST HEADER
.BLKL	1	: SP SAVED LONGWORD
.LONG	PIO2A10	: IFAB TABLE ADDRESS
.LONG	PIO2A20	: IRAB TABLE ADDRESS
.LONG	IMPSC_NPIOFILES	: # OF SLOTS PER TABLE


```

086C 973 :
086C 974 :
086C 975 :
086C 976 :
0870 977 :
0001 0870 978 :
000D 0870 979 :
0872 980 :
0874 981 :
00000000 0874 982 :
0878 983 :
0878 984 :
0878 985 :
00000000 0878 986 :
0878 987 :
087C 988 :
087C 989 :
087C 990 :
087C 991 :
7FFE027C 7FFE027C 087C 992 :
087C 993 :
0884 994 :
0884 995 :
00000888 0884 996 :
0884 997 :
0888 998 :
7FFE0294 0888 999 :
7FFE02D4 088C 1000 :
0000000F 0890 1001 :
0894 1002 :
0894 1003 :
00000000 0894 1004 :
000008D4 0898 1005 :
08D4 1006 :
00000000 08D4 1007 :
08D4 1008 :
00000914 08D8 1009 :
0914 1010 :
0914 1011 :
0000091C 0914 1012 :
00000001 091C 1013 :
7FFE031C 0920 1014 :
0924 1015 :
00000000 00000000 0924 1016 :
0924 1017 :
092C 1018 :
092C 1019 :
092C 1020 :
092C 1021 :
092C 1022 :
092C 1023 :
092C 1024 :
092C 1025 :
7FFE032C 092C 1026 :
092C 1027 :
092C 1028 :
092C 1029 :

```

IMAGE I/O SEGMENT CONTEXT AREA
.ALIGN QUAD
PIO GW_IIOIMPA ; IMAGE I/O IMPURE AREA
.WORD 1 ; FLAGS WITH I/O SEGMENT SET
.WORD PRTSC_UREW ; PROTECTION STO SET ON PAGES
.LONG 0 ; MUST BE INITIALIZED BY THE IMAGE
; ACTIVATOR TO THE ADDRESS OF THE FIRST
; PAGE IN THE IMAGE I/O SEGMENT.
; (NOTE: THIS LONGWORD IS REFERENCED BY:
; PIO\$GW_IIOIMPA+IMP\$L_IOSEGADDR)
.LONG 0 ; LENGTH OF IMAGE I/O SEGMENT IN BYTES
; MUST BE INITIALIZED BY THE IMAGE ACTIVATOR
; (NOTE: THIS LONGWORD IS REFERENCED BY:
; PIO\$GW_IIOIMPA+IMP\$L_IOSEGLEN)
PIOL B1 ;
.LONG PIOLB1,PIOLB1 ; FREEPAGE LIST HEAD
PIOL B2 ;
.BLKL 1 ; SP SAVE LONGWORD
.LONG PIOLB10 ; IFAB TABLE ADDRESS
.LONG PIOLB20 ; IRAB TABLE ADDRESS
.LONG IMP\$C_ENTPERSEG ; # OF SLOTS PER TABLE
PIOL B10 ;
.LONG 0 ; LINK TO NEXT TABLE SEGMENT
.BLKL IMP\$C_ENTPERSEG ; IFAB TABLE SLOTS
PIOL B20 ;
.LONG 0 ; LINK TO NEXT TABLE SEGMENT
.BLKL IMP\$C_ENTPERSEG ; IRAB TABLE SLOTS
PIO AL_RMSEXH ; EXIT HANDLER CONTROL BLOCK
.BLKL 2 ;
.LONG 1 ; ARGUMENT COUNT
.LONG PIO\$AL_RMSEXH+8 ; STORE EXIT CODE OVER ARGUMENT COUNT
PIO GQ_IIODEFAULT ; DEFAULT IMAGE I/O AREA
.LONG 0,0
.ALIGN LONG

DEFAULT DIRECTORY INFORMATION
PIO GT_DDSTRING ; DEFAULT DIRECTORY STRING
7FFE032C 092C 1026 FIL\$GT_DDSTRING==PIO\$GT_DDSTRING ; FILE READ DEFAULT DIRECTORY STRING
092C 1027 :
092C 1028 : THE DEFAULT DIRECTORY STRING IN THE PROCESS QUOTA BLOCK MOVED FROM A
092C 1029 : CREATOR PROCESS TO A NEWLY CREATED PROCESS MUST BE AT LEAST AS LARGE

5D 45 58 45 53 59 53 5B 00
08

0000092C

```

092C 1030 ; AS THE LARGEST DIRECTORY STRING ALLOWED BY RMS.
092C 1031
092C 1032 ASSUME PQBSS_DDSTRING GE FWASC_MAXDIRLEN
092C 1033
092C 1034 TEMPS.. -.
092C 1035 .ASCIC \[SYSEXE]\ ; DEFAULT DIRECTORY STRING
092C
0935 1036
0935 1037 : ***** NOTE WELL *****
0935 1038 :
0935 1039 : THE DEFAULT DIRECTORY STRING MUST BE THE LAST ELEMENT TO APPEAR IN THE
0935 1040 : RMS CONTEXT AREA. IN FACT, IT HANGS OFF THE END OF THE PAGE. IF STORAGE
0935 1041 : WERE ALLOCATED, THE SHELL WOULD GROW IN SIZE BY ONE PAGE, AN UNNECESSARY
0935 1042 : WASTE OF SPACE. THE .BLKB DIRECTIVE HERE IS SYMBOLIC, TO EXPRESS THE SIZE
0935 1043 : OF THE DIRECTORY STRING IF SPACE WERE REALLY ALLOCATED.
0935 1044 :
0935 1045 : .BLKB PQBSS_DDSTRING-<.-TEMPS...> ; FILL TO MAXIMUM POSSIBLE SIZE
0935 1046 :
0935 1047 : ***** END OF NOTE *****

