



```

SSSSSSSS YY YY SSSSSSS DDDDDDD GGGGGGG BBBB8888 LL SSSSSSS CCCCCCCC
SSSSSSSS YY YY SSSSSSS DDDDDDD GGGGGGG BBBB8888 LL SSSSSSS CCCCCCCC
SS SS YY YY SS SSSSSSS DD DD GG BB BB LL SS CC
SS SS YY YY SS SSSSSSS DD DD GG BB BB LL SS CC
SS SS YY YY SS SSSSSSS DD DD GG BB BB LL SS CC
SSSSSS YY YY SSSSSSS DD DD GG GGGGGG BB BB LL SS CC
SSSSSS YY YY SSSSSSS DD DD GG GGGGGG BB BB LL SS CC
SS SS YY YY SS SSSSSSS DD DD GG GGGGGG BB BB LL SS CC
SSSSSS YY YY SSSSSSS DDDDDDD GGGGGG BBBB8888 LLLLLLLLLL SSSSSSS CCCCCCCC
SSSSSS YY YY SSSSSSS DDDDDDD GGGGGG BBBB8888 LLLLLLLLLL SSSSSSS CCCCCCCC

```

```

LL LL I I I I I I SSSSSSS
LL LL I I I I I I SSSSSSS
LL LL I I I I I I SS
LL LL I I I I I I SS
LL LL I I I I I I SS
LL LL I I I I I I SSSSSS
LL LL I I I I I I SSSSSS
LL LL I I I I I I SS
LL LL I I I I I I SS
LL LL I I I I I I SS
LLLLLLLLLLLL I I I I I I SSSSSSS
LLLLLLLLLLLL I I I I I I SSSSSSS

```

SYS DGBLSC  
Table of contents

(2)	74
(3)	116
(4)	244
(5)	300
(7)	508
(9)	694
(10)	869

DECLARATIONS  
VFYSECFLG - VERIFY SECTION FLAGS  
GSDMTXULK - UNLOCK GLOBAL SECTION MUTEX  
DGBLSC - DELETE GLOBAL SECTION SYSTEM SERVICE  
GSDSCN - GLOBAL SECTION DESCRIPTOR SCANNER  
DELGBLSEC - DELETE GLOBAL SECTION SUBROUTINE  
DELGBLWCB - DELETE GLOBAL WINDOW CONTROL BLOCKS

```

0000 1 .TITLE SYSDGBLSC - DELETE GLOBAL SECTION SYSTEM SERVICE
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 9 * ALL RIGHTS RESERVED. *
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 16 * TRANSFERRED. *
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 20 * CORPORATION. *
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 24
0000 25
0000 26 *****
0000 27
0000 28
0000 29 **
0000 30 FACILITY:
0000 31
0000 32 ABSTRACT:
0000 33
0000 34
0000 35 ENVIRONMENT:
0000 36
0000 37 AUTHOR: PETER H. LIPMAN, CREATION DATE: 9-MAY-77
0000 38
0000 39 MODIFIED BY:
0000 40
0000 41 V03-010 TMK0001 Todd M. Katz 28-Apr-1984
0000 42 Modify MMG$GSDSCN so that the global section descriptor
0000 43 name maybe specified as a full length logical name
0000 44 ( ie - 0 < LOGNAM size =< LNMSC_NAMELENGTH ).
0000 45
0000 46 V03-009 WMC0004 Wayne Cardoza 28-Mar-1984
0000 47 Use hash byte to check for GSD match.
0000 48 Make sure file system is present before deallocating WCB.
0000 49
0000 50 V03-008 WMC0003 Wayne Cardoza 14-Mar-1984
0000 51 Fix deletion of resident section.
0000 52
0000 53 V03-007 WMC0002 Wayne Cardoza 24-Feb-1984
0000 54 Add resident global sections.
0000 55
0000 56 V03-006 MSH0004 Michael S. Harvey 26-Jan-1984
0000 57 Add support for lengthened global section name in GSDs.

```

0000	58	:			
0000	59	:	V03-005	WMC0001	Wayne Cardoza
0000	60	:		Add new flag -	SECSM_EXECUTE
0000	61	:			
0000	62	:	V03-004	LJK0199	Lawrence J. Kenah
0000	63	:		Do not charge	FILCNT quota for shared windows.
0000	64	:			
0000	65	:	V03-003	LJK0203	Lawrence J. Kenah
0000	66	:		Fix broken word	displacements.
0000	67	:			
0000	68	:	V03-002	KDM0037	Kathleen D. Morse
0000	69	:		Do not allow the	SECSV PAGFIL option for PFNMAP sections.
0000	70	:		Turn on the	DZRO and WRT bits for PAGFIL sections.
0000	71	:			
0000	72	:	--		

```
0000 74      .SBTTL  DECLARATIONS
0000 75      :
0000 76      : INCLUDE FILES:
0000 77      :
0000 78      $CCBDEF      :CHANNEL CONTROL BLOCK DEFINITIONS
0000 79      $DYNDEF      :DYNAMIC STRUCTURE TYPES
0000 80      $GSDDEF      :GLOBAL SECTION DESCRIPTOR DEFINITIONS
0000 81      $IPLDEF      :PROCESSOR PRIORITY LEVELS
0000 82      $JIBDEF      :JOB INFORMATION BLOCK DEFINITIONS
0000 83      $LNMDEF      :LOGICAL NAME STRING DEFINITIONS
0000 84      $MMGDEF      :OFFSETS FROM FP INTO SCRATCH AREA
0000 85      $PCBDEF      :PROCESS CONTROL BLOCK DEFINITIONS
0000 86      $PHDDEF      :PROCESS HEADER DEFINITIONS
0000 87      $PFNDEF      :PFN DATA BASE DEFINITIONS
0000 88      $PRDEF       :PROCESSOR REGISTER DEFINITIONS
0000 89      $PRVDEF      :PRIVILEGE BIT DEFINITIONS
0000 90      $PSLDEF      :PROCESSOR STATUS LONGWORD DEFINITIONS
0000 91      $PTEDEF      :PAGE TABLE ENTRY DEFINITIONS
0000 92      $SECDEF      :SECTION TABLE DEFINITIONS
0000 93      $SHBDEF      :SHARED MEMORY CONTROL BLOCK DEF
0000 94      $SSDEF       :SYSTEM STATUS DEFINITIONS
0000 95      $UCBDEF      :UNIT CONTROL BLOCK DEFINITIONS
0000 96      $WCBDEF      :WINDOW CONTROL BLOCK DEFINITIONS
0000 97      :
0000 98      : MACROS:
0000 99      :
0000 100     :
0000 101     : EQUATED SYMBOLS:
0000 102     :
0000 103     :
0000 104     :
0000 105     : OFFSETS FROM AP
0000 106     :
0000 107     $OFFSET 4,POSITIVE,<-
0000 108     FLAGS,-
0000 109     GSDNAM,-
0000 110     IDENT,-
0000 111     >
0004     FLAGS:
0008     GSDNAM:
000C     IDENT:
0000 112     :
0000 113     : OWN STORAGE:
0000 114     :
```

```

0000 116      .SBTTL VFYSECFLG - VERIFY SECTION FLAGS
0000 117      :++
0000 118      : FUNCTIONAL DESCRIPTION:
0000 119      :
0000 120      :     THIS ROUTINE VERIFIES THAT THE SECTION FLAGS CONTAIN ONLY
0000 121      :     THOSE FLAG BITS DEFINED AT THE USER INTERFACE
0000 122      :
0000 123      : CALLING SEQUENCE:
0000 124      :
0000 125      :     BSBW      MMG$VFYSECFLG
0000 126      :
0000 127      : INPUT PARAMETERS:
0000 128      :
0000 129      :     RO = SECTION FLAGS
0000 130      :
0000 131      : IMPLICIT INPUTS:
0000 132      :
0000 133      :     NONE
0000 134      :
0000 135      : OUTPUT PARAMETERS:
0000 136      :
0000 137      :     IF SECTION FLAGS ARE NOT LEGAL, THEN THIS ROUTINE EXITS WITH RET
0000 138      :     OTHERWISE IT RSB'S TO ITS CALLER WITH:
0000 139      :
0000 140      :     RO PRESERVED
0000 141      :
0000 142      : IMPLICIT OUTPUTS:
0000 143      :
0000 144      :     NONE
0000 145      :
0000 146      : COMPLETION CODES:
0000 147      :
0000 148      :     SSS_IVSECFLG      ;INVALID SECTION FLAGS
0000 149      :
0000 150      : SIDE EFFECTS:
0000 151      :
0000 152      :     NONE
0000 153      :
0000 154      : --
0000 155      :
0000 156      :
0000 157      : *****
0000 158      :
0000 159      : ***** THE FOLLOWING CODE MAY BE PAGED *****
0000 160      :
0000 161      :     .PSECT Y$EXEPAGED
0000 162      :
0000 163      : *****
0000 164      :
0000 165      MMG$VFYSECFLG::
50  FFE01F30 8F  D3 0000 166      BITL      #^C<SECSM_GBL ! -      ;GLOBAL/PROCESS SECTION
0007 167      SECSM_CRF ! -      ;COPY ON REFERENCE
0007 168      SECSM_DZRO ! -      ;DEMAND ZERO
0007 169      SECSM_WRT ! -      ;WRITABLE
0007 170      SECSM_SYSGBL ! -      ;SYSTEM/GROUP GLOBAL
0007 171      SECSM_PFNMAP ! -      ;MAP TO SPECIFIC PFN'S
0007 172      SECSM_EXPREG ! -      ;MAP TO FIRST FREE VA SPACE