```

```

0935 1049 .SBTTL INITIALZE SHELL WITH SYSGEN PARAMETERS
0935 1050 :++
0935 1051 : FUNCTIONAL DESCRIPTION:
0935 1052 : SWP$SHELINIT IS CALLED BY THE SWAPPER FOLLOWING THE INSWAP OF A SHELL
0935 1053 : PROCESS TO APPLY THE SYSGEN PARAMETERS AND CONFIGURE THE PROCESS
0935 1054 : FOR THE PROPER WORKING SET SIZE AND VIRTUAL ADDRESS SPACE.
0935 1055 : THIS CODE IS CONTAINED IN PAGES THAT ARE BECOME THE KERNEL
0935 1056 : STACK FOR THE NEW PROCESS.
0935 1057 :
0935 1058 : CALLING SEQUENCE:
0935 1059 : JSB @#SWP$SHELINIT
0935 1060 :
0935 1061 : INPUT PARAMETERS:
0935 1062 : R4 - PCB ADDRESS
0935 1063 : R9 - SWAPPER MAP POINTER
0935 1064 : R10 - PTC$M VALID!PTESC ERKW!PTESM_MODIFY
0935 1065 : R11 - SWAPPER END MAP POINTER
0935 1066 :
0935 1067 : OUTPUT PARAMETERS:
0935 1068 : PHD AND CONTROL REGION FOR THE NEW PROCESS
0935 1069 :
0935 1070 :--
0935 1071 :
0935 1072 .ALIGN PAGE ; START OUT ON PAGE BOUNDARY
0A00 1073
0A00 1074 SWP$SHELINIT=-.SHELL ;
50 00000000'9F D0 0A00 1075 MOVL PCB$P PHD(R4),R5 ; GET PROCESS HEADER BASE ADDRESS
50 00000000'9F D0 0A04 1076 MOVL @#SWP$GL_SHELLSIZ,R0 ; GET PAGES ALLOCATED FOR SHELL
50 00000000'9F 02 78 0A0B 1077 MOVAL (R9)[R0],R11 ; COMPUTE END ADDRESS IN MAP
50 00000000'9F 02 78 0A0F 1078 ASHL #2,@#SWP$GL_SHELIO,R0 ; GET I/O SIZE OF SHELL
50 00000000'9F 51 5B 59 C3 0A17 1079 SUBL3 R9,R11,R1 ; COMPUTE SIZE OF EXTENSION PAGES
50 00000000'9F 51 51 50 C2 0A1B 1080 SUBL R0,R1 ; LESS I/O SIZE
50 00000000'9F 09 9C 0A22 1081 ROTL #3,R1,R1 ; CONVERT TO DOUBLE QUAD COUNT
50 00000000'9F 80 7C 0A2A 1082 ROTL #9,@#SWP$GL_SHELIO,R0 ; COMPUTE ADDRESS AT END OF I/O TRANSFER
50 00000000'9F 80 7C 0A2C 1083 10$: CLRQ (R0)+ ; CLEAR PAGES
50 00000000'9F 80 7C 0A2E 1084 10$: CLRQ (R0)+ ; NOT READ FROM SHELL
50 00000000'9F F9 51 F5 0A2E 1085 SOBGTR R1,10$ ;
50 00000000'9F 16 0A31 1086 MOVL R5,R2 ; VA OF PHD
50 00000000'9F 16 0A34 1087 JSB @#MMG$SVAPTECHK ; GET SVA OF FIRST PHD PTE
50 00000000'9F 83 89 D0 0A3A 1088 MOVL (R9)+(R3)+ ; MAP PROCESS HEADER FIXED PAGE
50 00000000'9F 3C 0A3D 1089 MOVZWL @#SWP$GW_WSLPTE,R1 ; GET COUNT OF ADDED PAGES FOR WSL+PST
50 00000000'9F 08 13 0A44 1090 BEQL 30$ ; BR IF NONE
50 00000000'9F 83 7B D0 0A46 1091 DECL R1 ; LESS FIXED HEADER PAGE
50 00000000'9F FA 51 F5 0A48 1092 20$: MOVL -(R11),(R3)+ ; MAP A WSL PAGE
50 00000000'9F 3C 0A4E 1093 SOBGTR R1,20$ ; DO THEM ALL
50 00000000'9F 53 6340 DE 0A55 1094 30$: MOVZWL @#SWP$GW_EMPTYPTTE,R0 ; GET COUNT OF EMPTY PAGES
50 00000000'9F 3C 0A59 1095 MOVAL (R3)[R0],R3 ; UPDATE MAP POINTER
50 00000000'9F 83 7B D0 0A60 1096 MOVZWL @#SWP$GW_BAKPTE,R0 ; GET COUNT OF BAK/WSL/VAL/LCK PTE
50 00000000'9F FA 50 F5 0A63 1097 40$: MO.L -(R11),(R3)+ ; MAP BAK/WSL/VAL/LCK PAGES
50 00000000'9F 0A66 1098 SOBGTR R0,40$ ;
50 00000000'9F 0A69 1099 INVALID ; INVALIDATE TRANSLATION BUFFER
50 00000000'9F 0A69 1100 :
50 00000000'9F 0A69 1101 : ALL OF THE HEADER PAGES LESS PAGE TABLES HAVE NOW BEEN MAPPED
50 00000000'9F 0A69 1102 :
50 00000000'9F 3C 0A69 1103 MOVZWL @#SWP$GW_IBALSETX,R8 ; GET BALANCE SLOT INDEX
50 00000000'9F 42 A5 58 B0 0A70 1104 MOVW R8,PHD$W-PHVINDEX(R5) ; SET INTO PROCESS HEADER
34 A5 00000000'9F 33 0A74 1105 CVTWB @#SGN$GW_DFPFC,PHD$B_DFPFC(R5) ; SET DEFAULT PAGE FAULT CLUSTER