```

```

0007 173 SEC$M_PROTECT ! - ;PROTECTED SECTION
0007 174 SEC$M_PAGFIL ! - ;PAGE FILE BACKING STORE
0007 175 SEC$M_WRTMOD ! - ;WRITE ACCESS MODE
0007 176 SEC$M_EXECUTE ! - ;CHECK FOR EXECUTE ACCESS
0007 177 SEC$M_RESIDENT ! - ;RESIDENT GLOBAL SECTION
0007 178 SEC$M_PERM >,RO ;PERMANENT/TEMPORARY
56 12 0007 179 BNEQ 20$ ;BRANCH IF ANY BAD BITS
0009 180
41 50 10 E0 0009 181 BBS #SEC$V_PFNMAP,RO,18$ ;BR TO DO FURTHER CHECKS IF PFNMAP
54 50 0D E0 000D 182 BBS #SEC$V_RESIDENT,RO,50$ ;RESIDENT CHECKS
0011 183
0011 184 ;
0011 185 ; CHECKS FOR SEC$V_PAGFIL OPTION:
0011 186 ;
08 50 13 E1 0011 187 BBC #SEC$V_PAGFIL,RO,5$
46 50 01 E0 0015 188 BBS #SEC$V_CRF,RO,20$ ;CRF AND PFNMAP NOT ALLOWED FOR PAGFIL
42 50 00 E1 0019 189 BBC #SEC$V_GBL,RO,20$ ;MUST BE GBL
50 0C C8 001D 190 BLSL2 #<SEC$M_DZRO!SEC$M_WRT>,RO ;FORCE WRT/DZRO FOR PAGFIL OPTION
0020 191
0020 192 ;
0020 193 ; CHECKS FOR SEC$V_DZRO OPTION:
0020 194 ;
0C 50 02 E1 0020 195 5$: BBC #SEC$V_DZRO,RO,10$ ;IS THIS A DEMAND-ZERO SECTION?
50 0A D3 0024 196 BITL #<SEC$M_CRF!SEC$M_WRT>,RO ;IF YES, THEN IT MUST BE CRF OR
36 13 0027 197 BEQL 20$ ;WRT TOO, ELSE ERROR
0C29 198 ASSUME <SEC$M_GBL!SEC$M_CRF!SEC$M_DZRO> EQ 7
07 50 03 00 0029 199 CMPZV #0,#3,RO,#<SEC$M_GBL!SEC$M_CRF!SEC$M_DZRO> ;GBL, CRF, DZRO
2F 13 002E 200 BEQL 20$ ;SECTIONS ARE PROHIBITED
0030 201
0030 202 ;
0030 203 ; COMPUTE WRITE ACCESS MODE:
0030 204 ;
50 50 DD 0030 205 10$: PUSHL RO ;SAVE FLAGS
08 50 03 9A 0032 206 MOVZBL #PSL$C_USER,RO ;ASSUME WRITE ACCESS MODE NOT SPECIFIED
50 6E 02 12 E1 0035 207 BBC #SEC$V_PROTECT,(SP),15$ ;BR IF WRITE ACCESS MODE NOT SPECIFIED
6E 02 06 EF 0039 208 EXTZV #SEC$V_WRTMOD,#SEC$S_WRTMOD,(SP),RO ;GET WRT ACMODE SPECIFIED
FFBF' 30 003E 209 BSBW EX$MAXACMODE ;MAXIMIZE WRT MODE WITH PREV MODE
6E 02 06 50 F0 0041 210 15$: INSV RO,#SEC$V_WRTMOD,#SEC$S_WRTMOD,(SP) ;PUT MAX WRT MODE IN FLAGS
50 8E D0 0046 211 MOVL (SP)+,RO ;SET VERIFIED FLAGS AND WRT MODE
EB AD 50 D0 0049 212 MOVL RO,MMG$SL_VFYFLAGS(FP) ;REMEMBER VERIFIED FLAGS FOR LATER USE
05 004D 213 RSB ;OK, RETURN TO CALLER
004E 214
004E 215 ;
004E 216 ; CHECKS FOR SEC$V_PFNMAP OPTION:
004E 217 ;
50 00080006 8F D3 004E 218 18$: BITL #<SEC$M_CRF!SEC$M_DZRO!SEC$M_PAGFIL>,RO ;PFNMAP CANNOT BE CRF,
08 12 0055 219 BNEQ 20$ ;DZRO, OR PAGFIL. BR IF FLAGS ILLEGAL.
0057 220 ;
0057 221 ; THIS CODE REQUIRES SEC$V_PERM TO BE SET FOR ALL PFNMAP GLOBAL SECTIONS.
0057 222 ; TEMPORARY PFNMAP GLOBAL SECTIONS NEVER GO AWAY BECAUSE THE DELAYED
0057 223 ; DELETION MECHANISM FOR GLOBAL SECTIONS REQUIRES THE SECTION TO HAVE
0057 224 ; A SECTION TABLE ENTRY. PFNMAP SECTIONS DO NOT HAVE SECTION TABLE
0057 225 ; ENTRIES BECAUSE THEY ARE NOT ASSOCIATED WITH A FILE. THUS, THE
0057 226 ; ROUTINE MMG$DALCSTXSCN NEVER FINDS PFNMAP GLOBAL SECTIONS.
0057 227 ;
D5 50 00 E1 0057 228 BBC #SEC$V_GBL,RO,10$ ;BR IF PROCESS PFNMAP SECTION
D1 50 0E E0 005B 229 BBS #SEC$V_PERM,RO,10$ ;BR ON FLAGS OK, GBL PERM PFNMAP SEC

```



```

50 016C 8F 3C 005F 230 20$: MOVZWL #SS$_IVSECFLG,R0 ;INVALID SECTION FLAGS
      04 0064 231 RET ;RETURN FROM SYSTEM SERVICE
      0065 232
      0065 233
      0065 234
      0065 235 ; CHECKS FOR RESIDENT GLOBAL SECTIONS
      0065 236
0009000E 8F D3 0065 237 50$: BITL #<SEC$_CRF!SEC$_WRT!SEC$_DZRO!SEC$_PAGFIL!SEC$_PFNMAP>,-
      50 50 006B 238 R0 ;CHECK FOR ILLEGAL COMBINATIONS
      F1 12 006C 239 BNEQ 20$
ED 50 00 E1 006E 240 BBC #SEC$_GBL,R0,20$ ;MUST BE GLOBAL
      BC 11 0072 241 BRB 10$ ;FINISH UP
      0074 242

```

```

0074 244 .SBTTL GSDMTXULK - UNLOCK GLOBAL SECTION MUTEX
0074 245 :++
0074 246 : FUNCTIONAL DESCRIPTION:
0074 247 :
0074 248 : THIS ROUTINE UNLOCKS THE GLOBAL SECTION MUTEX
0074 249 :
0074 250 : CALLING SEQUENCE:
0074 251 :
0074 252 : BSBW MMG$GSDMTXULK
0074 253 :
0074 254 : INPUT PARAMETERS:
0074 255 :
0074 256 : NONE
0074 257 :
0074 258 : IMPLICIT INPUTS:
0074 259 :
0074 260 : SCH$GL_CURPCB = PROCESS CONTROL BLOCK OWNING THE MUTEX
0074 261 : EXE$GL_GSDMTX = MUTEX TO UNLOCK
0074 262 :
0074 263 : OUTPUT PARAMETERS:
0074 264 :
0074 265 : R4 = CURRENT PCB ADDRESS
0074 266 :
0074 267 : IMPLICIT OUTPUTS:
0074 268 :
0074 269 : NONE
0074 270 :
0074 271 : COMPLETION CODES:
0074 272 :
0074 273 : NONE
0074 274 :
0074 275 : SIDE EFFECTS:
0074 276 :
0074 277 : NONE
0074 278 :
0074 279 : --
0074 280 :
0074 281 :
0074 282 : THE FOLLOWING ROUTINE IS INVOKED BY THE SWAPPER THROUGH CHKDGBLSC.
0074 283 : IT MAY LATER BE REQUIRED TO REMAIN RESIDENT.
0074 284 :
0074 285 :
0074 286 : *****
0074 287 :
0074 288 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0074 289 :
00000074 290 : .PSECT Y$EXEPAGED
0074 291 :
0074 292 : *****
0074 293 :
0074 294 MMG$GSDMTXULK::
54 00000000'EF D0 0074 295 MOVL L^SCH$GL_CURPCB,R4 ;PROCESS CONTROL BLOCK ADDRESS
50 00000000'EF DE 007B 296 MOVAL L^EXE$GL_GSDMTX,R0 ;GLOBAL SECTION MUTEX
00000000'EF 16 0082 297 JSB SCH$UNLOCK ;UNLOCK THE MUTEX
05 0088 298 RSB ;AND RETURN

```

```

0089 300      .SBTTL DGBLSC - DELETE GLOBAL SECTION SYSTEM SERVICE
0089 301      :++
0089 302      : FUNCTIONAL DESCRIPTION:
0089 303      :
0089 304      :
0089 305      : CALLING SEQUENCE:
0089 306      :
0089 307      :     CALLG  ARGLIST,@#SYSS$DGBLSC
0089 308      :     CALLG  ARGLIST,MMG$DGBLSC1
0089 309      :
0089 310      : INPUT PARAMETERS:
0089 311      :
0089 312      :     FLAGS(AP) = BIT 0 - GBL - GLOBAL IF SET, PROCESS IF CLEAR
0089 313      :                BIT 1 - CRF - COPY ON REFERENCE
0089 314      :                BIT 2 - DZRO - DEMAND ZERO
0089 315      :                BIT 3 - WRT - WRITABLE IF SET, READ ONLY IF CLEAR
0089 316      :                BITS 4 - 13 RESERVED, MUST BE ZERO
0089 317      :                BIT 14 - PERM - PERMANENT IF SET, TEMPORARY IF CLEAR
0089 318      :                BIT 15 - SYSGBL - SYSTEM GLOBAL IF SET, GROUP GLOBAL IF CLEAR
0089 319      :                BIT 16 - PFNMAP - MAP TO SPECIFIC PFN'S
0089 320      :                BIT 17 - EXPREG, MAP TO FIRST FREE SPACE AVAILABLE
0089 321      :                        (UNUSED FOR $DGBLSC)
0089 322      :                BITS 18 - 31 RESERVED, MUST BE ZERO
0089 323      :                ***** NOTE THAT ONLY SYSGBL AND PFNMAP ARE REQUIRED FOR
0089 324      :                ***** THIS SERVICE.
0089 325      :     GSDNAM(AP) = ADDRESS OF DESCRIPTOR FOR GLOBAL SECTION NAME
0089 326      :     IDENT(AP)  = ADDRESS OF QUAD WORD CONTAINING SECTION IDENTIFICATION
0089 327      :                FIRST LONG WORD CONTAINS THE MATCH CONTROL INFORMATION
0089 328      :                0 = SEC$K_MATALL, MATCH ALWAYS
0089 329      :                1 = SEC$K_MATEQU, MATCH IF IDENT'S ARE EQUAL
0089 330      :                2 = SEC$K_MATLEQ, MATCH IF HIGH 8 BITS ARE EQUAL
0089 331      :                AND LOW 24 BITS ARE LESS THAN OR EQUAL TO
0089 332      :                THE ID STORED IN THE GLOBAL SECTION.
0089 333      :                SECOND LONG WORD CONTAINS THE IDENT TO BE COMPARED
0089 334      :     R4          = PROCESS CONTROL BLOCK ADDRESS
0089 335      :                SET UP BY THE CHANGE MODE TO KERNEL DISPATCHER
0089 336      :     IPL = ASTDEL IF CALLED AT MMG$DGBLSC1
0089 337      :
0089 338      : IMPLICIT INPUTS:
0089 339      :
0089 340      :     NONE
0089 341      :
0089 342      : OUTPUT PARAMETERS:
0089 343      :
0089 344      :     RO = SYSTEM STATUS CODE
0089 345      :     IPL PRESERVED WHEN CALLED AT MMG$DGBLSC1
0089 346      :
0089 347      : IMPLICIT OUTPUTS:
0089 348      :
0089 349      :     NONE
0089 350      :
0089 351      : COMPLETION CODES:
0089 352      :
0089 353      :     $$$_IVSECFLG - INVALID SECTION FLAGS
0089 354      :     $$$_NOPRIV  - NO PRIVILEGE FOR ATTEMPTED OPERATION (PRMGBL)
0089 355      :     $$$_NORMAL  - SUCCESS
0089 356      :     $$$_NOTCREATOR - NOT ON GLOBAL SECTION'S CREATOR PROCESSOR

```

```
0089 357 : SSS_IVLOGNAM - INVALID LOGICAL NAME
0089 358 : SSS_NOSUCHSEC - NO SUCH GLOBAL SECTION
0089 359 : SSS_IVSECIDCTL - INVALID SECTION MATCH IDENT CONTROL
0089 360 : SSS_ACCVIO - ACCESS VIOLATION
0089 361 : SSS_TOOMANYLNAM - TOO MANY LOGICAL NAMES (DEPTH > 10)
0089 362 : SSS_SHMNOTCNCT - SHARED MEM DATA STRUCTURES NOT CONNECTED TO LOCAL MEM
0089 363 :
0089 364 : SIDE EFFECTS:
0089 365 :
0089 366 : NONE
0089 367 :
0089 368 :--
```



```

00E7 427 ; R6 = GLOBAL SECTION DESCRIPTOR BLOCK ADDRESS
00E7 428 ; R10 IS 0 IF THE GSD WAS FOUND IN LOCAL MEMORY
00E7 429 ; -1 IF THE LOCAL MEMORY SEARCH EXTENDED INTO SHARED MEMORY TABLES
00E7 430 ; >1 IF A SPECIFIC SHARED MEMORY NAME WAS SPECIFIED
00E7 431 ;
51 16 A6 32 00E7 432 25$: CVTWL GSD$W_GSTX(R6),R1 ;SECTION INDEX
09 13 00EB 433 BEQL 30$ ;BR IF GS NOT MAPPED TO A FILE
53 55 20 A5 C1 00ED 434 ADDL3 PHD$L PSTBASOFF(R5),R5,R3 ;SECTION TABLE BASE ADDRESS
53 6341 DE 0CF2 435 MOVAL (R3)[R1],R3 ;SECTION ADDRESS
28 0A A6 91 00F6 436 30$: CMPB GSD$B_TYPE(R6),#DYN$C_EXTGSD ;IS THIS A NORMAL GSD?
44 1A 00FA 437 BGTRU SHMEM_DEL ;NO, BR AS IT IS A SHM GSD
72 13 00FC 438 BEQL NOT_MAP_TO_FILE ;NO, BR AS IT IS AN EXTENDED GSD
50 66 OF 00FE 439 REMQUE (R6),R0 ;REMOVE GSD FROM ITS LIST
00000000'EF 60 OE 0101 440 INSQUE (R0),L^EXE$GL GSDDELFL ;AND PLACE IT ON THE DELETE PENDING LIST
00 20 A6 0E E5 0108 441 BBCC #SEC$V_PERM,GSD$W_FLAGS(R6),40$ ;CLEAR PERM, ALLOW DELETION
51 D5 010D 442 40$: TSTL R1 ;IS THERE A SECTION TABLE ENTRY?
OF 13 010F 443 BEQL 50$ ;BR IF NO, NOT MAPPED TO A FILE
00 14 A3 0E E5 0111 444 BBCC #SEC$V_PERM,SEC$W_FLAGS(R3),45$ ;CLEAR PERM, ALLOW DELETION
18 A3 D5 0116 445 45$: TSTL SEC$L_REFCNT(R3) ;IF NO MORE REFERENCES
05 12 0119 446 BNEQ 50$ ;INDICATE THAT THERE IS A SECTION
00 36 A5 01 E6 011B 447 BBSSI #PHD$V_DALCSTX,PHD$W_FLAGS(R5),50$ ;TO BE DEALLOCATED
01 DD 0120 448 50$: PUSHL #SS$_NORMAL ;RECORD SUCCESS
0122 449 ;
0122 450 ; 0(SP) = SYSTEM STATUS CODE
0122 451 ; 4(SP) = SAVED CALLER' IPL
0122 452 ;
29 0A A6 91 0122 453 55$: CMPB GSD$B_TYPE(R6),#DYN$C_SHMGSD ;IS THERE A SHMGSD TO UNLOCK?
09 12 0126 454 BNEQ 60$ ;BR IF OTHER TYPE OF GSD
50 01 9A 0128 455 MOVZBL #1,R0 ;SET REFERENCE COUNT
00000000'GF 16 012B 456 JSB G^MMG$DEC$SHMREF ;RELEASE GSD SO CAN DISAPPEAR
FECC 30 0131 457 60$: BSBW MMG$DALCSTXSCN ;SCAN FOR SECTIONS TO DEALLOCATE
FF3D 30 0134 458 BSBW MMG$GSDMTXULK ;UNLOCK THE GLOBAL SECTION MUTEX
0186 30 0137 459 BSBW MMG$DELGBLWCB ;DELETE ANY WCB'S ON WCBDEL QUEUE
01 BA 013A 460 POPR #^M<R0> ;RESTORE SYSTEM STATUS CODE
04 013C 461 ENBINT ;RESTORE CALLER'S IPL
0140 462 RET ;AND RETURN FROM SYSTEM SERVICE
0140 463 ;
0140 464 ;
0140 465 ; SECTION BEING DELETED IS IN SHARED MEMORY. SEE IF PROCESS REQUESTING
0140 466 ; DELETION IS ON SAME PROCESSOR AS PROCESS THAT CREATED THE SECTION.
0140 467 ;
0140 468 SHMEM_DEL:
59 10 58 E8 0140 469 BLBS R8,65$ ;BR IF IMAGE ACT. REQ., NO PRIV CHK
00000000'9F DO 0143 470 MOVL @#CTL$GL PHD,R9 ;GET PHD OF CURRENT PROCESS
05 69 1B E0 014A 471 BBS #PRV$V_SHMEM,PHD$Q_PRIVMSK(R9),65$ ;BR IF HAVE PRIV FOR REQ
7E 24 3C 014E 472 MOVZWL #SS$_NOPRIV,-(SP) ;REPORT NO PRIVILEGE FOR REQUEST
CF 11 0151 473 BRB 55$ ;BR ON NO PRIV TO DO REQUEST
52 A6 95 0153 474 65$: TSTB GSD$B_CREATPORT(R6) ;NON-EXISTENT CREATOR?
07 19 0156 475 BLSS 67$ ;BRANCH IF NON-EXISTENT TO ALLOW DELETION
52 A6 15 A4 91 0158 476 CMPB SHB$B_PORT(R4),GSD$B_CREATPORT(R6) ;IS THIS PROC THE CREATOR?
06 12 015D 477 BNEQ 70$ ;BR IF IT IS NOT, CANNOT DELETE SECTION
AA 66 02 E6 015F 478 67$: BBSSI #GSD$V_DELPEND,GSD$L_GSDFL(R6),40$ ;MARK THE GSD FOR DELETION
A8 11 0163 479 BRB 40$ ;BRANCH TO COMMON CODE
7E 0384 8F 3C 0165 480 70$: MOVZWL #SS$_NOTCREATOR,-(SP) ;REPORT UNABLE TO DELETE SHM GLOB SEC
B6 11 016A 481 BRB 55$ ;BR TO REPORT ERROR CODE
016C 482 ;
016C 483 ; ERROR FROM THE GLOBAL SECTION DESCRIPTOR LIST SCAN