```

```

53 00000000'9F 50  D4 0A7C 1106  CLRL  R0          : COUNT MAXIMUM FREE SPACE TILL NOW
      51 53   3C 0A7E 1107  MOVZWL @#MMG$GW_MINPFIDX,R3  : START SCAN AT FIRST PAGE FILE
56 00000000'9F 50  D0 0A85 1108  MOVL  R3,R1       : SET DEFAULT PAGING FILE TO CHOOSE
      51 53   D0 0A88 1109  MOVL  @#MMG$GL_PAGSWPVC,R6  : GET ADDRESS OF PAGE FILES VECTOR
      51 53   D0 0A8F 1110  :
      51 53   D0 0A8F 1111  :
      51 53   D0 0A8F 1112  :
      51 53   D0 0A8F 1113  :
      52 5A A4   3C 0A8F 1114  MOVZWL PCBSB_PGFLINDEX(R4),R2  : GET THE REQUESTED PAGE FILE INDEX
      51 52   13 0A93 1115  BEQL  44$         : NONE SPECIFIED
      51 52   D1 0A95 1116  CML  R2,R1        : CHECK FOR LEGAL INDEX
00000000'9F 17  1F 0A98 1117  BLSSU 44$         : A SWAP FILE WAS SPECIFIED - GIVE UP
      52 OE   1A 0AA1 1119  CML  R2,@#MMG$GL_MAXPFIDX  :
      58 6642   D0 0AA3 1120  BGTRU 44$         : TOO HIGH - GIVE UP
05 23 A8 00  E1 0AA7 1121  MOVL  (R6)[R2],R8         : GET PFL BLOCK ADDRESS
      51 52   D0 0AAC 1122  BBC  #PFL$V_INITED,PFL$B_FLAGS(R8),44$ ; BRANCH IF NOT USABLE
      52 6643   D0 0AB1 1124 44$: BRB  47$         : WE HAVE A GOOD SELECTION
      58 A4 01  88 0AB5 1125  MOVL  (R6)[R3],R2         : GET ADDRESS OF NEXT PAGE FILE BLOCK
58 23 A2 58 A4 89 0AB9 1126  BISB  #PFL$M_INITED,PCBSW_PGFLCHAR(R4); ADD INITED TO THE REQUESTED FLAGS
      23 A2 58  91 0ABF 1127  BISB3 PCBSW_PGFLCHAR(R4),PFL$B_FLAGS(R2),R8 ; OR THE FLAGS TOGETHER
      0D 12  12 0AC3 1128  CMPB  R8,PFL$B_FLAGS(R2)  : DID THEY CHANGE
      50 18 A2  D1 0AC5 1129  BNEQ  46$         : YES - NO MATCH - TRY AGAIN
      50 18 A2  D1 0AC9 1130  CML  PFL$B_FREPAGECNT(R2),R0  : CHOOSE PAGING FILE WITH MOST FREE PAGES
      51 53   15 0ACB 1131  BLEQ  46$         : BRANCH IF WE HAVE BETTER CANDIDATE
      51 53   D0 0ACB 1131  MOVL  PFL$B_FREPAGECNT(R2),R0  : SAVE NEW FREE PAGE COUNT
      51 53   D0 0ACF 1132  MOVL  R3,R1          : SAVE NEW INDEX
D7 53 00000000'9F 51  F3 0AD2 1133 46$: AOBLEQ @#MMG$GL_MAXPFIDX,R3,44$: LOOP THROUGH ALL PAGE FILES
      51 53   90 0ADA 1134 47$: MOV  R1,PHD$B_PAGFIL(R5)   : SET PAGING FILE INDEX
35 A5 00000000'9F 90  OADE 1135  MOV  @#$GN$GB_PGTBPFC,PHD$B_PGTPFC(R5); SET SYSTEM DEFAULT PT CLUSTER
      08 A5 0063 8F  B0 0AEE 1136  MOV  #SWP$C_KSTACK_WSL,PHD$B_WSLIST(R5); INIT POINTER TO LIST BASE
57 00000000'9F 50  D0 0AEC 1137  MOVL @#$GN$GL_PHD$PAGCT,R7   : GET TOTAL COUNT OF HEADER PAGES
56 00000000'9F 3C  0AF3 1138  MOVZWL @#$WPS$GW_WSLPTE,R6   : GET COUNT OF WSL HEADER PAGES
      20 A5 56 09  78 0AFA 1139  ASHL  #9,R6,PHD$B_PSTBASOFF(R5) ; SET END AS BASE FOR PST
56 00000000'9F A0 0AFF 1140  ADDW  @#$WPS$GW_EMPTPTE,R6   : ASSUMES NO OVERFLOW POSSIBLE
      56 56 07  78 0B06 1141  ASHL  #7,R6,R6          : CONVERT TO LONGWORD COUNT
      48 A5 56  D0 0B0A 1142  MOVL  R6,PHD$B_WSLX(R5)     : SET BASE OFFSET TO WSLX AREA
57 00000000'9F C0 0B0E 1143  ADDL  @#$GN$GL_PTPAGCNT,R7   : ADD PAGE TABLES TO COUNT
      50 57 01  C1 0B15 1144  ADDL3 #1,R7,R0          : ROUND TO LONGWORD SIZE
      50 02  C6 0B19 1145  DIVL  #2,R0             :
      56 50  C0 0B1C 1146  ADDL  R0,R6             : ALLOCATE SPACE IN WHOLE LONGWORDS
      44 A5 56  D0 0B1F 1147  MOVL  R6,PHD$B_BAK(R5)     : SET BASE OF BACKING STORE VECTOR
      56 57  C0 0B23 1148  ADDL  R7,R6             : ALLOCATE SPACE FOR BACKING STORE VECTOR
      56 04  C4 0B26 1149  MULL  #4,R6             : CONVERT TO BYTE OFFSET
      64 A5 56  D0 0B29 1150  MOVL  R6,PHD$B_PTW$SLELCK(R5) ; SET BASE OF LOCKED COUNT VECTOR
57 00000000'9F 50  D0 0B2D 1151  MOVL  @#$GN$GL_PTPAGCNT,R7   : GET COUNT OF PAGE TABLES
      68 A5 6647 9E 0B34 1152  MOVAB (R6)[R7],PHD$B_PTW$SLEVAL(R5) ; ALLOCATE AND SET BASE OF VAL CNT
      50 68 B545 9E 0B39 1153  MOVAB @PHD$B_PTW$SLEVAL(R5)[R5],R0 ; SET BASE FOR VALID COUNT VEC
      52 01  CE 0B3E 1154  MNEGL #1,R2             : MINUS ONE FOR BACKGROUND
      51 64 B545 9E 0B41 1155  MOVAB @PHD$B_PTW$SLELCK(R5)[R5],R1 ; AND BASE FOR LOCKED COUNT
      80 52  90 0B46 1156 50$: MOV  R2,(R0)+         : INIT BOTH COUNT VECTORS TO MINUS ONE
      81 52  90 0B49 1157  MOV  R2,(R1)+         :
      F7 57  F5 0B4C 1158  SOBGTR R7,50$       : FOR ALL PAGE TABLE SLOTS
      FE A1 04  80 0B4F 1159  ADD  #KSTACK+1,-2(R1)    : COUNT OF LOCK PAGES FOR SHELL
      FE A0 06  80 0B53 1160  ADD  #KSTACK+3,-2(R0)    : COUNT OF VALID PAGES
56 00000000'9F 50  D0 0B57 1161  MOVL @#$GN$GL_PHD$APCNT,R6  : GET ACTUAL HEADER SIZE
50 00000000'9F 9A 0B5E 1162  MOVZBL @#$SIP$GB_SHLP1PT,R0 : NUMBER OF PERM PAGE TABLES

```

SH
SY
EX
FI
FI
FW
FW
IA
IA
IA
IA
IA
IA
IA
IA
IA
IA
IA
IA
IA
IA
IM
IM
IM
IM
IP
JI
JI
JI
JI
KR
KS
KS
KS
MM
MM
MM
MM
MM
MM
MM
MM
MM
ND
NO
NS
NS
P1
P1
P1
PC
PC
PC
PC
PC
PC


```

30 A5 50 C2 0C42 1220      SUBL   R0,PHD$$_FREPIVA(R5) ; COMPUTE CORRECT FREPIVA
50 50 51 50 C2 0C46 1221      SUBL   R0,R1 ; EXTRACT PT SIZE TOTAL
50 50 F7 8F 78 0C49 1222      ASHL  #-9,R0,R0 ; BACK TO PAGE COUNT
51 00D4 C5 50 C2 0C4E 1223      SUBL   R0,PHD$$_P1LR(R5) ; NOW CORRECT P1 LENGTH REGISTER
51 51 FE 8F 78 0C53 1224      ASHL  #-2,R1,RT ; COUNT OF AVAILABLE PTES
2C A5 51 50 C2 0C58 1225      SUBL   R0,R1 ; DIMINISH BY PHD WINDOW SIZE
2C A5 51 C0 0C5B 1226      ADDL  R1,PHD$$_FREPTCNT(R5) ; AND FORM COUNT OF FREE PTES
0C5F 1227
0C5F 1228 ; THE FOLLOWING SHUFFLE OF THE WORKING SET LIST REARRANGES ALL WORKING SET
0C5F 1229 ; LIST ENTRIES THAT ARE NOT KERNEL STACK PAGES. THE SWAPPER MAP IS BEING
0C5F 1230 ; REORDERED TO AGREE WITH THE WORKING SET LIST DEFINED IN THE SHELL.
0C5F 1231
0C5F 1232      .REPEAT SHUFFLE COUNT ; COUNT FROM WSL TEMPLATE
0C5F 1233      MOVL  (R9)+,(R11)+ ; REARRANGE SWAP PAGE LIST
8B 89 D0 0C5F 1234      .ENDR
0C65 1235
30 A5 00000200 8B D4 0C65 1236      CLRL  (R11)+ ; SET STOPPER IN MAP
00000688 8F C1 0C67 1237      ADDL3 #512,PHD$$_FREPIVA(R5),- ; SET ADDRESS OF WINDOW
0C6F 1238      @#<SWP$$_PTRPAG+ ; INTO POINTER PAGE
00 24 A4 12 E2 0C74 1239      <CTL$$_PRD-CTL$$_VECTORS>> ; THROUGH SWAPPER MAP
50 08 A5 3C 0C79 1241 100$ : BBSS #PCB$$_PHDRES,PCB$$_STS(R4),100$ ; MARK PHD RESIDENT
50 50 D7 0C7D 1243      MOVZWL PHD$$_WSLIST(R5),R0 ; GET INDEX TO WS BASE
51 00000000'9F 50 A1 0C7F 1244      DECL  R0 ;
0A A5 51 B0 0C87 1245      ADDW3 R0,@#SGN$$_MAXWSCNT,R1 ;
18 A5 51 B0 0C8B 1246      MOVW  R1,PHD$$_WSAUTH(R5) ; CALC MAX AUTHORIZED
16 A5 51 B0 0C8F 1247      MOVW  R1,PHD$$_WSQUOTA(R5) ; SET MAX AUTHORIZED
14 A5 51 B0 0C93 1248      MOVW  R1,PHD$$_WSEXTENT(R5) ; AND QUOTA
74 A5 00000000'9F B0 0C97 1249      MOVW  R1,PHD$$_WSAUTHEXT(R5) ; AND EXTENT
51 00000000'9F B0 0C9F 1250      MOVW  @#SGN$$_MINWSCNT,PHD$$_WSFLUID(R5) ; AND AUTHORIZED EXTENT
50 51 A0 0CA6 1251      MOVW  @#SGN$$_DFWSCNT,R1 ; SET FLUID REQUIREMENT
12 A5 50 B0 0CA9 1252      ADDW  R1,R0 ; GET DEFAULT WS SIZE
1A A5 50 B0 0CAD 1253      MOVW  R0,PHD$$_WSLAST(R5) ; CALC LAST
50 A5 51 B0 0CB1 1254      MOVW  R0,PHD$$_DFWSCNT(R5) ; SET LAST
00C8 C5 55 C0 0CB5 1255      MOVW  R1,PHD$$_WSSIZE(R5) ; AND DEFAULT COUNT
00D0 C5 55 C0 0CBA 1256      ADDL  R5,PHD$$_POBR(R5) ; BIAS P0 BASE REGISTER
0CBF 1257 ; AND P1 BASE REGISTER ALSO
0CBF 1258 ;
0CBF 1259 ;
52 7FFEDE00 8F D0 0CBF 1260      MOVL  #P1SYSVECTORS,R2 ; PUT THEM HERE
00000000'9F 16 0CC6 1261      JSB  @#MMG$$_SVAPTECHK ; GET ADDRESS OF PTE FOR VECTORS
50 00000000'9F D0 0CCC 1262      MOVL  @#MMG$$_SPTBASE,R0 ; SYSTEM PAGE TABLE
83 80 00200000 8F 9A 0CD3 1263      MOVZBL #SGN$$_SYSVECPGS,R1 ; NUMBER OF PAGES
F5 51 F5 C9 0CD6 1264 110$ : BISL3 #PTESM-WINDOW,(R0)+,(R3)+
F5 51 F5 C9 0CDE 1265      SOBGR R1,110$
0CE1 1266 ;
0CE1 1267 ; THE PAGE TABLE ARRAY FOR LOCKED WSLE'S MUST BE INCREMENTED ONE
0CE1 1268 ; FOR EACH WINDOW PTE. THIS BALANCES THE LOGIC IN $CREPAG/$DELPAG
0CE1 1269 ; THAT PLACES THE PAGE TABLE PAGE IN THE LOCKED PORTION OF THE
0CE1 1270 ; WORKING SET FOR USER PFNMAPPED PAGES.
0CE1 1271 ;
0CE1 1272 ; THIS LOGIC ASSUMES THAT THE VECTOR PAGES ALL LIVE IN THE SAME
0CE1 1273 ; PAGE TABLE PAGE.
50 53 00C8 C5 C3 0CE1 1274      SUBL3 PHD$$_POBR(R5),R3,R0 ; BYTE OFFSET OF PTE
50 50 50 F7 8F 78 0CE7 1275      ASHL  #-9,R0,R0 ; BYTE INDEX OF CONTAINING PAGE TABLE