```

```

50 DD 016C 484 :
C1 11 016C 485 80$: PUSHL R0 ;SAVE ERROR STATUS
016E 486 BRB 60$ ;EXIT THROUGH THE UNLOCK CODE
0170 487 :
0170 488 : THIS IS AN EXTENDED GSD. IT DOES NOT HAVE A CORRESPONDING SECTION TABLE
0170 489 : ENTRY, I.E., A PFN-MAPPED SECTION. THERE IS NO REFERENCE COUNT KEPT FOR
0170 490 : A PFN-MAPPED SECTION, REQUIRING A DELAYED DELETION MECHANISM.
0170 491 : SINCE THE GSD MUTEX IS BEING HELD FOR WRITE DURING THE $DGBLSC REQUEST,
0170 492 : THERE IS NO RISK THAT ANY OTHER PROCESS IS ACCESSING THE GSD.
0170 493 : THEREFORE, IT CAN BE DEALLOCATED TO PAGED POOL IMMEDIATELY.
0170 494 :
0170 495 NOT_MAP_TO_FILE:
59 10 58 E8 0170 496 BLBS R8,85$ ;BR IF IMAGE ACT. REQ., NO PRIV CHK
00000000'9F D0 0173 497 MOVL @#CTL$GL PHD,R9 ;GET PHD OF CURRENT PROCESS
05 69 1A E0 017A 498 BBS #PRV$V PFNMAP,PHD$Q_PRIVMSK(R9),85$ ;BR IF HAVE PRIV FOR REQ
50 24 3C 017E 499 MOVZWL #SS$_NOPRIV,R0 ;REPORT NO PRIVILEGE FOR REQUEST
E9 11 0181 500 BRB 80$ ;BR ON NO PRIV TO DO REQUEST
50 66 OF 0183 501 85$: REMQUE (R6),R0 ;REMOVE EXTENDED GSD FROM ACTIVE LIST
00000000'EF 16 0186 502 JSB L^EXE$DEAPAGED ;DEALLOCATE EXTGSD TO PAGED POOL
01 DD 018C 503 PUSHL #SS$_NORMAL ;RECORD SUCCESS
A1 11 018E 504 BRB 60$ ; AND JOIN COMMON CODE
0190 505
0190 506 .DSABL LSB

```

S  
V  
  
P  
-  
I  
C  
P  
S  
P  
S  
P  
C  
A  
  
T  
1  
T  
9  
3  
  
M  
-  
-  
T  
1  
T  
M

```

0190 508 .SBTTL GSDSCN - GLOBAL SECTION DESCRIPTOR SCANNER
0190 509 :++
0190 510 : FUNCTIONAL DESCRIPTION:
0190 511 :
0190 512 :     THIS ROUTINE SCANS THE GLOBAL SECTION DESCRIPTOR QUEUE FOR
0190 513 :     A SECTION WITH THE SPECIFIED NAME.
0190 514 :
0190 515 : CALLING SEQUENCE:
0190 516 :
0190 517 :     BSBW    MMG$GSDSCN
0190 518 :
0190 519 : INPUT PARAMETERS:
0190 520 :
0190 521 :     R0 = DESCRIPTOR ADDRESS OF GLOBAL SECTION NAME
0190 522 :     R1 = ADDRESS OF QUAD WORD OF SECTION IDENT INFORMATION
0190 523 :     R6 = SECTION FLAGS
0190 524 :           IF SYSGBL IS SET SEARCH SYSTEM GLOBAL LIST
0190 525 :           OTHERWISE SEARCH GROUP GLOBAL LIST
0190 526 :     R7 = ADDRESS OF SCRATCH AREA TO STORE THE GLOBAL SECTION NAME IN
0190 527 :           COUNTED STRING FORMAT. ALSO INCLUDES SCRATCH SPACE FOR
0190 528 :           GLOBAL SECTION IDENT.
0190 529 :     GSD MUTEX IS LOCKED (FOR READ OR WRITE AS APPROPRIATE) BEFORE CALLING
0190 530 :     IPL = ASTDEL
0190 531 :
0190 532 : IMPLICIT INPUTS:
0190 533 :
0190 534 :     NONE
0190 535 :
0190 536 : OUTPUT PARAMETERS:
0190 537 :
0190 538 :     R0 = SYSTEM STATUS CODE
0190 539 :     R4 = ADDRESS OF SHARED MEMORY CONTROL BLOCK, IF GSD IS IN SHARED MEMORY
0190 540 :     R5 = SYSTEM PROCESS HEADER ADDRESS
0190 541 :     R6 = GLOBAL SECTION DESCRIPTOR BLOCK ADDRESS IF FOUND
0190 542 :     R10 = 0 IF GSD IS IN LOCAL MEMORY,
0190 543 :           -1 IF LOCAL MEMORY SEARCH EXTENDED INTO SHARED MEMORY TABLES,
0190 544 :           > 0 IF IN SHARED MEMORY
0190 545 :
0190 546 : IMPLICIT OUTPUTS:
0190 547 :
0190 548 :     GLOBAL SECTION NAME (ASCIC) IS RETURNED IN THE SCRATCH AREA.
0190 549 :
0190 550 : COMPLETION CODES:
0190 551 :
0190 552 :     SSS_NORMAL                ;SUCCESSFUL COMPLETION
0190 553 :     SSS_IVLOGNAM              ;INVALID NAME
0190 554 :     SSS_IVSECIDCTL           ;INVALID SECTION IDENT MATCH CONTROL
0190 555 :     SSS_NOSUCHSEC            ;NO SUCH (GLOBAL) SECTION
0190 556 :     SSS_ACCVIO                ;ACCESS VIOLATION
0190 557 :
0190 558 : SIDE EFFECTS:
0190 559 :
0190 560 :     NONE
0190 561 :
0190 562 :--

```



```

0190 564 :
0190 565 : *****
0190 566 :
0190 567 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0190 568 :
00000190 569 : .PSECT YSEXEPAGED
0190 570 :
0190 571 : *****
0190 572 :
0190 573 :
0190 574 : .ENABL LSB
0190 575 :
0190 576 BADNAME:
50 0154 8F 3C 0190 577 MOVZWL #SS$ IVLOGNAM,R0 ;SET ERROR STATUS
0A 11 0195 578 BRB ERREXIT ;
0197 579 BADIDENT:
50 02E4 8F 3C 0197 580 MOVZWL #SS$ IVSECIDCTL,R0 ;INVALID SECTION ID MATCH CONTROL
03 11 019C 581 BRB ERREXIT ;
019E 582 ACCVIO:
50 0C 9A 019E 583 MOVZBL #SS$_ACCVIO,R0 ;REPORT ACCESS VIOLATION
01A1 584 ERREXIT:
5A D4 01A1 585 CLRL R10 ;INDICATE LOCAL MEMORY ON ERROR
0100 31 01A3 586 BRW 35$ ;
01A6 587
01A6 588 MMG$GSDSCN:
0B00 8F BB 01A6 589 PUSHR #*M<R8,R9,R11> ;REMEMBER REGISTERS
52 7C 01AA 590 CLRQ R2 ;ASSUME NULL IDENT IFORMATION
51 D5 01AC 591 TSTL R1 ;0 ADDRESS MEANS NO IDENT TO MATCH
0E 13 01AE 592 BEQL 5$ ;BRANCH IF NO IDENT
01B0 593 IFNORD #8,(R1),ACCVIO ;BRANCH IF CALLER CANNOT READ THE QUAD WORD
52 61 7D 01B6 594 MOVQ (R1),R2 ;GET THE IDENT INFORMATION
02 52 D1 01B9 595 CMPL R2,#SECSK_MATLEQ ;MATCH CONTROL OK?
D9 14 01BC 596 BGTR BADIDENT ;BRANCH IF BAD
2C A7 52 7D 01BE 597 5$: MOVQ R2,44(R7) ;STORE IDENT INFO IN SCRATCH AREA
01C2 598 IFNORD #8,(R0),ACCVIO ;CHECK ACCESSABILITY OF DESCRIPTOR
59 60 3C 01C8 599 MOVZWL (R0),R9 ;GET SIZE OF STRING
C3 13 01CB 600 BEQL BADNAME ;BR IF NO STRING SPECIFIED
5A 04 A0 D0 01CD 601 MOVL 4(R0),R10 ;GET ADR OF INPUT GS STRING
01D1 602 ASSUME LNMSC,NAMLENGTH,LE,512
59 00FF 8F B1 01D1 603 CMPW #LNMSC,NAMLENGTH,R9 ;IS NAME LONGER THAN MAXIMUM ALLOWED?
B8 1F 01D6 604 BLSSU BADNAME ;BR IF ILLEGAL NAME SIZE
01D8 605 IFNORD R9,(R10),ACCVIO ;CHK IF STRING ACCESSABLE
7E 59 7D 01DE 606 MOVQ R9,-(SP) ;MOVE INPUT GS DESC TO STACK
59 5E D0 01E1 607 MOVL SP,R9 ;GET ADDR
7E 01 A7 9E 01E4 608 MOVAB 1(R7),-(SP) ;SET ADR OF GS BUFFER IN STR DESC
2B DD 01E8 609 PUSHL #<11*4>-1 ;SET SIZE OF GS BUFFER IN STR DESC
5B 5E D0 01EA 610 MOVL SP,R11 ;SET ADR OF GS DESC
5E 10 C2 01ED 611 SUBL #<4*4>,SP ;GRAB SHARED MEMORY NAME BUFFER
7E 01 AE 9E 01F0 612 MOVAB 1(SP),-(SP) ;SET ADR OF SH MEM BUFFER IN STR DESC
OF DD 01F4 613 PUSHL #15 ;SET SIZE OF SH MEM BUFFER IN STR DESC
5A 5E D0 01F6 614 MOVL SP,R10 ;SET ADR OF SH MEM DESC
FE04 30 01F9 615 BSBW MMG$GETGSNAM ;GET GS AND SH MEM NAMES
05 50 E8 01FC 616 BLBS R0,7$ ;BR ON NAMES CORRECT
6A D4 01FF 617 CLRL (R10) ;INDICATE LOCAL MEMORY
009C 31 0201 618 BRW ERROR_EXIT ;
04 04 AA D7 0204 619 7$: DECL 4(R10) ;ADR OF ASCII SH MEM NAME
BA 6A 90 0207 620 MOVB (R10),24(R10) ;MAKE SH MEM NAME INTO ASCII STRING

```

```

      67  6B  90  020B  621  MOVB  (R11),(R7)      ;MAKE GS NAME INTO ASCIC STRING
      03  12  020E  622  BNEQ  8$             ;IF NEQ THEN GS NAME SPECIFIED
      009F 31  0210  623  BRW    NO_GS_NAME      ;ERROR, NO GS NAME SPECIFIED
56  56  58  00  D2  0213  624  8$:  MCOML  #0,R8-      ;NEGATIVE ONE FOR SYSTEM GLOBAL
      01  0F  EF  0216  625  EXTZV  #SEC$V_SYSGBL,#1,R6,R6 ;SYSTEM GLOBAL BIT
      0C  12  021B  626  BNEQ  10$             ;BRANCH IF SYSTEM GLOBAL
      021D  627  ;
      021D  628  ; GROUP GLOBAL SECTION SCAN - GET GROUP CODE FROM PCB
      021D  629  ;
58  00000000'EF D0  021D  630  MOVL  L^SCH$GL_CURPCB,R8 ;PROCESS CONTROL BLOCK ADDRESS
      58  00BE C8 3C  0224  631  MOVZWL PCB$W_GRP(R8),R8 ;GROUP CODE TO MATCH
      0229  632  ;
      0229  633  ; COMPUTE A SIMPLE HASH CODE FOR THE GSD NAME
      0229  634  ;
      50  59  D4  0229  635  10$: CLRL  R9             ;HASH ACCUMULATOR
      51  57  D0  022B  636  MOVL  R7,R0          ;STRING POINTER FOR HASH CALCULATION
      51  80  9A  022E  637  MOVZBL (R0)+,R1        ;LENGTH
      59  80  80  0231  638  15$:  ADDB  (R0)+,R9      ;DO A SIMPLE FUNCTION
      FA 51  F5  0234  639  SOBGTR R1,15$
      0237  640  ;
      FDC6' 30  0237  641  BSBW  MMG$FIND1STGSD ;POSITION POINTERS FOR GSD SEQ SEARCH
      63 50  E9  023A  642  BLBC  R0,ERROR_EXIT ;BR ON ERROR FINDING SPECIFIC SH MEM
      06 11  023D  643  BRB    25$             ;GO CHECK FIRST GSD
      00000000'GF 16  023F  644  20$:  JSB  G^MMG$GETNXTGSD ;GET NEXT LOGICAL GSD ENTRY ADR
      56  D5  0245  645  25$:  TSTL  R6             ;WAS THERE ANOTHER GSD?
      70  13  0247  646  BEQL  GS_NOT_FOUND ;BR ON NO MORE GSD'S
      29  OA A6  91  0249  647  CMPB  GSD$B_TYPE(R6),#DYN$C_SHMGSD ;IS THIS A SHARED MEM GSD?
      08  12  024D  648  BNEQ  27$             ;BR IF NOT SH MEM GSD
      EC 66  01  E0  024F  649  BBS  #GSD$V_LOCKED,GSD$L_GSDFL(R6),20$ ;BR IF GS IS LOCKED FOR READ
      E8 66  02  E0  0253  650  BBS  #GSD$V_DELPEND,GSD$[_GSDFL(R6),20$ ;BR IF GS IS MARKED FOR DELETE
      06 58  1F  E0  0257  651  27$:  BBS  #31,R8-30$ ;BRANCH IF NOT COMPARING GROUP CODE
      OE A6  58  B1  025B  652  CMPW  R8,GSD$W_PCBGRP(R6) ;DOES THIS SECTION BELONG TO THIS GROUP?
      DE  12  025F  653  BNEQ  20$             ;BRANCH IF NOT, TRY NEXT GSD
      50  22 A6  9E  0261  654  30$:  MOVAB GSD$T_GSDNAM(R6),R0 ;ASSUME LOCAL OR SHARED MEMORY GSD
      28  OA A6  91  0265  655  CMPB  GSD$B_TYPE(R6),#DYN$C_EXTGSD ;PFN-MAPPED GSD?
      04  12  0269  656  BNEQ  31$             ;IF NEQ NO, ASSUMPTION WAS CORRECT
      50  30 A6  9E  026B  657  MOVAB GSD$T_PFN$GSDNAM(R6),R0 ;POINT TO PFN-MAPPED SECTION NAME
      OB A6  59  91  026F  658  31$:  CMPB  R9,GSD$B_HASH(R6) ;DOES THE HASH MATCH
      CA  12  0273  659  BNEQ  20$             ;NO
      52  67  9B  0275  660  MOVZBW (R7),R2 ;GET LENGTH OF NAME TO MATCH
      80  52  91  0278  661  CMPB  R2,(R0)+ ;DOES LENGTH MATCH AT LEAST?
      C2  12  027B  662  BNEQ  20$             ;IF NEQ NO, TRY NEXT GSD
      60  01 A7  52  29  027D  663  CMPC3 R2,1(R7),(R0) ;DOES ACTUAL NAME STRING MATCH?
      BB  12  0282  664  BNEQ  20$             ;BRANCH IF NO, TRY NEXT GSD
      0284  665  ;
      0284  666  ASSUME SEC$K_MATALL EQ 0
      0284  667  ASSUME SEC$K_MATEQU EQ 1
      0284  668  ASSUME SEC$K_MATLEQ EQ 2
      0284  669  ;
      50  2C A7  D0  0284  670  MOVL  44(R7),R0 ;FETCH IDENT MATCH CONTROL
      13  13  0288  671  BEQL  33$             ;BRANCH IF ALWAYS MATCH
      1B A6  33 A7  91  028A  672  CMPB  <44+7>(R7),GSD$L_IDENT+3(R6) ;DOES MAJOR ID MATCH
      AE  12  028F  673  BNEQ  20$             ;BRANCH IF NOT
      18 A6  30 A7  D1  0291  674  CMPL  <44+4>(R7) GSD$L_IDENT(R6) ;NOW COMPARE ENTIRE LONG WORD
      A7  1A  0296  675  BGTRU 20$ ;IF LARGER, THEN NO MATCH
      02 50  E9  0298  676  BLBC  R0,33$ ;BRANCH IF EXACT MATCH NOT REQUIRED
      A2  12  029B  677  BNEQ  20$             ;BRANCH IF NOT AN EXACT MATCH