```

50	64	A5	C0	OCEC	1277	ADDL	PHD\$\$_PTWSLELCK(R5),R0 ; COMPUTE THE ADDRESS OF THE LOCKED
	50	55	C0	OCF0	1278	ADDL	R5,R0 ; WORKING SET LIST PAGE TABLE BYTE
	60	05	80	OCF3	1279	ADDB	#SGN\$C_SYSVECPGS,(R0) ; ARRAY AND INCR BY NUMBER OF VEC PAGES
				OCF6	1280		
				OCF6	1281		
				OCF6	1282		FINISH UP INITIALIZATION, SET DATA NEEDED BY SWAPPER
				OCF6	1283		
20	A4	20	A4	CE	OCF6	1284	MNEGL PCB\$\$_WSSWP(R4),PCB\$\$_WSSWP(R4)
					OCFB	1285	; INDICATE SHELL FINISHED TO SWAPPER
52	A5	5C	A4	F7	OCFB	1286	CVTLW PCB\$\$_SWAPSIZE(R4),PHD\$\$_SWAPSIZE(R5) ; SET INITIAL SWAP ALLOCATION
				OD00	1287		
				OD00	1288		; CLEAR THE PCB FIELDS THAT WERE USED TO CARRY PAGE FILE AND SWAP FILE
				OD00	1289		; INFORMATION TO THE NEW PROCESS. THESE FIELDS WILL BECOME THE COMMON EVENT
				OD00	1290		; FLAG CLUSTER POINTERS.
				OD00	1291		
				OD00	1292		ASSUME PCB\$\$_EFC3P EQ <PCB\$\$_EFC2P + 4>
				OD00	1293		ASSUME PCB\$\$_PGFLCHAR EQ PCB\$\$_EFC2P
				OD00	1294		ASSUME PCB\$\$_PGFLINDEX EQ <PCB\$\$_EFC2P + 2>
				OD0C	1295		ASSUME PCB\$\$_SWAPSIZE EQ <PCB\$\$_EFC2P + 4>
				OD00	1296		
	58	A4	7C	OD00	1297	CLRQ	PCB\$\$_EFC2P(R4) ; START WITH NO COMMON EF CLUSTERS
				OD03	1298		
			05	OD03	1299	RSB	; RETURN
				OD04	1300		

```

          ODO4 1302      .SBTTL FILLPHD - SETUP A VALID PHD PTE
          ODO4 1303      :
          ODO4 1304      :
          ODO4 1305      : R0 - WORKING SET LIST ENTRY, ADDRESS AND FLAGS (UPDATED)
          ODO4 1306      : R1 - COUNT OF PAGES TO FILL (UPDATED)
          ODO4 1307      : R2 - WORKING SET LIST INDEX (UPDATED)
          ODO4 1308      : R3 - SVAPTE FOR PHD PAGE (UPDATED)
          ODO4 1309      : R4 - PCB ADDRESS
          ODO4 1310      : R5 - PHD ADDRESS
          ODO4 1311      : R7 - SCRATCH
          ODO4 1312      : R8 - SCRATCH
          ODO4 1313      : R10 - PFNSC_ERKW!PFNSM_MODIFY!PFNSM_VALID
          ODO4 1314      :
          6542 50      DO ODO4 1315      FILLPHD:          ; SETUP VALID PHD PTES
          57 83 5A      CB ODO8 1316      10$:          MOVL R0,(R5)[R2]          ; STORE WORKING SET LIST ENTRY
          00000000'9F    DD ODOC 1317      BICL3 R10,(R3)+,R7          ; GET PFN FROM SPT
          9E47 87 8F    90 OD12 1318      PUSHL @#PFNSAB STATE          ; STACK ADDRESS BASE FOR PFN STATE
          00000000'9F    DD OD17 1319      MOVBL #<PFNSC_ACTIVE!PFNSM_MODIFY>,@(SP)+[R7] ; MARK PAGE ACTIVE
          9E47 04 90    90 OD1D 1320      PUSHL @#PFNSAB TYPE          ; STACK ADDRESS BASE FOR PFN TYPE
          00000000'9F    DD OD21 1321      MOVBL #PFNSC_PPGTBL,@(SP)+[R7] ; SET PAGE TYPE TO PAGE TABLE
          58 1F A5      9A OD27 1322      PUSHL @#PFNSAL BAK          ; STACK ADDRESS BASE FOR PFN BAK
          9E47 58 18    9C OD28 1323      MOVZBL PHDSB PAGFIL(R5),118          ; GET PAGE FILE NUMBER
          00000000'9F    DD OD30 1324      ROTL #PFNSV PGFLX,R8,@(SP)+[R7] ; SET BACKING STORE ADDRESS
          9E47 FC A3    DE OD36 1325      PUSHL @#PFNSAL PTE          ; STACK ADDRESS OF PFN PTE BACK POINTER
          ODO4 1326      :          MOVAL -4(R3),@(SP)+[R7]          ; SET PTE BACK POINTER FOR PAGE
          ODO4 1327      :
          ODO4 1328      : The following use of the PFN_REFERENCE macro must force absolute addressing
          ODO4 1329      : because of the peculiar method in which this code executes. In addition,
          ODO4 1330      : this code is not a part of the nonpaged executive and cannot have its opcode
          ODO4 1331      : automatically fixed up by INIT.
          00000000'9F    DD OD38 1332      PUSHL @#PFNSAx WSLX          ; STACK BASE ADDRESS OF WSLX VECTOR
          ODO4 1333      :          PFN_REFERENCE -
          ODO4 1334      :          MOVW <R2,@(SP)+[R7]>,-          ; SET WSLX FOR PAGE
          ODO4 1335      :          LONG_OPCODE=MOVZWL,-
          ODO4 1336      :          IMAGE=SHELL,-
          ODO4 1337      :          MODE=@#
          50 0200 52      D6 OD53 1338      INCL R2          ; FORCE POSITION INDEPENDENCE
          A7 51 51      DE OD55 1339      MOVAL 512(R0),R0          ; NEXT WORKING SET LIST ENTRY
          ODO4 1340      :          SOBGTR R1,10$          ; NEXT VA
          ODO4 1341      :          RSB          ; FILL ALL REQUESTED PAGES
          ODO4 1342      :          ; AND RETURN
          ODO4 1343      :
          00000007      OE00 1344      .ALIGN PAGE          ; PAGE ALIGN
          FFFFFFFF      OE00 1345      SWPSC_SHELLSIZ=<.-SHELL>@-9          ; SIZE OF SHELL PROCESS IN PAGES
          OE00 1346      .IF GT SWPSC_SHELLSIZ-8
          OE00 1347      .ERROR SWPSC_SHELLSIZ; Shell size changed, update swapper and FREELIM
          OE00 1348      .ENDC