```

	50	01	3C	029D	678	33\$:	MOVZWL	#SS\$NORMAL,R0		:INDICATE GSD FOUND
				02A0	679	ERROR_EXIT:				
	5A	6A	D0	02A0	680		MOVL	(R10),R10		:SET INDICATOR IF GS IS IN SHMEM
	5E	28	C0	02A3	681	34\$:	ADDL	#<10*4>,SP		:CLEAN OFF GS DESC,SHM DESC AND BUFFER
	0B00	8F	BA	02A6	682	35\$:	POPR	#^M<R8,R9,R11>		:RESTORE REGISTERS
55	00000000	'FF	DE	02AA	683		MOVAL	@L^MMG\$GL_SYSPHD,R5		:SYSTEM PROCESS HEADER ADDRESS
			05	02B1	684		RSB			:AND RETURN
				02B2	685	NO_GS_NAME:				
50	0154	8F	3C	02B2	686		MOVZWL	#SS\$IVLOGNAM,R0		:INDICATE NO GS NAME SPECIFIED
		E7	11	02B7	687		BRB	ERROR_EXIT		:RETURN
				02B9	688	GS_NOT_FOUND:				
50	0978	8F	3C	02B9	689		MOVZWL	#SS\$NOSUCHSEC,R0		:INDICATE GSD NOT FOUND
		E0	11	02BE	690		BRB	ERROR_EXIT		:RETURN
				02C0	691					
				02C0	692	.DSABL	LSB			

```

02C0 694      .SBTTL DELGBLSEC - DELETE GLOBAL SECTION SUBROUTINE
02C0 695      :++
02C0 696      : FUNCTIONAL DESCRIPTION:
02C0 697      :
02C0 698      :
02C0 699      : CALLING SEQUENCE:
02C0 700      :
02C0 701      :     BSBW     MMG$DELGBLSEC
02C0 702      :
02C0 703      : INPUT PARAMETERS:
02C0 704      :
02C0 705      :     R1 = GLOBAL SECTION TABLE INDEX
02C0 706      :     R3 = GLOBAL SECTION TABLE ADDRESS
02C0 707      :     R5 = SYSTEM PROCESS HEADER ADDRESS
02C0 708      :
02C0 709      :     IPL = ASTDEL
02C0 710      :     GLOBAL SECTION MUTEX LOCKED FOR WRITING
02C0 711      :
02C0 712      : IMPLICIT INPUTS:
02C0 713      :
02C0 714      :     NONE
02C0 715      :
02C0 716      : OUTPUT PARAMETERS:
02C0 717      :
02C0 718      :     NONE
02C0 719      :
02C0 720      : IMPLICIT OUTPUTS:
02C0 721      :
02C0 722      :     NONE
02C0 723      :
02C0 724      : COMPLETION CODES:
02C0 725      :
02C0 726      :     NONE
02C0 727      :
02C0 728      : SIDE EFFECTS:
02C0 729      :
02C0 730      :     NONE
02C0 731      :
02C0 732      : --
02C0 733      :
02C0 734      : *****
02C0 735      :
02C0 736      : ***** THE FOLLOWING CODE MUST BE RESIDENT *****
02C0 737      :
00000000 738      :     .PSECT $MMGCODE
0000 739      :
0000 740      : *****
0000 741      :
0000 742      :
0000 743      : DELGBLSEC BUG:
0000 744      :     BUG_CHECK DELGBLSEC,FATAL      ;BAD MASTER PTE FORMAT FOR THIS CONTEXT
0004 745      :
0004 746      : MMG$DELGBLSEC::
0004 747      :     MOVL     SEC$L_GSD(R3),R2      ;GET ADR OF GLOBAL SECTION DESCRIPTOR
0007 748      :     BEQL     10$,R2               ;BRANCH IF PARTIALLY CREATED (NO GSD)
0009 749      :     REMQUE   (R2),R2              ;REMOVE GSD FROM NORMAL LIST
00000000'EF 62 0E 000C 750      :     INSQUE   (R2),L^EXE$GL_GSDDELFL ;PUT GSD ON DELPEND LIST, PROHIBIT USE

```

```

50 1C A3 DO 0013 751 10$: MOVL SECSL_PAGCNT(R3),R0 ;NUMBER OF PAGES IN SECTION
      03 12 0017 752 BNEQ 11$ ;CHECK FOR INCOMPLETE SECTION
      00A8 31 0019 753 BRW 75$
      08 BB 001C 754 11$: PUSHR #*M<R0,R1,R3> ;PUSH SECTION ADR, INDEX, PAGE COUNT
      16 00 EF 001E 755 EXTZV #SECSV_VPX,#SECS$ VPX,- ;FIRST GLOBAL PAGE TABLE INDEX
      53 53 08 A3 0021 756 SECSL_VPXPC(R3),R3
53 0000'DF43 DE 0024 757 MOVAL @W^MMGSGL_GPTBASE[R3],R3 ;ADDRESS OF FIRST GPTE
      09 BB 002A 758 PUSHR #*M<R0,R3> ;PUSH 1ST GPTE ADR, GPTE COUNT
      002C 759 :
      002C 760 : 0(SP) = COUNT OF GPTE'S
      002C 761 : 4(SP) = FIRST GPTE ADDRESS
      002C 762 : 8(SP) = COUNT OF GPTE'S
      002C 763 : 12(SP) = GLOBAL SECTION INDEX
      002C 764 : 16(SP) = GLOBAL SECTION ADDRESS
      002C 765 :
      002C 766 SETIPL #IPL$_SYNCH ;RAISE TO SYNCH FOR THIS SCAN
      002F 767 :
      002F 768 : THE FOLLOWING SCAN OF THE GLOBAL PAGE TABLE ENTRIES IS DONE TO ELIMINATE
      002F 769 : ALL TRANSITION PAGES POINTING AT THE GPTE'S.
      002F 770 : THE SCAN IS SKIPPED FOR RESIDENT GLOBAL SECTIONS
      002F 771 :
      002F 772 MOVL 16(SP),R0 ;GET SECTION TABLE ENTRY
50 83 50 10 AE DO 002F 773 BBS #SECSV_RESIDENT,SECSW_FLAGS(R0),50$
      2A 14 A0 OD EO 0033 773 BBS #SECSV_RESIDENT,SECSW_FLAGS(R0),50$
      78800000 8F CB 0038 774 20$: BICL3 #*C<PTESM_VALID ! - ;GET THE VALID BIT
      0040 775 PTESM_TYPT ! PTESM_TYPO ! - ;PTE TYPE BITS
      0040 776 PTESM_GPTX>,(R3)+,R0 ;AND PFN/GPTX BITS FROM THE PTE
      1D 13 0040 777 BEQL 40$ ;DZRO
      BC 19 0042 778 BLSS DELGBLSEC BUG ;BRANCH IF INCONSISTENT
51 50 EA 8F 78 0044 779 ASHL #-PTESV_TYPO,R0,R1 ;ONLY TYPE BITS LEFT, RIGHT ADJUSTED
      14 12 0049 780 BNEQ 40$ ;BRANCH IF NOT A TRANSITION PAGE
      004B 781 ASSUME PFNSC_FREPAGLST EQ 0
      03 00 EF 004B 782 EXTZV #PFNSV_LOC,#PFNS$ LOC,- ;GET THE PAGE'S LOCATION
52 0000'DF40 004E 783 @W^PFNSAB_STATE[R0],R2
      03 13 0053 784 BEQL 30$ ;IF NOT ON FREE LIST, WAIT UNTIL IT IS
      009E 31 0055 785 BRW 100$
      53 DD 0058 786 30$: PUSHL R3 ;SAVE REGISTERS AROUND THE FOLLOWING
      FFA3' 30 005A 787 BSBW MMGSDELPFNLIST ;DELETE PAGE FROM PFN LIST
      08 BA 005D 788 POPR #*M<R3> ;RESTORE REGISTERS
      D6 6E FS 005F 789 40$: SOBGTR (SP),20$ ;ONCE FOR EACH GLOBAL PAGE TABLE ENTRY
      8E D5 0062 790 50$: TSTL (SP)+ ;CLEAN OFF EXHAUSTED COUNT
      53 6E DO 0064 791 MOVL (SP),R3 ;1ST GPTE ADDRESS
50 04 AE DO 0067 792 MOVL 4(SP),R0 ;COUNT OF GPTE'S
      006B 793 :
      006B 794 : THE SECOND SCAN OF THE GLOBAL PAGE TABLE ENTRIES IS DONE TO DECREMENT
      006B 795 : THE GLOBAL PAGE TABLE REFERENCE COUNT AND TO RELEASE PAGING FILE
      006B 796 : BACKING STORE. THIS WAS NOT DONE IN THE PREVIOUS SCAN BECAUSE THAT
      006B 797 : SCAN REQUIRED THAT THERE BE NO SIDE EFFECTS SO THAT IT COULD FAIL
      006B 798 : AND BE RESTARTED.
      006B 799 :
      63 D5 006B 800 60$: TSTL (R3) ;IF VALID, ASSUME RESIDENT GLOBAL SECTION
      18 19 006D 801 BLSS 70$
      2E 63 16 EO 006F 802 BBS #PTESV_TYPO,(R3),72$ ;BRANCH IF NOT PAGING FILE BACKING STORE
      09 BB 0073 803 PUSHR #*M<R0,R3> ;SAVE REGISTER AROUND THE FOLLOWING
50 63 16 00 EF 0075 804 EXTZV #PTESV_PGFLVB,#PTES$ PGFLVB,(R3),R0 ;GET PAGE VBN TO DEALLOCATE
      07 13 007A 805 BEQL 65$ ;DZRO
      53 1F A5 9A 007C 806 MOVZBL PHDSB_PAGFIL(R5),R3 ;INDEX OF SYSTEM PAGING FILE
      FF7D' 30 0080 807 BSBW MMGSDELCPAGFIL ;DEALLOCATE PAGING FILE

```

```

09 BA 0083 808 65$: POPR #*M<R0,R3> ;RESTORE REGISTERS
1A 11 0085 809 BRB 72$
50 DD 0087 810 70$: PUSHL R0
63 15 00 EF 0089 811 EXTZV #PTESV PFN,#PTES$ PFN,(R3),R0 ;GET THE PFN
00000000'FF40 B7 008E 812 DECV @PFNS$@ REFCNT[R0] ;ONE LESS REFERENCE
0000'DF40 10 88 0095 813 BISB #PFNS$M DELCON,@W*PFNS$AB_STATE[R0] ;DON'T NEED IT ANY MORE
FF62' 30 0098 814 BSBW MMGS$R[PFN] ;FREE THE PAGE
50 8E DO 009E 815 MOVL (SP)+,R0
FF5C' 30 00A1 816 72$: BSBW MMGS$D&CPTREF ;ONE LESS PAGE TABLE REFERENCE
83 D4 00A4 817 CLRL (R3)+ ;DELETE THE GPTE
C2 50 F5 00A6 818 SOBGR RO,60$ ;NEXT GPTE
00A9 819
00A9 820 : 0(SP) = FIRST GPTE ADDRESS
00A9 821 : 4(SP) = COUNT OF GPTE'S
00A9 822 : 8(SP) = GLOBAL SECTION INDEX
00A9 823 : 12(SP) = GLOBAL SECTION ADDRESS
00A9 824
00A9 825 SETIPL #IPL$ ASTDEL ;BACK TO CALLED IPL
50 8E 04 C3 00AC 826 SUBL3 #4,(SP)+,R0 ;ADDRESS OF FRONT STOPPER FOR THIS SECTION
51 8E 03 C1 00B0 827 ADDL3 #3,(SP)+,R1 ;GPTE COUNT + 2 FOR STOPPERS + 1 TO ROUND EV
51 01 CA 00B4 828 BICL #1,R1 ;ROUND TO EVEN NO. OF GPTE'S
51 04 C4 00B7 829 MULL #4,R1 ;NO. OF BYTES OF GPTE TO RELEASE
53 0000'CF DE 00BA 830 MOVAL W*EXE$GL GPT,R3 ;TO GPT FREE POOL
FF3E' 30 00BF 831 BSBW EXE$DEAL[LOCATE] ;RELEASE THEM
0A BA 00C2 832 POPR #*M<R1,R3> ;R1=SECTION INDEX, R3=SECTION ADDRESS
50 0C A3 DO 00C4 833 75$: MOVL SEC$L_WINDOW(R3),R0 ;GET WCB ADDRESS FOR SECTION
40 13 00C8 834 BEQL 120$ ;IF NO WINDOW, SKIP ALL THIS STUFF
05 0B A0 03 E1 00CA 835 BBC #WCB$V SHRWCB,WCB$B_ACCESS(R0),80$ ;BRANCH IF NOT SHARED WINDOW
OE A0 B7 00CF 836 DECV WCB$W_REFCNT(R0) ;LAST REFERENCE ON SHARED WINDOW?
09 14 00D2 837 BGTR 90$ ;BRANCH IF NOT
16 A0 B4 00D4 838 80$: CLRW WCB$W_NMAP(R0) ;NO RETRIEVAL POINTERS
00D7 839
00D7 840 ASSUME WCB$W_P1_COUNT&3 EQ 0 ;STARTS AT LONG WORD OFFSET
0000'CF 30 A0 OE 00D7 841 INSQUE WCB$W_P1_COUNT(R0),W*EXE$GL_WCBDELFL ;QUEUE WINDOW ON WCB DELETE LIS
00DD 842
00DD 843 ASSUME SEC$L_GSD EQ 0 ;1ST LONG WORD IN SECTION IS GSD ADDRESS
50 63 DO 00DD 844 90$: MOVL (R3),R0 ;GET GSD ADDRESS
OE 13 00E0 845 BEQL 95$ ;BRANCH IF NONE
50 60 OF 00E2 846 REMQUE (R0),R0 ;REMOVE GLOBAL SECTION DESCRIPTOR
00E5 847 ;FROM GSD ACTIVE LIST
00E5 848 PUSHL R1 ;SAVE R1 ACROSS CALL
00000000'GF 16 00E7 849 JSB G*EXE$DEAPAGED ;DEALLOCATE GSD TO PAGED POOL
51 8E DO 00ED 850 MOVL (SP)+,R1 ;RESTORE R1
00000000'EF 17 00F0 851 95$: JMP MMGS$DALCSTX ;DEALLOCATE THE GLOBAL SECTION INDEX
00F6 852 ;AND RETURN
00F6 853
00F6 854 : PAGE IS IN TRANSITION STATE. IT IS GUARANTEED TO END UP ON THE FREE PAGE
00F6 855 : LIST WHEN WHATEVER IS GOING ON IS COMPLETED
00F6 856
0000'CF D4 00F6 857 100$: CLRL W*SCH$GL_MFYLOLIM ;FORCE THE MODIFIED PAGE WRITER TO
0000'CF B4 00FA 858 CLRW W*SCH$GL_MFYLIM ;START, TO GET PAGES ON FREE LIST
00FE 859 SETIPL #IPL$ ASTDEL ;BACK TO CALLER'S IPL
5E 14 C0 0101 860 ADDL #5*4,SP ;CLEAN UP THE STACK
00 36 A5 01 E6 0104 861 BBSSI #PHD$V_DALCSTX,PHD$W_FLAGS(R5),110$ ;SECTION STILL TO BE DEALLOCATED
05 0109 862 110$: RSB ;AND RETURN
010A 863
010A 864 : Add page count to the global section page file limit

```

SYSDGBLSC  
V04-000

N 13  
- DELETE GLOBAL SECTION SYSTEM SERVICE 16-SEP-1984 02:00:54 VAX/VMS Macro V04-00  
DELGBLSEC - DELETE GLOBAL SECTION SUBROU 5-SEP-1984 03:52:42 [SYS.SRC]SYSDGBLSC.MAR;1

Page 20  
(9)

S  
V

00000000'EF 1C A3 CO 010A 865 ;  
C9 11 0112 866 120\$: ADDL2 SECSL\_PAGCNT(R3),MMG\$GL\_GBLPAGFIL  
BRB 90\$

```

0114 869 .SBTTL DELGBLWCB - DELETE GLOBAL WINDOW CONTROL BLOCKS
0114 870 :++
0114 871 : FUNCTIONAL DESCRIPTION:
0114 872 :
0114 873 : THIS ROUTINE REMOVES ANY WINDOW CONTROL BLOCKS FROM THE WCBDEL
0114 874 : LIST AND DEACCESSSES THE FILES BY FILLING IN A CHANNEL AND DFPASSIGNING IT
0114 875 :
0114 876 : CALLING SEQUENCE:
0114 877 :
0114 878 : BSBW MMG$DELGBLWCB
0114 879 :
0114 880 : INPUT PARAMETERS:
0114 881 :
0114 882 : R4 = PCB ADDRESS OF PROCESS
0114 883 : IPL = ASTDEL, MAY BE LOWERED TO 0 FOR A WHILE, BUT RETURNS AT ASTDEL
0114 884 :
0114 885 : IMPLICIT INPUTS:
0114 886 :
0114 887 : NONE
0114 888 :
0114 889 : OUTPUT PARAMETERS:
0114 890 :
0114 891 : NONE
0114 892 :
0114 893 : IMPLICIT OUTPUTS:
0114 894 :
0114 895 : NONE
0114 896 :
0114 897 : COMPLETION CODES:
0114 898 :
0114 899 : NONE
0114 900 :
0114 901 :--
0114 902 :
0114 903 :*****
0114 904 :
0114 905 :***** THE FOLLOWING CODE MAY BE PAGED *****
0114 906 :
000002C0 907 .PSECT Y$EXEPAGED
02C0 908 :
02C0 909 :*****
02C0 910 :
02C0 911 :
02C0 912 MMG$DELGBLWCB::
00000000'EF D5 02C0 913 TSTL CTL$GL_F11BXQP ;HAS FILE SYSTEM BEEN MAPPED
OE 13 02C6 914 BEQL 15$ ;NO - TRY AGAIN LATER
OE A4 B5 02C8 915 TSTW PCB$W_MTXCNT(R4) ;DOES THE PROCESS OWN ANY MUTEXES?
09 12 02CB 916 BNEQ 15$ ;BR IF IT DOES, CAN'T CALL ANY SYS SRV
53 00000000'FF OF 02CD 917 10$: REMQUE @L^EXE$GL_WCBDELFL,R3 ;GET NEXT WCB TO DEACCESS
01 1C 02D4 918 BVC 20$ ;BRANCH IF GOT ONE
05 02D6 919 15$: RSB ;NO MORE ON THE LIST
53 30 C2 02D7 920 20$: SUBL #WCB$W_P1_COUNT,R3 ;ADDRESS OF WINDOW CONTROL BLOCK
FD23' 30 02DA 921 BSBW IOC$FFCHAN ;FIND A FREE CHANNEL
3A 50 E9 02DD 922 BLBC RO,40$ ;BRANCH IF NONE AVAILABLE
02EC 923 :
02EC 924 : R1 = CHANNEL NUMBER
02E0 925 : R2 = CCB ADDRESS