```



```

OE00 1350
OE00 1351 :
OE00 1352 : Process IFI/ISI Tables
OE00 1353 :
OE00 1354 :
OE00 1355 ; The rest of this module merely defines global symbols and offsets into
OE00 1356 ; various P1 pages. No more storage is added to the SHELL module.
OE00 1357
OE00 1358 .PSECT $ABSS,ABS
0000 1359
7FFE0C00 0000 1360 .PIOTBL
OC00 1361 TBL:
OC00 1362 PIO2 A10 ; LINK TO NEXT TABLE SEGMENT
7FFF0C04 OC00 1363 .BLKL 1 ; INITIALLY ZERO
7FFE0D00 OC04 1364 .BLKL IMP$C_NPIOFILES ; IFAB TABLE SLOTS
OD00 1365
7FFE0D04 OD00 1366 PIO2 A20 ; LINK TO NEXT TABLE SEGMENT
7FFE0E00 OD00 1367 .BLKL 1 ; INITIALLY ZERO
OD04 1368 .BLKL IMP$C_NPIOFILES ; IRAB TABLE SLOTS
OE00 1369

```

```
0E00 1371 .SBTTL COMMAND LANGUAGE INTERPRETER DATA AREA
0E00 1372 :-----
0E00 1373 :      GENERIC COMMAND LANGUAGE INTERPRETER DATA
0E00 1374 :-----
0E00 1375
7FFE3000 0E00 1376      .=CLIDATAPAG      ; SET BASE VALUE FOR REGION
3000 1377 CTL$AL_CLICALBK:;      ; CALL BACK VECTOR FOR CLI
7FFE3004 3000 1378      .BLKL 1
7FFE3008 3004 1379      .BLKL 1      ; ALLOW FOR EXPANSION
3008 1380 CTL$AG_CLIMAGE:;
7FFE3010 3008 1381      .BLKL 2      ; VA RANGE INTO WHICH CLI IS MAPPED
3010 1382 CTL$AG_CLITABLE:;      ; VA RANGE INTO WHICH CLI TABLE IS
7FFE3018 3010 1383      .BLKL 2      ; MAPPED.
3018 1384
3018 1385 CTL$GL_UAF_FLAGS:;
7FFE301C 3018 1386      .BLKL 1      ; FLAGS FROM AUTHORIZATION RECORD
301C 1387
301C 1388 CTL$GT_CLINAME:;      ; CLI NAME (FILE NAME ONLY)
7FFE303C 301C 1389      .BLKB PQB$$_CLI_NAME      ; STORED AS COUNTED ASCII
303C 1390 CTL$GT_TABLENAME:;      ; CLI TABLE NAME (FULL FILE SPEC)
7FFE313C 303C 1391      .BLKB PQB$$_CLI_TABLE      ; STORED AS COUNTED ASCII
313C 1392 CTL$GT_SPAWNCLI:;      ; SPAWN CLI NAME (FILE NAME ONLY)
7FFE315C 313C 1393      .BLKB PQB$$_SPAWN_CLI      ; STORED AS COUNTED ASCII
315C 1394 CTL$GT_SPAWNTABLE:;      ; SPAWN CLI TABLE NAME (FULL FILE SPEC)
7FFE325C 315C 1395      .BLKB PQB$$_SPAWN_TABLE      ; STORED AS COUNTED ASCII
325C 1396
325C 1397 ASSUME <CTL$GT_TABLENAME -CTL$GT_CLINAME> EQ <PQB$$_CLI_TABLE -PQB$$_CLI_NAME>
325C 1398 ASSUME <CTL$GT_SPAWNCLI -CTL$GT_CLINAME> EQ <PQB$$_SPAWN_CLI -PQB$$_CLI_NAME>
325C 1399 ASSUME <CTL$GT_SPAWNTABLE-CTL$GT_CLINAME> EQ <PQB$$_SPAWN_TABLE-PQB$$_CLI_NAME>
325C 1400
000015A4 325C 1401 CTL$AG_CLIDATA:;      ; START OF DATA AREA
325C 1402 CTL$C_CLIDATASZ==CLIDATAEND-CTL$AG_CLIDATA      ; SIZE OF DATA AREA
325C 1403
```

```
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 1414 .SBTTL GLOBAL USER-MODE CONTEXT PAGE
1E28 1415 :-----
1E28 1416 : GLOBAL USER-MODE-ONLY CONTEXT PAGE
1E28 1417 :
1E28 1418 : This page may ONLY be used by user-mode code, since the data
1E28 1419 : has absolutely no protection from errant user-mode programs.
1E28 1420 : Do NOT use this area for cells which cannot be wiped out by
1E28 1421 : any user-mode program at any time during image execution,
1E28 1422 : except when it only prevents that program from running.
1E28 1423 :-----
7FFE2200 1E28 1424 . =UWVECPAG ; POINT TO START OF AREA
2200 1425
2200 1426 CTL$GL_DCLPRSOWN:: ; ADDRESS OF DCL WRK AREA CREATED
7FFE2204 2200 1427 .BLKL 1 ; BY DCL OR CLISDCL PARSE
2204 1428 CTL$GL_CLINTOWN:: ; ADDRESS OF CONTEXT BLOCK USED BY
7FFE2208 2204 1429 .BLKL 1 ; CLISINTERFACE PACKAGE IN DCL/MCR
2208 1430
2208 1431 ; THE REST OF THIS PAGE IS UNUSED
2208 1432
```

```

2208 147 .SBTTL IMAGE ACTIVATOR CONTEXT PAGE
2208 1435 -----
2208 1436 : IMAGE ACTIVATOR CONTEXT PAGE
2208 1437 :
2208 1438 : The following page contains image activator context that must
2208 1439 : remain behind after an image is activated. The IAFLIST survives
2208 1440 : across successive activations. The IAFEXE and IAFMERGE lists are
2208 1441 : cleared when a new image activation is initiated. The FIXUP link
2208 1442 : is cleared as soon as a given fixup pass completes.
2208 1443 :
2208 1444 :-----
7FFE2E00 2208 1445 . =IMGACTCTX ; Point to start of area
2E00 1446
7FFE2E04 2E00 1447 CTL$GL_IAFLINK:: ; Listhead of linked list of fixup vectors
2E00 1448 .BLKL 1 ; (containing shareable image lists)
2E04 1449
7FFE2E08 2E04 1450 CTL$GL_IAFLAST:: ; Address of last fixup vector in list
2E04 1451 .BLKL 1 ; (used in normal and merged image activation)
2E08 1452
7FFE2E0C 2E08 1453 CTL$GL_FIXUPLNK:: ; Listhead of linked list of fixup
2E08 1454 .BLKL 1 ; vectors used during fixup pass
2E0C 1455
7FFE2E10 2E0C 1456 CTL$GL_P1MERGE:: ; Listhead of linked list of fixup vectors
2E0C 1457 .BLKL 1 ; for images merged into P1 space
2E10 1458
2E10 1459 ; The following data structure is an empty fixup vector that facilitates
2E10 1460 ; the fixup vector list manipulation. It is plugged into the fixup vector
2E10 1461 ; lists by PROCSTRT when a process is first created.
2E10 1462
7FFE2E50 2E10 1463 CTL$GL_IAPERM::
2E10 1464 .BLKB IAF$K_LENGTH
2E50 1465
2E50 1466 ; The following label locates the link field in the just allocated structure.
2E50 1467 ; While an image is active, it locates the fixup vector for an executable image.
2E50 1468 ; It is cleared as part of the image activation initialization code path.
2E50 1469
7FFE2E10 2E50 1470 CTL$GL_IAFEXE == CTL$GL_IAPERM + IAF$L_IAFLINK
2E50 1471
7FFE2E54 2E50 1472 IAC$GL_IMAGCTX::
2E50 1473 .BLKL 1 ; Context that exists for life of image
2E54 1474
7FFE2E58 2E54 1475 IAC$GL_PROCCTX::
2E54 1476 .BLKL 1 ; Context that exists beyond image exit
2E58 1477
7FFE2E68 2E58 1478 IAC$AL_VECADDR::
2E58 1479 .BLKL 4 ; Array of altered opcode addresses
2E68 1480
7FFE2E6C 2E68 1481 IAC$AL_VECOPCOD::
2E68 1482 .BLKB 4 ; Array of saved opcodes
2E6C 1483
7FFE2E74 2E6C 1484 IAC$AW_VECRESET::
2E6C 1485 .BLKW 4 ; Array of offsets used to reset vectors
2E74 1486
7FFE2E7C 2E74 1487 IAC$AW_VECSET::
2E74 1488 .BLKW 4 ; Array of offsets used to locate vectors
2E7C 1489
2E7C 1490 ; The following linked list contains image control blocks for all of the

```

```
2E7C 1491 ; images currently mapped into a process' address space.
2E7C 1492
2E7C 1493 IAC$GL_IMAGE_LIST::
7FFE2E84 2E7C 1494 .BLKC 2 ; Room for forward and backward links
2E84 1495
2E84 1496 ; The following linked list is used by the image activator to record work
2E84 1497 ; in progress. It is empty while an image is executing.
2E84 1498
2E84 1499 IAC$GL_WORK_LIST::
7FFE2E8C 2E84 1500 .BLRL 2 ; Room for forward and backward links
2E8C 1501
2E8C 1502 ; The following list is a potential source of unused image control blocks.
2E8C 1503 ; Although empty when a process is created, it grows to reflect the largest
2E8C 1504 ; number of images activated at the same time.
2E8C 1505
2E8C 1506 IAC$GL_ICBFL::
7FFE2E94 2E8C 1507 .BLKL 2 ; Room for forward and backward links
2E94 1508
2E94 1509 ; The following two cells locate the ICB for the main image and the ICB for
2E94 1510 ; the image most recently merged into the address space.
2E94 1511
2E94 1512 IAC$GL_MAIN_ICB::
7FFE2E98 2E94 1513 .BLRL 1 ; ICB of main image
2E98 1514
2E98 1515 IAC$GL_FIRST_ICB::
7FFE2E9C 2E98 1516 .BLKC 1 ; ICB of image just merged
2E9C 1517
2E9C 1518 IAC$GL_STACK_SIZE::
7FFE2EA0 2E9C 1519 .BLKC 1 ; Amount by which to expand user stack
```

```
2EAO 1521
2EAO 1522           .IFT                               ; IF USRVECTOR MODULE:
2EAO 1523
2EAO 1524 :
2EAO 1525 : These symbols (SYSS...) specify which cells in the P1 pointer page may
2EAO 1526 : be used by any program, regardless of system version, without having
2EAO 1527 : to link with SYS.SIB (and thus, make it system dependant).
2EAO 1528 :
2EAO 1529 : These symbols will remain constant forever and will continue to be supported
2EAO 1530 : from release to release.
2EAO 1531 :
2EAO 1532
2EAO 1533           .MACRO USRSYM SYM,VALCHECK,CTLSYM
2EAO 1534           .IF B,CTLSYM
2EAO 1535 SYSS'SYM' == CTLS'SYM
2EAO 1536           .IF NB,VALCHECK
2EAO 1537           .IF NE,CTLS'SYM'^X'VALCHECK'
2EAO 1538           .ERROR ; *** CTLS'SYM' MUST EQ ^X'VALCHECK' TO REMAIN COMPATIBLE WITH PREVI
2EAO 1539           .ENDC
2EAO 1540           .ENDC
2EAO 1541           .IFF
2EAO 1542 SYSS'SYM' == CTLS'CTLSYM
2EAO 1543           .IF NB,VALCHECK
2EAO 1544           .IF NE,CTLS'CTLSYM'^X'VALCHECK'
2EAO 1545           .ERROR ; *** CTLS'CTLSYM' MUST EQ ^X'VALCHECK' TO REMAIN COMPATIBLE WITH PR
2EAO 1546           .ENDC
2EAO 1547           .ENDC
2EAO 1548           .ENDC
2EAO 1549           .ENDM
2EAO 1550
2EAO 1551           .DISABLE TRACEBACK
2EAO 1552
2EAO 1553 USRSYM GL_CMCNTX,7FFEFF88 : Address of 2 pages of AME storage
2EAO 1554 USRSYM GL_IAFLINK,7FFEFF8C,- : Points to IMGACT fixup listhead
2EAO 1555           GL_IAFLNKPTR
2EAO 1556 USRSYM GL_IMGLSTPTR,7FFEFEB4 : Points to ICM list (for debugger)
2EAO 1557
2EAO 1558           .ENDC : END OF SHELL / USRVECTOR CONDITIONAL
2EAO 1559
2EAO 1560           .END
```

SHELL
Symbol table

PROCESS DEFINITION

N 10

16-SEP-1984 01:11:32 VAX/VMS Macro V04-00
5-SEP-1984 03:47:44 [SYS.SRC]SHELL.MAR;1

Page 37
(1)