```



```

OF OB A3 03 E0 02E0 926 ;
                    02E0 927 ;
                    02E5 928 ;
50 0080 C4 D0 02E5 929 ;
    30 A0 B5 02EA 930 ;
    2B 13 02ED 931 ;
    30 A0 B7 02EF 932 ;
    03 11 02F2 933 ;
    02F4 934 ;
    OE A3 B6 02F4 935 25$: INCW WCB$W_REFCNT(R3) ;COUNT A REFERENCE IF SHARED WINDOW
    02F7 936 ;
50 10 A3 D0 02F7 937 30$: MOVL WCB$L_ORGUCB(R3),R0 ;UCB ADDRESS FROM WINDOW
    5C A0 B6 02FB 938 INCW UCB$W_REFC(R0) ;COUNT A REFERENCE AND
    62 50 D0 02FE 939 MOVL R0,CCB$L_UCB(R2) ;PUT IT IN THE CHANNEL
04 A2 53 D0 030 940 MOVL R3,CCB$L_WIND(R2) ;PUT WINDOW IN CHANNEL
    09 A2 96 0305 941 INCB CCB$B_AMOD(R2) ;ASSIGN CHANNEL IN KERNEL MODE
    0308 942 SETIPL #0 ;DROP TO IPL 0
    030B 943 $DASSGN_S R1 ;DEASSIGN CHANNEL (THUS DEACCESSING FILE)
    0315 944 SETIPL #IPL$_ASTDEL ;BACK TO ASTDEL
    B3 11 0318 945 BRB 10$ ;GO GET ANOTHER WCB IF ANY MORE
    031A 946 ;
    031A 947 ; R3 = WCB ADDRESS, PUT IT BACK ON THE WCBDEL QUEUE
    031A 948 ;
00000000'EF 30 A3 OE 031A 949 40$: INSQUE WCB$W_P1_COUNT(R3),L^EXE$GL WCBDELFL ;RE-QUEUE THE WCB
    05 0322 950 RSB ;AND RETURN
    0323 951
    0323 952
    0323 953
.END

```

ACCVIO	0000019E	R	02	MMG\$GETNXTGSD	*****	X	02
BADIDENT	00000197	R	02	MMG\$GL_GBLPAGFIL	*****	X	03
BADNAME	00000190	R	02	MMG\$GL_GPTBASE	*****	X	03
BUGS_DELGBLSEC	*****	X	03	MMG\$GL_SYSPHD	*****	X	02
CCBSB_AMOD	= 00000009			MMG\$GSDMTXULK	00000074	RG	02
CCBSL_UCB	= 00000000			MMG\$GSDSCN	000001A6	RG	02
CCBSL_WIND	= 00000004			MMG\$L_VFYFLAGS	= FFFFFFFE8		
CTLSGL_F11BXQP	*****	X	02	MMG\$RELPFN	*****	X	03
CTLSGL_PHD	*****	X	02	MMG\$VIFYSECFLG	00000000	RG	02
DELGBLSEC_BUG	00000000	R	03	NOT MAP TO FILE	00000170	R	02
DIR...	= 00000001			NO GS_NAME	000002B2	R	02
DYN\$C_EXTGSD	= 00000028			PCBSL_JIB	= 00000080		
DYN\$C_SHM\$SD	= 00000029			PCBSQ_PRIV	= 00000084		
ERREXIT	000001A1	R	02	PCBSW_GRP	= 000000BE		
ERROR_EXIT	000002A0	R	02	PCBSW_MTXCNT	= 0000000E		
EXES\$DEALLOCATE	*****	X	03	PFNSAB_STATE	*****	X	03
EXES\$DEAPAGED	*****	X	02	PFNSAW_REFCNT	*****	X	03
EXESDGBLSC	00000094	RG	02	PFNSC_FREPAGLST	= 00000000		
EXESGL_GPT	*****	X	03	PFNSM_DELCON	= 00000010		
EXESGL_GSDDELFL	*****	X	02	PFNSS_LOC	= 00000003		
EXESGL_GSDMTX	*****	X	02	PFNSV_LOC	= 00000000		
EXESGL_WCBDELFL	*****	X	03	PHDSB_PAGFIL	= 0000001F		
EXESMAXACMODE	*****	X	02	PHDSL_PSTBASOFF	= 00000020		
FLAGS	00000004			PHDSQ_PRIVMSK	= 00000000		
GSD\$B_CREATPORT	= 00000052			PHDSV_DALCSTX	= 00000001		
GSD\$B_HASH	= 0000000B			PHDSW_FLAGS	= 00000036		
GSD\$B_TYPE	= 0000000A			PR\$ IPL	= 00000012		
GSD\$S_GSDFL	= 00000000			PRIV ERR	00000089	R	02
GSD\$S_IDENT	= 00000018			PRV\$V_PFNMAP	= 0000001A		
GSD\$T_GSDNAM	= 00000022			PRV\$V_PRMGBL	= 00000018		
GSD\$T_PFN\$SDNAM	= 00000030			PRV\$V_SHMEM	= 0000001B		
GSD\$V_DELPEND	= 00000002			PRV\$V_SYSGBL	= 00000019		
GSD\$V_LOCKED	= 00000001			PSL\$C_USER	= 00000003		
GSD\$W_FLAGS	= 00000020			PTE\$M_GPTX	= 003FFFFFF		
GSD\$W_GSTX	= 00000016			PTE\$M_TYPO	= 00400000		
GSD\$W_PCBGRP	= 0000000E			PTE\$M_TYPI	= 04000000		
GSDNAM	00000008			PTE\$M_VALID	= 80000000		
GS_NOT_FOUND	000002B9	R	02	PTE\$S_PFN	= 00000015		
IDENT	0000000C			PTE\$S_PGFLVB	= 00000016		
IOCSFFCHAN	*****	X	02	PTE\$V_PFN	= 00000000		
IPL\$ASTDEL	= 00000002			PTE\$V_PGFLVB	= 00000000		
IPL\$SYNCH	= 00000008			PTE\$V_TYPO	= 0C000016		
JIB\$Q_FILCNT	= 00000030			SAVABS...	= 00000010		
LNMSC_NAMLENGTH	= 000000FF			SCH\$GL_CURPCB	*****	X	02
MMG\$C_LENGTH	= FFFFFFFE4			SCH\$GL_MFYLIM	*****	X	03
MMG\$DALCPAGFIL	*****	X	03	SCH\$GL_MFYLOLIM	*****	X	03
MMG\$DALCSTX	*****	X	03	SCH\$LOCKW	*****	X	02
MMG\$DALCSTXSCN	*****	X	02	SCH\$UNLOCK	*****	X	02
MMG\$DECPTRF	*****	X	03	SEC\$K_MATALL	= 00000000		
MMG\$DECSHMREF	*****	X	02	SEC\$K_MATEQU	= 00000001		
MMG\$DELGBLSEC	00000004	RG	03	SEC\$K_MATLEQ	= 00000002		
MMG\$DELGBLWCB	000002C0	RG	02	SEC\$S_GSD	= 00000000		
MMG\$DELPFNLIST	*****	X	03	SEC\$S_PAGCNT	= 0000001C		
MMG\$DGBLSC1	0000008D	RG	02	SEC\$S_REFCNT	= 00000018		
MMG\$FIND1STGSD	*****	X	02	SEC\$S_VPX_FC	= 00000008		
MMG\$FINDGSNOTRN	*****	X	02	SEC\$S_WIN_OW	= 000C000C		
MMG\$GETGSNAM	*****	X	02	SEC\$M_CRF	= 00000002		

```

SECSM_DZRO      = 00000004
SECSM_EXECUTE   = 00100000
SECSM_EXPREG    = 00020000
SECSM_GBL       = 00000001
SECSM_PAGFIL    = 00080000
SECSM_PERM      = 00004000
SECSM_PFNMAP    = 00010000
SECSM_PROTECT   = 00040000
SECSM_RESIDENT  = 00002000
SECSM_SYSGBL    = 00008000
SECSM_WRT       = 00000008
SECSM_WRTMOD    = 000000C0
SECSS_VPX       = 00000016
SECSS_WRTMOD    = 00000002
SECSV_CRF       = 00000001
SECSV_DZRO      = 00000002
SECSV_GBL       = 00000000
SECSV_PAGFIL    = 00000013
SECSV_PERM      = 0000000E
SECSV_PFNMAP    = 00000010
SECSV_PROTECT   = 00000012
SECSV_RESIDENT  = 0000000D
SECSV_SYSGBL    = 0000000F
SECSV_VPX       = 00000000
SECSV_WRTMOD    = 00000006
SECSW_FLAGS     = 00000014
SHBSB_PORT      = 00000015
SHMEM_DEL       = 00000140 R    02
SS$_ACCVIO      = 0000000C
SS$_IVLOGNAM    = 00000154
SS$_IVSECFLG    = 0000016C
SS$_IVSECIDCTL  = 000002E4
SS$_NOPRIV      = 00000024
SS$_NORMAL      = 00000001
SS$ NOSUCHSEC   = 00000978
SS$ NOTCREATOR  = 00000384
SYSDASSGN       = ***** GX 02
UCBSW_REFC      = 0000005C
WCBSB_ACCESS    = 0000000B
WCBSL_ORGUCB    = 00000010
WCBSV_SHRUCB    = 00000003
WCBSW_NMAP      = 00000016
WCBSW_P1 COUNT  = 00000030
WCBSW_REFCNT    = 0000000E
    
```

-----  
! 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	00000010 ( 16.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
Y\$EXEPAGED	00000323 ( 803.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$MMGCOD	00000114 ( 276.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.10	00:00:01.03
Command processing	105	00:00:00.50	00:00:04.52
Pass 1	459	00:00:18.13	00:00:56.18
Symbol table sort	0	00:00:02.91	00:00:06.58
Pass 2	179	00:00:03.87	00:00:08.58
Symbol table output	20	00:00:00.17	00:00:00.24
Psect synopsis output	1	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	795	00:00:25.72	00:01:17.16

The working set limit was 1800 pages.  
102297 bytes (200 pages) of virtual memory were used to buffer the intermediate code.  
There were 100 pages of symbol table space allocated to hold 1815 non-local and 57 local symbols.  
953 source lines were read in Pass 1, producing 21 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:[SYS.OBJ]LIB.MLB;1	19
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	13
TOTALS (all libraries)	32

1986 GETS were required to define 32 macros.

There were no errors, warnings or information messages.

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