S
V

CLIDATAEND	= 7FFE4800	
CLIDATAPAG	= 7FFE3000	
COMMON	= 00000004	
CTLSAG_CLIDATA	7FFE325C	G
CTLSAG_CLIMAGE	7FFE3008	G
CTLSAG_CLITABLE	7FFE3010	G
CTLSAG_CMEDATA	= 7FFE1E00	G
CTLSAL_CLICALBK	7FFE3000	G
CTLSAL_CMCNTX	7FFE1E00	G
CTLSAL_FINALXEC	= 7FFEFF28	G
CTLSAL_IPASTVEC	= 7FFEFF68	G
CTLSAL_STACK	= 7FFEFE10	G
CTLSAL_STACKLIM	= 7FFEFE6C	G
CTLSAQ_EXCVEC	= 7FFEFE34	G
CTLSA_COMMON	= 7FFE1600	G
CTLSA_DISPVEC	= 7FFE6000	G
CTLSC_CLIDATASZ	= 000015A4	G
CTLSC_KRP_COUNT	= 00000004	G
CTLSC_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_CCBASE	= 7FFEFF38	G
CTL\$GL_CLINTOWN	7FFE2204	G
CTL\$GL_CMCNTX	= 7FFEFF88	G
CTL\$GL_CMHANDLR	= 7FFEFE30	G
CTL\$GL_CMSI'PR	= 7FFEFE28	G
CTL\$GL_CMUSER	= 7FFEFE2C	G
CTL\$GL_CREPRC_FLAGS	= 7FFEFFC0	G
CTL\$GL_CTLBASVA	= 7FFEFE7C	G
CTL\$GL_DCLPRSON	7FFE2200	G
CTL\$GL_F11BXQP	= 7FFEFF90	G
CTL\$GL_FINALSTS	= 7FFEFEB8	G
CTL\$GL_FIXUPLNK	7FFE2E08	G
CTL\$GL_GETMSG	= 7FFEFE68	G
CTL\$GL_IAFESE	= 7FFE2E10	G
CTL\$GL_IAFLAST	7FFE2E04	G
CTL\$GL_IAFLINK	7FFE2E00	G
CTL\$GL_IAFLNKPTR	= 7FFEFF8C	G
CTL\$GL_IAPERM	7FFE2E10	G
CTL\$GL_IBIOCNT	= 7FFEFEE8	G
CTL\$GL_ICPUTIM	= 7FFEFED0	G
CTL\$GL_IDIOCNT	= 7FFEFEE4	G
CTL\$GL_I'FAULTIO	= 7FFEFED8	G
CTL\$GL_I'FAULTS	= 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_KSTKBASXP	= 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_PPMMSG	= 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	= 7FFEFECB	G	
CTL\$GQ_ALLOCREG	= 7FFEFE8C	G	
CTL\$GQ_COMMON	= 7FFEFE60	G	
CTL\$GQ_DBGAREA	= 7FFEFF3C	G	
CTL\$GQ_HELPFLAGS	= 7FFEFFA8	G	
CTL\$GQ_I'START	= 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_PPMMSGCHN	= 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	
CTLVECE'ND	= 000007D0	R	03
CTLVECE'PAG	= 00000600	R	03
DBGAREA	= 7FFF0000		
DBGPTCNT	= 00000001		
DYNWSL	= 0000019C	R	03
END...	= 00000404	R	03
ESPINI	= 7FFE9E00		
ESTACK	= 00000010		
EXE\$EXCPTN	*****	X	03

SHELL
Symbol table

PROCESS DEFINITION

B 11

16-SEP-1984 01:11:32 VAX/VMS Macro V04-00
5-SEP-1984 03:47:44 [SYS.SRC]SHELL.MAR;1

Page 38
(1)

SH
VO

EXE\$EXECPTNE	*****	X	03
EXE\$PROCSTR	*****	X	03
FIL\$GT_DDSTRING	= 7FFE032C	R	03
FILLPHD	= 00000D04		
FWASC_MAXDIRLEN	= 000000FF		
FWSL...	= 000001A0	R	03
IACSAL_IMGACTBUF	= 7FFE4800	G	
IACSAL_VECADDR	= 7FFE2E58	G	
IACSAL_VECOPCOD	= 7FFE2E68	G	
IACSAL_VECRESET	= 7FFE2E6C	G	
IACSAL_VECSET	= 7FFE2E74	G	
IAC\$GL_FIRST_ICB	= 7FFE2E98	G	
IAC\$GL_CBFL	= 7FFE2E8C	G	
IAC\$GL_IMGACTX	= 7FFE2E50	G	
IAC\$GL_IMAGE_LIST	= 7FFE2E7C	G	
IAC\$GL_MAIN_ICB	= 7FFE2E94	G	
IAC\$GL_PROCTX	= 7FFE2E54	G	
IAC\$GL_STACK_SIZE	= 7FFE2E9C	G	
IAC\$GL_WORK_LIST	= 7FFE2E84	G	
IAFSK_LENTR	= 00000040		
IAFSL_IAFLINK	= 00000000		
IMGACTBUF	= 00000008		
IMGACTCTX	= 7FFE2E00		
IMPSC_ENTPERSEG	= 0000000F		
IMPSC_NPIOFILES	= 0000003F		
IPLS_A\$TDEL	= 00000002		
JIB\$S_ACCOUNT	= 00000008		
JIB\$S_USERNAME	= 0000000C		
JIB\$T_ACCOUNT	= 00000018		
JIB\$T_USERNAME	= 0000000C		
KRP_COUNT	= 00000004		
KSPINI	= 7FFE7E00		
KSTACK	= 00000003		
KSTACK_EX	= 00000004		
MMG\$GL_MAXPFIDX	*****	X	03
MMG\$GL_PAGSWPVC	*****	X	03
MMG\$GL_SPTBASE	*****	X	03
MMG\$GW_BIGPFN	*****	X	03
MMG\$GW_MINPFIDX	*****	X	03
MMG\$IMGHDRBUF	= 7FFE6400	G	
MMG\$SVAPTECHK	*****	X	03
NDYN	= 00000001		
NOT_KSTACK_WSL	= 00000198	R	03
NSA\$K_IDT_PAGES	= 00000003		
NSA\$T_IDT	= 7FFE2800	G	
P1PTBAS	= 00000200	R	03
P1PTLEN	= 000000FF		
P1SYSVECTORS	= 7FFEDE00	G	
PCB\$B_PGFLINDEX	= 0000005A		
PCB\$B_EFC2P	= 00000058		
PCB\$B_EFC3P	= 0000005C		
PCB\$B_PHD	= 0000006C		
PCB\$B_STS	= 00000024		
PCB\$B_SWAPSIZE	= 0000005C		
PCB\$B_WSSWP	= 00000020		
PCB\$V_PHDRES	= 00000012		
PCB\$W_PGFLCHAR	= 00000058		

PFILPGCNT	= 00000068		
PFL\$B_FLAGS	= 00000023		
PFL\$B_FREPAGECNT	= 00000018		
PFL\$M_INITED	= 00000001		
PFL\$V_INITED	= 00000000		
PFNSAB_STATE	*****	X	03
PFNSAB_TYPE	*****	X	03
PFNSAL_BAK	*****	X	03
PFNSAL_PTE	*****	X	03
PFNSAX_WSLX	*****	X	03
PFNSC_ACTIVE	= 00000007		
PFNSC_PPGTBL	= 00000004		
PFNSM_MODIFY	= 00000080		
PFNSV_PGFLX	= 00000018		
PHD\$B_CPUMODE	= 00000060		
PHD\$B_DFPFC	= 00000034		
PHD\$B_PAGFIL	= 0000001F		
PHD\$B_PGTBPFC	= 00000035		
PHD\$C_LENGTH	= 0000017C		
PHD\$D_BAK	= 00000044		
PHD\$D_FREP1VA	= 00000030		
PHD\$D_FREPTCNT	= 0000002C		
PHD\$D_KSP	= 00000078		
PHD\$D_POBR	= 000000C8		
PHD\$D_POLRASTL	= 000000CC		
PHD\$D_P1BR	= 000000D0		
PHD\$D_P1LR	= 000000D4		
PHD\$D_PC	= 000000C0		
PHD\$D_PSL	= 000000C4		
PHD\$D_PSTBASOFF	= 00000020		
PHD\$D_PTWSLELCK	= 00000064		
PHD\$D_PTWSLEVAL	= 00000068		
PHD\$D_WSLX	= 00000048		
PHD\$Q_PRIVMSK	= 00000000		
PHD\$V_ASTLVL	= 00000018		
PHD\$W_DFWSCNT	= 0000001A		
PHD\$W_PHVINDEXT	= 00000042		
PHD\$W_PRCLM	= 0000003E		
PHD\$W_PTCNTACT	= 00000070		
PHD\$W_PTCNTLCK	= 0000006C		
PHD\$W_PTCNTMAX	= 00000072		
PHD\$W_PTCNTVAL	= 0000006E		
PHD\$W_QUANT	= 0000003C		
PHD\$W_SWAPSIZE	= 00000052		
PHD\$W_WSAUTH	= 0000000A		
PHD\$W_WSAUTHEXT	= 00000014		
PHD\$W_WSDYN	= 0000000E		
PHD\$W_WSEXTENT	= 00000016		
PHD\$W_WSFLUID	= 00000074		
PHD\$W_WSLAST	= 00000012		
PHD\$W_WSLIST	= 00000008		
PHD\$W_WSLOCK	= 0000000C		
PHD\$W_WSNEXT	= 00000010		
PHD\$W_WSQUOTA	= 00000018		
PHD\$W_WSSIZE	= 00000050		
PHD...	= 00000000	R	03
PIO	= 00000800	R	03

SHELL
Symbol table

PROCESS DEFINITION

C 11

16-SEP-1984 01:11:32 VAX/VMS Macro V04-00
5-SEP-1984 03:47:44 [SYS.SRC]SHELL.MAR;1

Page 39
(1)

SH
VO

```

PIOSAL_RMSEXH      = 7FFE0314 G
PIOSA_DIRCACHE    = 7FFE0800 G
PIOSA_TRACE       = 7FFE0600 G
PIOSGB_DFMBC      = 7FFE0224 G
PIOSGB_DFMBFHSH   = 7FFE022A G
PIOSGB_DFMBFIDX   = 7FFE0229 G
PIOSGB_DFMBFREL   = 7FFE0228 G
PIOSGB_DFMBFSDK   = 7FFE0225 G
PIOSGB_DFMBF SMT  = 7FFE0226 G
PIOSGB_DFMBF SUR  = 7FFE0227 G
PIOSGB_DFNBC      = 7FFE022B G
PIOSGB_RMSPROLOG  = 7FFE022C G
PIOSGL_DIRCACHE   = 7FFE0230 G
PIOSGL_DIRCRLH    = 7FFE0238 G
PIOSGL_FMLH       = 7FFE0200 G
PIOSGL_IIOF SPLH  = 7FFE0208 G
PIOSGL_NXTIRBSEQ  = 7FFE0240 G
PIOSGL_RULOCK     = 7FFE023C G
PIOSGQ_IIODEFAULT = 7FFE0324 G
PIOSGT_DDSTRING   = 7FFE032C G
PIOSGT_ENDSTR     = 7FFE0212 G
PIOSGW_DFPROT     = 7FFE0222 G
PIOSGW_IIOIMPA    = 7FFE0270 G
PIOSGW_PIOIMPA    = 7FFE0248 G
PIOSGW_RMSEXTEND  = 7FFE022D G
PIOSGW_STATUS     = 7FFE0210 G
PIOSS_EODSTR      = 00000010 G
PIO2A10           = 7FFE0C00
PIO2A20           = 7FFE0D00
PIOBASE          = 7FFE0200
PIOLA1           = 7FFE0254
PIOLB1           = 7FFE027C
PIOLB10          = 7FFE0294
PIOLB2           = 7FFE0284
PIOLB20          = 7FFE02D4
PIOTBL           = 7FFE0C00
PQBSS_CLI_NAME   = 00000020
PQBSS_CLI_TABLE  = 00000100
PQBSS_DDSTRING   = 00000100
PQBSS_SPAWN_CLI  = 00000020
PQBSS_SPAWN_TABLE = 00000100
PQBST_CLI_NAME   = 00000088
PQBST_CLI_TABLE  = 000000A8
PQBST_SPAWN_CLI  = 000001A8
PQBST_SPAWN_TABLE = 000001C8
PRS_TBIA         = ***** X 03
PRTSC_UREW       = 0000000D
PSLSC_USER       = 00000003
PSLSV_IPL        = 00000010
PTESC_EOWN       = 00800000
PTESC_ERKW       = 30000000
PTESC_KOWN       = 00000000
PTESC_KW         = 10000000
PTESC_NA         = 00000000
PTESC_SOWN       = 01000000
PTESC_SREW       = 48000000
PTESC_SRKW       = 50000000

```

```

PTESC_UOWN       = 01800000
PTESC_UREW       = 68000000
PTESC_URKW       = 70000000
PTESC_URSW       = 60000000
PTESC_UW         = 20000000
PTESM_TYP1       = 04000000
PTESM_WINDOW     = 00200000
SAV...           = 00000600 R 03
SGNSC_SYSVECPGS  = 00000005
SGNSGB_PGTBPFC   = ***** X 03
SGNSGL_MAXWSCNT  = ***** X 03
SGNSGL_PHDAPCNT  = ***** X 03
SGNSGL_PHDPAGCT  = ***** X 03
SGNSGL_PTPAGCNT  = ***** X 03
SGNSGW_DFPFC     = ***** X 03
SGNSGW_DFWSCNT   = ***** X 03
SGNSGW_MINWSCNT  = ***** X 03
SHELL            = 00000000 R 03
SHUFFLE_COUNT    = 00000002
SSPINI           = 7FFEDE00
SSTACK          = 00000020
SWPSAL_PTRPAG    = 00000600 G
SWPSC_DBGPTCNT   = 00000001 G
SWPSC_KSTACK     = 00000003 G
SWPSC_KSTACK_EX  = 00000004 G
SWPSC_KSTACK_EX_WSL = 0000005F G
SWPSC_KSTACK_WSL = 00000063 G
SWPSC_NDYN       = 00000001 G
SWPSC_SHELLPFIL  = 00000068 G
SWPSC_SHELLSIZ   = 00000007
SWPSC_SHLFPT     = 00000081 G
SWPSC_SHLP1PT    = 00000002 G
SWPSGB_SHLP1PT   = ***** X 03
SWPSGL_SHELIO    = 00000000 RG 02
SWPSGL_SHELLBAS  = 00000000 RG 03
SWPSGL_SHELLSIZ  = ***** X 03
SWPSGW_BAKPTE    = ***** X 03
SWPSGW_EMPTPTE   = ***** X 03
SWPSGW_IBALSETX  = ***** X 03
SWPSGW_WSLPTE    = ***** X 03
SWPSSHELINIT     = 00000A00 G
TBL              = 7FFE0C00
TEMPS...         = 0000092C R 03
TMP              = 00000400
TMP...           = 04000000
USPINI           = 7FFE0200
UWVECPAG         = 7FFE2200
VASH_SYSTEM      = 80000000
VA...           = 7FFE0200
VECTORS          = 7FFEFE00
WSLSC_PPGTBL     = 00000008
WSLSC_PROCESS    = 00000000
WSLSM_MODIFY     = 00000100
WSLSM_VALID      = 00000001
WSLSM_WSLOCK     = 00000020
WSL...           = 0000018C R 03

```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	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

